INTERNET ARCHIVE
archive.org

AFFIDAVIT OF NATHANIEL E FRANK-WHITE

1. I am a Records Request Processor at the Internet Archive. I make this declaration of my own personal knowledge.

2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to browse more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a display of available dates. The visitor may select one of those dates, and begin browsing an archived version of the Web. Links on archived files in the Wayback Machine point to other archived files (whether HTML pages or other file types), if any are found for the URL indicated by a given link. For instance, the Wayback Machine is designed such that when a visitor clicks on a hyperlink on an archived page that points to another URL, the visitor will be served the archived file found for the hyperlink's URL with the closest available date to the initial file containing the hyperlink.

4. The archived data made viewable and browsable by the Wayback Machine is obtained by use of web archiving software that automatically stores copies of files available via the Internet, each file preserved as it existed at a particular point in time.

5. The Internet Archive assigns a URL on its site to the archived files in the format http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] aka an "extended URL". Thus, the extended URL http://web.archive.org/web/19970126045828/http://www.archive.org/ would be the URL for the record of the Internet Archive home page HTML file (http://www.archive.org/) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). The date indicated by an extended URL applies to a preserved instance of a file for a given URL, but not necessarily to any other files linked therein. Thus, in the case of a page constituted by a primary HTML file and other separate files (e.g., files with images, audio, multimedia, design elements, or other embedded content) linked within that primary HTML file, the primary HTML file and the other files will each have their own respective extended URLs and may not have been archived on the same dates.

6. Attached hereto as Exhibit A are true and accurate copies of browser screenshots of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each printout.

MICROSOFT CORP.
EXHIBIT 1258

Document Id: B0B7A1C0-84EA-11EE-8540-1FF845118E42
OnlineNotary.net

INTERNET ARCHIVE
archive.org

7. Attached hereto as Exhibit B are true and accurate copies of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each file.

8. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

DATE: 11/16/2023

Nathaniel Frank-White

Nathaniel E Frank-White

Document Id: B0B7A1C0-84EA-11EE-8540-1FF845118E42
OnlineNotary.net

JURAT ATTACHMENT

A notary public or other officer completing this certificate verifies only the identity of the individual who signed the document to which this certificate is attached, and not the truthfulness, accuracy, or validity of that document.

STATE OF Virginia }

COUNTY OF Alexandria }

The foregoing instrument was subscribed and sworn before me this date of 11/16/2023, by Nathaniel Frank-White

This notarial act was an online notarization.

Document Notarized using a Live Audio-Video Connection

AMEENHA HAFEESAH GUILLORY
ELECTRONIC NOTARY PUBLIC
COMMONWEALTH OF VIRGINIA
NOTARY ID: 7889093
COMISSION EXP: JUNE 30, 2024

(Notary Seal)

Notary's Signature

Registration No.: 7889093

Commission Expiration Date: June 30, 2024

Document Id: B0B7A1C0-84EA-11EE-8540-1FF845118E42
OnlineNotary.net

EXHIBIT A

https://web.archive.org/web/20051125231148/http://www.nokiausa.com/support/phones/guides/1,7840,9300,00.html

INTERNET ARCHIVE
WayBackMachine
http://www.nokiausa.com/support/phones/guides/1%2C7840%2C9300%2C00.html
Go
OCT NOV FEB
20 captures
25 Nov 2005 - 21 Jun 2007
2004 25 2005 2007
About this capture

Nokia 9300 Smartphone Support

User's Guide and Interactive Demonstration

The User's Guide contains basic instructions on using your Nokia 9300 Smartphone and configuring its settings. You may download a user guide at no charge.

Download Nokia 9300 Smartphone User Guide

Nokia 9300 User Guide in English PDF Download now
9300_US_en.PDF
3.78 MB | English

Nokia 9300 User Guide in Spanish PDF Download now
9300_US_es.PDF
5.18 MB | Spanish

In order to read PDF files, you need to have Adobe® Acrobat® Reader® installed in your computer. You can download the latest Acrobat® Reader® for free from Adobe's website.

Get Acrobat Reader

FAQ Search

Search for answers to questions about Nokia products and services.
Search now >>

Warranty Registration

Register your Nokia warranty online.
Register now >>

Copyright © 2005 Nokia. All rights reserved. Nokia, Nokia Connecting People, Xpress-on, Pop-Port, and all Nokia phone models are trademarks or registered trademarks of Nokia Corporation.

INTERNET ARCHIVE WayBackMachine
http://www.nokiausa.com/phones/9300/0%2C7747%2C%2C00.html
Go JUN AUG 01 NOV
99 captures
18 Mar 2005 - 11 Feb 2021
2004 2005 2006
About this capture

NOKIA Connecting People

Select a location
United States

Search GO

31-Jul-2005

Items: 0 Total: $0.00 Order Status

NOKIA 9300

Main Page
Phone Features
Spotlight
Accessories

Coming Soon
Sign up now for e-mail updates about the **Nokia 9300** smartphone.
Sign up >>

E-mail Page & Receive $5 Accessory E-coupon >>

Nokia 9300 Smartphone Features

- Advanced voice features: handsfree speakerphone and conference calling capability[1]
- Full keyboard and two 65,536-color displays
- Messaging options: email with attachments, SMS, Instant Messaging and MMS[1,14]
- Office applications: Documents, Sheet, and Presentations (supports the most common features of Microsoft Word, Excel and PowerPoint; Microsoft Office 97, 98, 2000, XP and 2003)
- Organizer (calendar, contacts, tasks) and email with PC synchronization via Nokia PC Suite software[4]
- Large memory storage: 80MB built-in memory plus Multimedia Memory Card (MMC)
- High-speed data connectivity; up to 53.6 kbps GPRS, up to 236.8 kbps EGPRS (EDGE)[1]

See more Nokia 9300 smartphone features >>

Tools

Compare Phones >>
View and compare Nokia phones side-by-side.

Shopping Assistant >>
Find the right phone for you.

Nokia 9300 Support

Warranty Information >>
User Guides >>
Free Newsletter >>
Support for Owners >>
FCC Certification Information >>
Eco Declaration >>

Accessories

Digital Pen§ >>
Price: $249.00

Wireless Boom Headset§ >>
Price: $119.00

Personalize

Downloadable ring tones, games and graphics are not yet available for the Nokia 9300 smartphone. Please check back.

---

(1) To use the phone you must have service from a service provider. Many features and mobile services require network support. Some networks have limitations that affect how you can use phone features. Your service provider also may have requested that certain features not be activated in a phone. If so, they may not appear in the phone's menu. Contact your service provider about feature support and availability.

(2) Some Mobile Internet WAP services have been pre-configured/specified by the mobile service provider.

(3) Obey all applicable local laws governing recording of conversations and/or calls.

(4) Requires Nokia 9300 PC suite, available for free download at http://www.nokia.com/us, and data cable. Data cable sold separately.

(5) Requires Nokia 9300 phone and compatible computer with infrared port. Nokia PC Suite available for free download at http://www.nokia.com/us

(12) Mobile Internet (Internet content/services designed for mobile devices)

(14) Only devices that offer compatible multimedia message or e-mail features can receive and display multimedia messages. Content appearance may vary. Some images and ring tones can not be forwarded.

‡ Phone may be used in countries or areas with compatible GSM (900/1800/1900 Mhz) frequency systems, subject to roaming services provided or arranged by your mobile provider.

Java is a trademark of Sun Microsystems, Inc.

(22) The Nokia 9300 phone is designed to meet Bluetooth Specification 1.1, Generic Access, Service Discovery, Dial-up Networking, Hands-Free, Serial Port, Generic Object Exchange, Object Push, SIM Access, and File Transfer profiles. To ensure compatibility with other Bluetooth devices, check with the device manufacturer.

<< Nokia e-coupon is worth $5 off of your phone or phone accessory purchase of $50 or more at NokiaUSA.com. Not valid on phones purchased with or without service at nokia.letstalk.com or letstalk.com. E-coupon expires December 31, 2005 and may not be combined with any other NokiaUSA.com e-coupons. E-Coupons are nontransferable and valid only for phone, phone accessory or Nokia Gear purchases made at the Nokia USA web site, URL: http://www.nokiausa.com/.

§ WARNING: Use only batteries, chargers, and enhancements approved by Nokia for use with this particular model. The use of any other types may invalidate any approval or warranty, and may be dangerous.

The use of mobile phones and their accessories may be prohibited or restricted in certain areas. Obey applicable law regarding usage.

Copyright © 2005 Nokia. All rights reserved. Nokia, Nokia Connecting People, Xpress-on, Pop-Port, and all Nokia phone models are trademarks or registered trademarks of Nokia Corporation.

https://web.archive.org/web/20070624105704/http://www.nokiausa.com/support/phones/guides/1,7840,N75,00.html

INTERNET ARCHIVE WayBackMachine
http://www.nokiausa.com/support/phones/guides/1%2C7840%2CN75%2C00.html
Go MAY JUN 24 JUL
3 captures
21 May 2007 - 24 Jun 2007
2006 2007 2008
About this capture

Nokia N75 Device Support

Support Main

Nokia N75 Device

Main Page
Tutorials
User's Guides
FAQ Search
Warranty

User's Guide and Interactive Demonstration

The User's Guide contains basic instructions on using your Nokia N75 Device and configuring its settings. You may download a user guide at no charge.

Download Nokia N75 Device User Guide		
Nokia N75 User Guide in English N75_US_en.PDF 11594.0 (KB) | English	PDF	Download now
Nokia N75 User Guide in Spanish N75_US_es.PDF 11594.0 (KB | Spanish	PDF	Download now
Nokia N75 Getting Started Guide in English N75_US_en_GS.PDF 1031.0 (KB) | English	PDF	Download now
Nokia N75 Getting Started Guide in Spanish N75_US_es_GS.PDF 1054.0 (KB) | English	PDF	Download now
Nokia N75 Additional Application Guide in Spanish N75_US_es_AAL.PDF 465.0 (KB) | English	PDF	Download now
Nokia N75 Additional Application Guide English N75_US_en_AAL.PDF 323 KB | English	PDF	Download now

Nokia N75 Device Interactive Demonstration	
Explore phone features, see simulations, learn tips and tricks. (Requires Macromedia Flash.)	START DEMO (opens in a pop up)

In order to read PDF files, you need to have Adobe® Acrobat® Reader® installed in your computer. You can download the latest Acrobat® Reader® for free from Adobe's website.

FAQ Search

Search for answers to questions about Nokia products and services. Search now >>

Warranty Registration

Register your Nokia warranty online.

Register now >>

Copyright © 2007 Nokia. All rights reserved. Nokia, Nokia Connecting People, Xpress-on, Pop-Port, and all Nokia phone models are trademarks or registered trademarks of Nokia Corporation.

INTERNET ARCHIVE WayBackMachine
http://www.nokiausa.com/phones/N75/0%2C7747%2C%2C00.html
Go APR JUN JUL
3 captures
29 Apr 2007 - 12 Jun 2007
2006 12 2007 2008
About this capture

Main Page
Device Features
Support and Software for Owners

N75 Accessories

Accessibility
Bluetooth
Car Kits
Cases & Clips
Data
Desktop
GPS
Headsets
Power

Something went wrong : (

Ruffle failed to load the Flash SWF file.

Access to fetch has likely been blocked by CORS policy.

If you are the server administrator, please consult the Ruffle wiki for help.

View Ruffle Wiki View Error Details

- 3G connectivity* with Cingular service
- Mini Map web browsing technology
- Dedicated music keys
- Large 2.4" crystal clear display
- Access to Cingular mobile video entertainment with Cingular service
- Headset not included

**Coverage areas may be limited in the U.S.*

- Service: AT&T; Wireless

See more Nokia N75 features >>

Charger Adaptor§ >>
Price: $9.95

Travel Charger§ >>
Price: $24.95

Downloadable ring tones, games and graphics are not yet available for the Nokia N75 Device . Please check back.

Tools

Compare Phones >>
View and compare Nokia phones side-by-side.

Nokia N75 Support

Warranty Information >>
User Guides >>
Free Newsletter >>
Support for Owners >>
FCC Certification Info >>
Eco Declaration >>

(1) To use the phone you must have service from a service provider. Many features and mobile services require network support. Some networks have limitations that affect how you can use phone features. Your service provider also may have requested that certain features not be activated in a phone. If so, they may not appear in the phone's menu. Contact your service provider about feature support and availability.

(2) Some Mobile Internet WAP services have been pre-configured/specified by the mobile service provider.

(12) Mobile Internet (Internet content/services designed for mobile devices)

(14) Only devices that offer compatible multimedia message or e-mail features can receive and display multimedia messages. Content appearance may vary. Some images and ring tones can not be forwarded.

(15) GPRS may not be available in all networks. GPRS data transmission speeds may be as high as 153 kbps, but may vary based upon network capabilities and other conditions. The establishment and continuation of a GPRS connection depends on network availability, provider support and signal strength.

(22) The Nokia N75 phone is designed to meet Bluetooth Specification 2.0, GAP General Access, OPP Object Push, FTP File Transfer, SPP Serial Port, HSP Headset, HFP Hands-free, DUN Dial-Up Network, FAX, BIP Basic Imaging, BPP Basic Printing, SIM Access, GOEP (Generic Object Exchange profiles. To ensure compatibility with other devices with Bluetooth technology, check with the device manufacturer. Bluetooth is a registered trademark of Bluetooth SIG, Inc.

† Shared Memory
Some phone features such as calendar, images and ringing tones in gallery, Java™ games and applications, phone book, multimedia and text messages, to-do notes may share memory. Using such features may reduce the memory available for any features sharing

memory. See your phone's user guide for details and additional information.

Bluetooth is a registered trademark of Bluetooth SIG, Inc.

Java is a trademark of Sun Microsystems, Inc.

þProduct and pricing availability based on service offered in your area. Subject to change. Reflects price after available promotions or rebates. Mail-in rebates may require customer to submit rebate form with proof of purchase to carrier, manufacturer, or other sponsor. May require a one or two year service activation with featured wireless carrier. One-time wireless activation fee and early termination fees may apply. Please check specific rebate instructions and carrier terms and conditions for details.

§ WARNING: Use only batteries, chargers, and enhancements approved by Nokia for use with this particular model. The use of any other types may invalidate any approval or warranty, and may be dangerous.

The use of mobile phones and their accessories may be prohibited or restricted in certain areas. Obey applicable law regarding usage.

Copyright © 2007 Nokia. All rights reserved. Nokia, Nokia Connecting People, Xpress-on, Pop-Port, and all Nokia phone models are trademarks or registered trademarks of Nokia Corporation.

https://web.archive.org/web/20060824212346/http://www.nokiausa.com/support/phones/guides/1,7840,N91,00.html

INTERNET ARCHIVE
WayBackMachine
http://www.nokiausa.com/support/phones/guides/1%2C7840%2CN91%2C00.html
Go
JUL AUG JAN
7 captures
24 Aug 2006 - 17 Apr 2021
2005 24 2006 2021
About this capture

Nokia N91 Device Support

User's Guide and Interactive Demonstration

The User's Guide contains basic instructions on using your Nokia N91 Device and configuring its settings. You may download a user guide at no charge.

Download Nokia N91 Device User Guide

Nokia N91 User Guide in English PDF Download now
N91_US_en.PDF
4.69 MB | English

Nokia N91 User Guide in Spanish PDF Download now
N91_US_es.PDF
4.69 MB | Spanish

Nokia N91 Device Interactive Demonstration

Explore phone features, see simulations, learn tips and tricks.
(Requires Macromedia Flash.)

START DEMO
(opens in a pop up)

In order to read PDF files, you need to have Adobe® Acrobat® Reader® installed in your computer. You can download the latest Acrobat® Reader® for free from Adobe's website.

Get Acrobat Reader

FAQ Search

Search for answers to questions about Nokia products and services.
Search now >>

Warranty Registration

Register your Nokia warranty online.
Register now >>

Copyright © 2006 Nokia. All rights reserved. Nokia, Nokia Connecting People, Xpress-on, Pop-Port, and all Nokia phone models are trademarks or registered trademarks of Nokia Corporation.

INTERNET ARCHIVE WayBackMachine
http://www.nokiausa.com/phones/N91/0%2C7747%2C%2C00.html
Go JUN JUL 03 2006 AUG 2005 2007
51 captures
19 Jun 2006 - 17 Apr 2021
About this capture

NOKIA N91

Main Page
Phone Features
Accessories
Support and Software for Owners

Nokia N91 Phone In Short

- Storage for up to 3000* songs (4GB) plus built-in FM Radio
- Seamless switching between music player and smartphone functions
- Create playlists on the go
- Free software update, expected to be available in July, will allow playback of Windows Media DRM (Digital Rights Management) protected audio files

*Capacity based on 3:45 per song and 48 kbps eAAC+ encoding. At 128 kbps, AAC encoding is up to 1000 songs.

- **Service: GSM 900/1800/1900 MHz[23]**

See more Nokia N91 features >>

Tools

Compare Phones >>
View and compare Nokia phones side-by-side.

Shopping Assistant >>
Find the right phone for you.

Nokia N91 Support

Warranty Information >>
User Guides >>
Free Newsletter >>
Support and Software for Owners >>
FCC Certification Information >>
Eco Declaration >>

Accessories

Extended Li-Ion Battery§ >>
Price: $49.95

Advanced Car Kit§ >>
Price: $179.00

Personalize

Downloadable ring tones, games and graphics are not yet available for the Nokia N91 phone. Please check back.

---

(1) To use the phone you must have service from a service provider. Many features and mobile services require network support. Some networks have limitations that affect how you can use phone features. Your service provider also may have requested that certain features not be activated in a phone. If so, they may not appear in the phone's menu. Contact your service provider about feature support and availability.

(2) Some Mobile Internet WAP services have been pre-configured/specified by the mobile service provider.

(3) Obey all applicable local laws governing recording of conversations and/or calls.

(4) Requires Nokia PC suite, available for free download at http://www.nokia.com/us, and data cable. Data cable sold separately.

(12) Mobile Internet (Internet content/services designed for mobile devices)

(14) Only devices that offer compatible multimedia message or e-mail features can receive and display multimedia messages. Content appearance may vary. Some images and ring tones can not be forwarded.

(19) Please notice that many jurisdictions have laws and regulations about taking pictures in public or private areas and regarding the processing and further use of such pictures. Nokia encourages its customers to obey all laws and to honor the personal rights of others.

† Shared Memory
Some phone features such as calendar, images and ringing tones in gallery, Java™ games and applications, phone book, multimedia and text messages, to-do notes may share memory. Using such features may reduce the memory available for any features sharing memory. See your phone's user guide for details and additional information.

§ Reserved Memory
Certain features of this phone have limited storage capacity, such as phone book, calendar, to-do list, ring tones and Java™ applications, voice dialing and commands. The amount of detail stored for an entry for such features may affect the total number of entries that can be stored.

Bluetooth is a registered trademark of Bluetooth SIG, Inc.

Java is a trademark of Sun Microsystems, Inc.

<< Nokia e-coupon is worth $5 off of your phone or phone accessory purchase of $50 or more at NokiaUSA.com. Not valid on phones purchased with or without service at nokia.letstalk.com or letstalk.com. E-coupon expires June 30, 2006 and may not be combined with any other NokiaUSA.com e-coupons. E-Coupons are nontransferable and valid only for phone, phone accessory or Nokia Gear purchases made at the Nokia USA web site, URL: http://www.nokiausa.com/.

§ WARNING: Use only batteries, chargers, and enhancements approved by Nokia for use with this particular model. The use of any other types may invalidate any approval or warranty, and may be dangerous.

The use of mobile phones and their accessories may be prohibited or restricted in certain areas. Obey applicable law regarding usage.

Copyright © 2006 Nokia. All rights reserved. Nokia, Nokia Connecting People, Xpress-on, Pop-Port, and all Nokia phone models are trademarks or registered trademarks of Nokia Corporation.

https://web.archive.org/web/20030413190333/http://www.irda.org/standards/specifications.asp

INTERNET ARCHIVE WayBackMachine
http://www.irda.org/standards/specifications.asp
Go FEB APR 13 JUN
78 captures
17 Apr 1999 - 21 May 2023
2002 2003 2004
About this capture

IrDA® Standard Specifications

Notice: The following documents are available at no charge both here and on the IrDA FTP site.

*By downloading these specifications, you acknowledge and agree that you or your organization will use the IrDA specifications for your own purposes only and will not copy, resell or otherwise distribute or make commercial use of IrDA Standard Specifications. IrDA Standard Specifications are protected by United States Copyright Law and no part of the standards may be reproduced or distributed without the prior written consent of the Infrared Data Association.

Each document is a separate file and must be downloaded. The documents are formatted in Adobe Acrobat (.PDF). You will need the appropriate viewer to process the documents. The documents are offered on a "as is basis" and comments are welcome. Please e-mail your comments to information@irda.org.

PUBLICATION CONTENTS

- NEW

- UPDATED

NAME	DESCRIPTION
IrDA® SIR Data Specification	IrDA® Serial Infrared Physical Layer Link Specification (IrPHY), IrDA Serial Infrared Link Access Protocol (IrLAP) and IrDA Serial Infrared Link Management Protocol (IrLMP), IrDA Tiny TP (1,331KB)and IrDA Point and Shoot Profile and Test Specification. Get specification. To obtain the updated IrPHY 1.4 ONLY, click here.
IrDA® Financial Messaging (IrFM™)Point and Pay Specification	IrFM™ Point and Pay specifications are now available. To obtain them click here!.
IrDA® Financial Messaging (IrFM™)Test Specification	IrFM™ Test Specifications are now available. To obtain them click here!.
IrDA® IrLAP Fast Connect Application Note	The IrLAP Fast Connect Application Note is now available. Get document!.
IrDA® Adapter Application Profile and Test Specification	IrDA® Adapter Application Profile and Test Specification. Get document.
IrDA® Control Specification	IrDA® Command and Control Ir Standard (Formerly IrBUS) (701KB). Get specification.
IrCOMM 1.0	IrDA® Infrared Communications Protocol (380KB). Get specification.
IrTinyTP 1.1	IrDA® Infrared Tiny Transport Protocol (82KB). Get specification.
IrLAN 1.0	IrDA® Infrared LAN Access Extensions for Link Management Protocol (92KB). Get specification.
OBEX™ 1.3	IrDA® Object Exchange Protocol (OBEX™)Ver.1.3. Get specification.
OBEX™ Test Specification 1.0.1	IrDA® Object Exchange Protocol (OBEX™)Test Specification Ver.1.0.1. Get specification.
IrDA Lite	IrDA® Minimal IrDA® Protocol Implementation (110KB). Get specification.

IrDA® Plug and Play	IrDA® Plug & Play Extensions to IrLMP 1.0 (80KB). Get specification.
IrMC	IrDA® Infrared Mobile Communications (1,020KB) See Press release. Get entire specification. iMelody is part of IrMC. To Get **iMelody ONLY** click here.
IrTran-P	IrDA® Infrared Transfer Picture Specifications (240KB). IrTranP provides image exchange protocol used in Digital Image capture devices/cameras. To see more about IrTran-P visit the IrTran-P website. Get specification.
IrDA® Dongle Interface	IrDA® Dongle Interface specifications (85KB). Get specification.
IrWW	IrDA® Infared for Wrist Watches specification, profile and test documents. Get specification.
Serial Port Profile, IrModem Profile and Test Specs	IrDA® Serial Port Profile, IrModem Profile and test documents. Get document.
Serial Interface for Transceivers	Recommended Serial Interface for Transceiver Control. Get specification.
IrDA® Point and Shoot	IrDA® Point and Shoot Application Profile. Get Profile. IrDA Point and Shoot Test Specification. Get Test specification.

INTERNET ARCHIVE WayBackMachine
http://www.bluetooth.com/dev/specifications.asp
Go NOV DEC APR
87 captures
12 Dec 2001 - 20 Oct 2022
2000 12 2001 2003
About this capture

SEARCH
FAQ'S

Bluetooth Wireless Solutions

Interoperable, usable, and competitively priced, Bluetooth wireless enabled products solve customer's problems - cable free.

DOWNLOADS:

More free specification downloads available to Bluetooth SIG members.
Become a member now.

- v.1.1 CORE
- v.1.1 PROFILES

the specification

THE BLUETOOTH WIRELESS SPECIFICATION IS A ROYALTY-FEE STANDARD THAT FUELS INNOVATION IN IDEAS AND ENABLES CREATIVE NEW PRODUCTS. LOW-COST AND POWER EFFICIENT, BLUETOOTH WIRELESS TECHNOLOGY PROVIDES NEW BUSINESS SOLUTIONS AND PRODUCT CAPABILITIES WORLDWIDE.

Unlike many other wireless standards, the Bluetooth wireless specification includes both link layer and application layer definitions for product developers which supports data, voice and content-centric applications. Radios that comply with the Bluetooth wireless specification operate in the unlicensed, 2.4 GHz radio spectrum ensuring communication compatibility worldwide. These radios use a spread spectrum, frequency hopping, full-duplex signal at up to 1600 hops/sec. The signal hops among 79 frequencies at 1 MHz intervals to give a high degree of interference immunity. Up to seven simultaneous connections can established and maintained.

The Bluetooth Specification contains the information necessary to ensure that diverse devices supporting the Bluetooth wireless technology can communicate with each other worldwide. The document is divided into two sections: a Core Specification (Volume I) and Profile Definitions (Volume II).

Core Specification
Requirements for components such as the radio, baseband, link manager, and service discovery protocol, transport layer, along with interoperability information regarding different communication protocols.

Profiles Definition
Higher-level protocols and procedures required to implement user level functions using Bluetooth wireless technology.

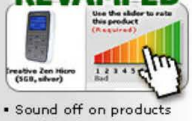
© 2001 BLUETOOTH ALL RIGHTS RESERVED

https://web.archive.org/web/20050328034057/http:/reviews.cnet.com/Sony_CLIE_PEG_TH55/4505-3127_7-30733801-3.html

INTERNET ARCHIVE WayBackMachine
http://reviews.cnet.com/Sony_CLIE_PEG_TH55/4505-3127_7-30733801-3.html
Go FEB MAR 28 2005 MAY
3 captures 28 Mar 2005 - 25 Aug 2005
2004 2006 About this capture

advertisement
Click Here.
advertisement

CNET user opinions REVAMPED



- Sound off on products
- Find the best opinions
- Rate other users

See what's new

sponsored
TOP PRODUCTS FROM TOP BRANDS
- Dell products
- IBM products
- HP products
- Sony products

Sony CLIE PEG-TH55

Design

At 4.9 by 3.0 by 0.6 inches, the elegant, dark-gray Sony CLIE PEG-TH55 has the solid feel of a serious handheld. The 3.8-inch, 320x480-pixel transflective TFT screen dominates the face of the device. Add in the Virtual Graffiti area, and there's plenty of room to view applications. The TH55 weighs 6.5 ounces, which isn't bad, considering it includes both Wi-Fi and a built-in camera. Its sturdy exterior feels as if it could endure some rough treatment. However, the translucent-gray plastic protective screen cover doesn't feel nearly as tough. And while it didn't suffer any scratches during our tests, it did show a lot of fingerprints. Plus, it feels like it could easily snap off its hinges if you weren't careful. The nice thing about the see-through cover, however, is that you can tell with a glance which alarm just rang. Sony includes a wrist strap with the PDA, but it does not include an extra protective case.

Visit com.com

A PDA with a view: The TH55's 320x480 display takes center stage.

Perhaps in an effort to accommodate the protective cover, the scrollwheel typically found below the screen on other CLIEs and handhelds has been relocated to the back of the device, above the digital camera (located at the top), making it difficult to use. We never felt comfortable with the new layout. And if you plan on playing games, steer clear of this handheld in favor of one with a more common directional keypad or joystick. Four traditional shortcut keys line the bottom edge of the PDA.

On the left side of the TH55 are the headphone jack (which accepts Walkman-style headphones), a Memory Stick slot, a photo-capture button, and three identical-feeling switches that control the power, the camera lens cover, and the voice recorder. Unfortunately, it is impossible to tell which switch you are using by relying on touch alone.

The TH55's case was apparently so packed that only a telescoping stylus could fit. Even when fully extended, the stylus is still fairly short, and it frequently collapsed to its miniature length while we were trying to use it.

To keep you in sync and powered up, the TH55 comes with a USB cable and a power cord, both of which connect through the dongle that clips to the bottom of the PDA. For frequent travelers, this is an ideal setup. Deskbound buyers should budget for the optional charging cradle, which is available from Sony for $30.

Next: Sony CLIE PEG-TH55 Features

Page: 1 | 2 | 3 | 4 | 5

User opinions for Sony CLIE PEG-TH55

Buying choices

for the Sony CLIE PEG-TH55

Compu America
$385.00 | In stock: Yes

Prices from 1 store $385

advertisement
EarthLink
Access for all your needs
High Speed
Dial-Up
Wireless

CNET AT WORK

Featherweight phones

Look sharp and stay in touch. These tiny mobile phones won't ruin the lines of your suit, and they won't overload you with features you'll never use.
Read more

The latest reviews

Dell Inspiron XPS 2 for business
Read review
Check prices
8.3 out of 10

advertisement
Click Here.

Show us your home office & WIN!

Submit your home-office setup to Show Us Yours, and you could win a Microsoft Optical Desktop Elite.

Business solutions

Ask the EDITORS
Use your cell phone overseas
Tablets vs. notebooks
Buy software on eBay?

TECH NIGHTMARES
Cell phone terrors
Outlook horror stories
Windows XP freak show

The Ultimate Office
Boss's office
Cubicle
Conference room

CNET At Work
The latest small-business content is always right here:
work.cnet.com

advertisement
EarthLink
Access for all your needs
High Speed
Dial-Up
Wireless

Help.com
ONLINE COURSES
Build Your Own PC
Digital Photography
PC Troubleshooting
Spam & Spyware
Wireless Home Theater
Enroll for FREE

COMPUTER SHOPPER
SUBSCRIBE & GET FREE SOFTWARE

CNET NEWSLETTERS
Click on a title below to
On-the-Go Weekly
AnchorDesk
Photo & Video Weekly
Digital Dispatch
SPECIAL OFFERS FROM OUR PARTNERS
Handheld Computers
Wireless Communication
SIGN UP NOW
Manage My Newsletters

Average user rating
Good
7.8
out of 10

Average user rating: from 77 users

How would you rate this product?

Write your own review

81 User opinions

Sort by: Rating | Date | Comments posted | Most helpful

User rating unavailable

Insufficant battery life (read more)
January 18, 2005

0 out of **1** users found this user opinion helpful.

User rating
Perfect
10
out of 10

it is the best pda ever (read more)
January 12, 2005

User rating
Poor
4
out of 10

Loved it until it stopped working (read more)
January 12, 2005

Read all user opinions

Basic specs for Sony CLIE PEG-TH55

Weight-- **6.5 oz**
Battery installed (max)-- **1 Lithium polymer**
Mfr estimated battery life-- **7.5 hour(s)**
Built-in devices-- **Camera, Display**
OS provided-- **Palm OS 5.2**
Installed RAM-- **32 MB**
Processor-- **Sony**
Input device type-- **Stylus, Jog Dial, Touch-screen**
Display type-- **TFT active matrix**
Wireless connectivity-- **IrDA, IEEE 802.11b**
Connectivity device(s) included-- **None**

See complete specs

Where to buy

Store	CNET Certified	In-stock?	Shipping	Price
Compu America	★★★★★ Write a store review	Yes	13.8625	**$385.00**

See **Sony CLIE PEG-TH55 prices** from 1 store

People who looked at this product also looked at

Sony CLIÉ PEG-UX50
October 10, 2003 lowest price: $398
Product review | Compare Prices | Add to my list

Dell Axim X30
May 17, 2004 lowest price: $349
Product review | Compare Prices | Add to my list

Sony CLIE PEG-TJ37
April 30, 2004 lowest price: $239
Product review | Compare Prices | Add to my list

PalmOne Zire 72
April 27, 2004 lowest price: $249
Product review | Compare Prices | Add to my list

Comparing this product to

Editor's top rated handheld devices Compare
Most popular handheld devices Compare

advertisement
Buy it here:
CDW
Anytime
Anywhere
Immediate
Cost Savings

IBM Go Wireless
$1078 w/ rebates
1.3GHz Intel® Celeron™ M
802.11b/g wireless
Netgear 802.11b/g Point

Toshiba A70
$1028* w/ rebate
3.06GHz Intel Pentium 4
60GB hard drive
Netgear 802.11b/g point

Sony VAIO B100
$778* w/rebates
1.5GHz Intel® Pentium® M
512MB, CD-RW/DVD
Netgear 802.11b/g point

IBM. Buy Now! TOSHIBA Buy Now! SONY Buy Now!

Tell us how we're doing

We welcome your feedback on our pages. We'd love to hear your opinion about the quality of our coverage, as well as any bugs you find on our site. Please select from the list or fill out the box below to share your two cents (we can't promise that we will answer all feedback, but we will read it all):

Your e-mail address:

Please mark as many of the following boxes that apply:

- [ ] Product review needs more info or is outdated
- [ ] Spec information is not complete or is outdated
- [ ] Pricing details are incorrect or not comprehensive
- [ ] The video on this page never loaded or played
- [ ] The page took forever to load or broke my browser

Comments:

Send suggestions

Print this review | E-mail to a friend | Add to my list

Dell Home PCs

advertisement

Up to $500 off select • Dell Home PCs. • Details.

Copyright ©1995-2004 CNET Networks, Inc. All rights reserved. **Privacy policy** | Terms of use | About CNET Networks | Jobs

EXHIBIT B

https://web.archive.org/web/20011217082445/http://www.bluetooth.com/pdf/Bluetooth_11_Specifications_Book.pdf

Specification of the Bluetooth System

Wireless connections made easy

Core

Bluetooth™

Version 1.1
February 22 2001

Revision History

The Revision History is shown in Appendix I on page 886

Contributors

The persons who contributed to this specification are listed in Appendix II on page 897.

Web Site

This specification can also be found on the Bluetooth website:
http://www.bluetooth.com

Disclaimer and Copyright Notice

The copyright in these specifications is owned by the Promoter Members of Bluetooth SIG, Inc. (“Bluetooth SIG”). Use of these specifications and any related intellectual property (collectively, the “Specification”), is governed by the Promoters Membership Agreement among the Promoter Members and Bluetooth SIG (the “Promoters Agreement”), certain membership agreements between Bluetooth SIG and its Adopter and Associate Members (the “Membership Agreements”) and the Bluetooth Specification Early Adopters Agreements (“1.2 Early Adopters Agreements”) among Early Adopter members of the unincorporated Bluetooth special interest group and the Promoter Members (the “Early Adopters Agreement”). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to Bluetooth SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of Bluetooth SIG or a party to an Early Adopters Agreement (each such person or party, a “Member”), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by aplicable law to Bluetooth SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WAR-

RANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the Bluetooth™ technology ("Bluetooth™ Products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of Bluetooth™ Products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Bluetooth™ Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Bluetooth™ Products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate and to adopt a process for adding new Bluetooth™ profiles after the release of the Specification.

Copyright © 1999, 2000, 2001

3Com Corporation,
Agere Systems, Inc.,
Ericsson Technology Licensing, AB,
IBM Corporation,
Intel Corporation,
Microsoft Corporation,
Motorola, Inc.,
Nokia Mobile Phones,
Toshiba Corporation

*Third-party brands and names are the property of their respective owners.

MASTER TABLE OF CONTENTS

For the Bluetooth Profiles, *See Volume 2.*

Part A

RADIO SPECIFICATION

Contents 17

1 Scope 18
2 Frequency Bands and Channel Arrangement 19
3 Transmitter Characteristics 20
4 Receiver Characteristics 24
5 Appendix A 28
6 Appendix B 31

Part B

BASEBAND SPECIFICATION

Contents 35

1 General Description 41
2 Physical Channel 43
3 Physical Links 45
4 Packets 47
5 Error Correction 67
6 Logical Channels 76
7 Data Whitening 78
8 Transmit/Receive Routines 79
9 Transmit/Receive Timing 85
10 Channel Control 93
11 Hop Selection 127
12 Bluetooth Audio 139
13 Bluetooth Addressing 143
14 Bluetooth Security 149
15 List of Figures 179
16 List of Tables 183

Part C

LINK MANAGER PROTOCOL

Contents 187

1 General 191
2 Format of LMP 193
3 The Procedure Rules and PDUs 195
4 Connection Establishment 234
5 Summary of PDUs 236
6 Test Modes 248
7 Error Handling 250
8 List of Figures 251
9 List of Tables 253

Part D

LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL SPECIFICATION

Contents 257

1 Introduction 259
2 General Operation 263
3 State Machine 268
4 Data Packet Format 282
5 Signalling 285
6 Configuration Parameter Options 299
7 Service Primitives 305
8 Summary 323
9 References 324
10 List of Figures 325
11 List of Tables 326
Terms and Abbreviations 327
Appendix A: Configuration MSCs 328
Appendix B: Implementation Guidelines 331

Part E

SERVICE DISCOVERY PROTOCOL (SDP)

Contents 335

1 Introduction 337
2 Overview 340
3 Data Representation 351
4 Protocol Description 354
5 Service Attribute Definitions 368
6 Appendix 379
Appendix A– Background Information 380
Appendix B – Example SDP Transactions 381

Part F:1

RFCOMM WITH TS 07.10

Contents 397

1 Introduction 399
2 RFCOMM Service Overview 401
3 Service Interface Description 405
4 TS 07.10 Subset Supported by RFCOMM 406
5 TS 07.10 Adaptations for RFCOMM 408
6 Flow Control 417
7 Interaction with Other Entities 421
8 References 424
9 Terms and Abbreviations 425

Part F:2

IrDA INTEROPERABILITY

Contents 429

1 Introduction 430
2 OBEX Object and Protocol 433
3 OBEX over RFCOMM 437
4 OBEX over TCP/IP 439
5 Bluetooth Application Profiles using OBEX 441
6 References 443
7 List of Acronyms and Abbreviations 444

Part F:3

TELEPHONY CONTROL PROTOCOL SPECIFICATION

Contents 447

1 General Description 451
2 Call Control (CC) 455
3 Group Management (GM) 465
4 Connectionless TCS (CL) 472
5 Supplementary Services (SS) 473
6 Message formats 476
7 Message coding 488
8 Message Error handling 504
9 Protocol Parameters 506
10 Bluetooth Standard Character Set 507
11 References 508
12 List of Figures 509
13 List of Tables 510
Appendix 1 - TCS Call States 511

Part F:4

INTEROPERABILITY REQUIREMENTS FOR BLUETOOTH AS A WAP BEARER

Contents

1 Introduction 517
2 The Use of WAP In the Bluetooth Environment 518
3 WAP Services Overview 520
4 WAP in the Bluetooth Piconet 524
5 Interoperability Requirements 529
6 Service Discovery 530
7 Link Manager 534
8 Generic Modes 535
9 References 536

Part H:1

HOST CONTROLLER INTERFACE FUNCTIONAL SPECIFICATION

Contents 541

1 Introduction 546
2 Overview of Host Controller Transport Layer 550
3 HCI Flow Control 551
4 HCI Commands 553
5 Events 727
6 List of Error Codes 769
7 List of Acronyms and Abbreviations 780
8 List of Figures 781
9 List of Tables 782

Part H:2

HCI USB TRANSPORT LAYER

Contents 785

1 Overview 786
2 USB Endpoint Expectations 788
3 Class Code 795
4 Device Firmware Upgrade 796
5 Limitations 797

Part H:3

HCI RS232 TRANSPORT LAYER

Contents 801

1 General 802
2 Overview 803
3 Negotiation Protocol 804
4 Packet Transfer Protocol 808
5 Using delimiters with COBS for synchronization 809
6 Using RTS/CTS for Synchronization 812
7 References 818

Part H:4

HCI UART TRANSPORT LAYER

Contents 823

1 General 824

2 Protocol 825

3 RS232 Settings 826

4 Error Recovery 827

Part I:1

BLUETOOTH TEST MODE

Contents 831

1 General Description 832

2 Test Scenarios 834

3 Outline of Proposed LMP Messages 843

4 References 846

Part I:2

BLUETOOTH COMPLIANCE REQUIREMENTS

Contents 849

1 Scope 850

2 Definitions 851

3 Legal Aspects 853

4 Introduction to the Bluetooth Qualification Program 854

Part I:3

TEST CONTROL INTERFACE

Contents 859

1 Introduction 861

2 General Description 863

3 Test Configurations 868

4 TCI-L2CAP Specification 870

5 Abbreviations 881

Appendix I

REVISION HISTORY 883

Appendix II

CONTRIBUTORS 895

Appendix III

ACRONYMS AND ABBREVIATIONS 907

Appendix IV

SAMPLE DATA

Contents 919

1 Encryption Sample Data 920

2 Frequency Hopping Sample Data—Mandatory Scheme...955

3 Access Code Sample Data.....968

4 HEC and Packet Header Sample Data 971

5 CRC Sample Data.....972

6 Complete Sample Packets 973

7 Whitening Sequence Sample Data 975

8 FEC Sample Data 978

9 Encryption Key Sample Data 979

Appendix V

BLUETOOTH AUDIO

Contents 1007

1 General Audio Recommendations 1008

Appendix VI

BASEBAND TIMERS

Contents 1015

1 Baseband Timers 1016

Appendix VII

OPTIONAL PAGING SCHEMES 1019

Contents 1021

1 General 1023

2 Optional Paging Scheme I 1024

Appendix VIII

BLUETOOTH ASSIGNED NUMBERS

The former appendix Bluetooth Assigned Numbers is now published as a separate document and the latest version is always available at: http://www.bluetooth.org/assigned-numbers.htm

Appendix IX

MESSAGE SEQUENCE CHARTS

Contents 1031

1 Introduction 1033
2 Services without connection request 1034
3 ACL connection establishment and detachment 1038
4 Optional activities after ACL Connection establishment... 1046
5 SCO Connection establishment and detachment 1055
6 Special modes: sniff, hold, park 1058
7 Buffer management, flow control 1064
8 Loopback Mode 1066
9 List of Acronyms and Abbreviations 1069
10 List of Figures 1070
11 List of Tables 1071
12 References 1072

Index

Alphabetical Index 1073

Part A

RADIO SPECIFICATION

CONTENTS

1 Scope 18

2 Frequency Bands and Channel Arrangement 19

3 Transmitter Characteristics 20

3.1 MODULATION CHARACTERISTICS 21

3.2 SPURIOUS EMISSIONS 22

3.2.1 In-band Spurious Emission 22

3.2.2 Out-of-Band Spurious Emission 23

3.3 RADIO FREQUENCY TOLERANCE 23

4 Receiver Characteristics 24

4.1 ACTUAL SENSITIVITY LEVEL 24

4.2 INTERFERENCE PERFORMANCE 24

4.3 OUT-OF-BAND BLOCKING 25

4.4 INTERMODULATION CHARACTERISTICS 25

4.5 MAXIMUM USABLE LEVEL 26

4.6 SPURIOUS EMISSIONS 26

4.7 RECEIVER SIGNAL STRENGTH INDICATOR (optional) 26

4.8 REFERENCE SIGNAL DEFINITION 27

5 Appendix A 28

6 Appendix B 31

1 SCOPE

The Bluetooth transceiver is operating in the 2.4 GHz ISM band. This specification defines the requirements for a Bluetooth transceiver operating in this unlicensed band.

Requirements are defined for two reasons:

- Provide compatibility between the radios used in the system
- Define the quality of the system

The Bluetooth transceiver shall fulfil the stated requirements under the operating conditions specified in Appendix A and Appendix B. The Radio parameters must be measured according to the methods described in the RFTest Specification.

This specification is based on the established regulations for Europe, Japan and North America. The standard documents listed below are only for information, and are subject to change or revision at any time.

Europe (except France and Spain)**:**

Approval Standards: European Telecommunications Standards Institute, ETSI
Documents: ETS 300-328, ETS 300-826
Approval Authority: National Type Approval Authorities

France:

Approval Standards: La Reglementation en France por les Equipements fonctionnant dans la bande de frequences 2.4 GHz "RLAN-Radio Local Area Network"
Documents: SP/DGPT/ATAS/23, ETS 300-328, ETS 300-826
Approval Authority: Direction Generale des Postes et Telecommunications
Note: A new R&TTE EU Directive will be in effect by March 2000, with consequent effects on the manufacturer's declaration of conformity and free circulation of products within the EU.

Spain:

Approval Standards: Supplemento Del Numero 164 Del Boletin Oficial Del Estado (Published 10 July 91, Revised 25 June 93)
Documents: ETS 300-328, ETS 300-826
Approval Authority: Cuadro Nacional De Atribucion De Frecuesias

Japan:

Approval Standards: Association of Radio Industries and Businesses, ARIB
Documents: RCR STD-33A
Approval Authority: Ministry of Post and Telecommunications, MPT
Note: The Japanese rules are in revision. Decisions on the revision will take place in Q2 1999.

North Americas:

Approval Standards: Federal Communications Commission, FCC, USA
Documents: CFR47, Part 15, Sections 15.205, 15.209, 15.247

Approval Standards: Industry Canada, IC, Canada
Documents: GL36

Approval Authority: FCC (USA), Industry Canada (Canada)

2 FREQUENCY BANDS AND CHANNEL ARRANGEMENT

The Bluetooth system is operating in the 2.4 GHz ISM (Industrial Scientific Medicine) band. In a vast majority of countries around the world the range of this frequency band is 2400 - 2483.5 MHz. Some countries have however national limitations in the frequency range. In order to comply with these national limitations, special frequency hopping algorithms have been specified for these countries. It should be noted that products implementing the reduced frequency band will not work with products implementing the full band. The products implementing the reduced frequency band must therefore be considered as local versions for a single market. The Bluetooth SIG has launched a campaign to overcome these difficulties and reach total harmonization of the frequency band.

Geography	Regulatory Range	RF Channels
USA, Europe and most other countries[1)]	2.400-2.4835 GHz	f=2402+k MHz, k=0,…,78

Table 2.1: Operating frequency bands

Note 1. The Bluetooth Specification includes a special frequency hopping pattern to provide provisions for compliance with national limitations like in France. The frequency range for France is 2.4465 - 2.4835 GHz and the corresponding RF channels are f = 2454 + k MHz, k= 0,...,22..

Channel spacing is 1 MHz. In order to comply with out-of-band regulations in each country, a guard band is used at the lower and upper band edge.

Geography	Lower Guard Band	Upper Guard Band
USA, Europe and most other countries	2 MHz	3.5 MHz

Table 2.2: Guard Bands

3 TRANSMITTER CHARACTERISTICS

The requirements stated in this section are given as power levels at the antenna connector of the equipment. If the equipment does not have a connector, a reference antenna with 0 dBi gain is assumed.

Due to difficulty in measurement accuracy in radiated measurements, it is preferred that systems with an integral antenna provide a temporary antenna connector during type approval.

If transmitting antennas of directional gain greater than 0 dBi are used, the applicable paragraphs in ETSI 300 328 and FCC part 15 must be compensated for.

The equipment is classified into three power classes.

Power Class	Maximum Output Power (Pmax)	Nominal Output Power	Minimum Output Power[1)]	Power Control
1	100 mW (20 dBm)	N/A	1 mW (0 dBm)	Pmin<+4 dBm to Pmax Optional: Pmin[2)] to Pmax
2	2.5 mW (4 dBm)	1 mW (0 dBm)	0.25 mW (-6 dBm)	Optional: Pmin[2)] to Pmax
3	1 mW (0 dBm)	N/A	N/A	Optional: Pmin[2)] to Pmax

Table 3.1: Power classes

Note 1. Minimum output power at maximum power setting.

Note 2. The lower power limit Pmin<-30dBm is suggested but is not mandatory, and may be chosen according to application needs.

A power control is required for power class 1 equipment. The power control is used for limiting the transmitted power over 0 dBm. Power control capability under 0 dBm is optional and could be used for optimizing the power consumption and overall interference level. The power steps shall form a monotonic sequence, with a maximum step size of 8 dB and a minimum step size of 2 dB. A class 1 equipment with a maximum transmit power of +20 dBm must be able to control its transmit power down to 4 dBm or less.

Equipment with power control capability optimizes the output power in a link with LMP commands (see Link Manager Protocol). It is done by measuring RSSI and report back if the power should be increased or decreased.

Note that power class 1 must not be used for sending packets from one device to another if the receiving side of a connection does not support the necessary messaging for power control of the sending side (i.e. RSSI measurements and

related messages). In this case, the transmitter should comply with the rules of a class 2 or class 3 transmitter.

Also note that if a class 1 device is paging or inquiring very close to another device, the input power could be larger than the requirement in 4.5 Maximum usable level. This can cause the listening device to fail to respond. It is therefore useful to page and inquire as well using transmission according to power class 2 or class 3.

3.1 MODULATION CHARACTERISTICS

The Modulation is GFSK (Gaussian Frequency Shift Keying) with a BT=0.5. The Modulation index must be between 0.28 and 0.35. A binary one is represented by a positive frequency deviation, and a binary zero is represented by a negative frequency deviation. The symbol timing shall be better than ±20 ppm.

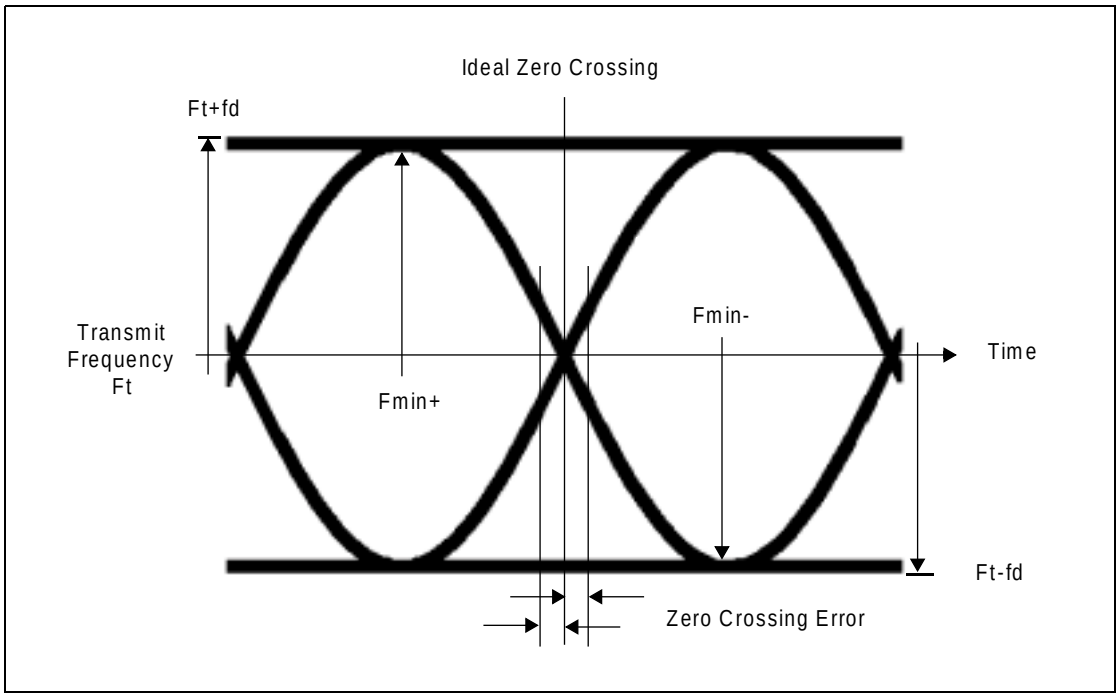


Figure 3.1: Figure 3-1 Actual transmit modulation.

For each transmit channel, the minimum frequency deviation (Fmin = the lesser of {Fmin+, Fmin-}) which corresponds to 1010 sequence shall be no smaller than ±80% of the frequency deviation (fd) which corresponds to a 00001111 sequence.
In addition, the minimum deviation shall never be smaller than 115 kHz. The data transmitted has a symbol rate of 1 Ms/s.

The zero crossing error is the time difference between the ideal symbol period and the measured crossing time. This shall be less than ± 1/8 of a symbol period.

3.2 SPURIOUS EMISSIONS

The spurious emission, in-band and out-of-band, is measured with a frequency hopping transmitter hopping on a single frequency; this means that the synthesizer must change frequency between receive slot and transmit slot, but always returns to the same transmit frequency.

For the USA, FCC parts 15.247, 15.249, 15.205 and 15.209 are applicable regulations. For Japan, RCR STD-33 applies and, for Europe, ETSI 300 328.

3.2.1 In-band Spurious Emission

Within the ISM band the transmitter shall pass a spectrum mask, given in Table 3.2. The spectrum must comply with the FCC's 20-dB bandwidth definition and should be measured accordingly. In addition to the FCC requirement an adjacent channel power on adjacent channels with a difference in channel number of two or greater an adjacent channel power is defined. This adjacent channel power is defined as the sum of the measured power in a 1 MHz channel. The transmitted power shall be measured in a 100 kHz bandwidth using maximum hold. The transmitter is transmitting on channel M and the adjacent channel power is measured on channel number N. The transmitter is sending a pseudo random data pattern throughout the test.

Frequency offset	Transmit Power
± 500 kHz	-20 dBc
|M-N| = 2	-20 dBm
|M-N| ≥ 3	-40 dBm

Table 3.2: Transmit Spectrum mask.

Note: If the output power is less than 0dBm then, wherever appropriate, the FCC's 20 dB relative requirement overrules the absolute adjacent channel power requirement stated in the above table.

Exceptions are allowed in up to three bands of 1 MHz width centered on a frequency which is an integer multiple of 1 MHz. They must, however, comply with an absolute value of –20 dBm.

3.2.2 Out-of-Band Spurious Emission

The measured power should be measured in a 100 kHz bandwidth.

Frequency Band	Operation mode	Idle mode
30 MHz - 1 GHz	-36 dBm	-57 dBm
1 GHz – 12.75 GHz	-30 dBm	-47 dBm
1.8 GHz – 1.9 GHz	-47 dBm	-47 dBm
5.15 GHz – 5.3 GHz	-47 dBm	-47 dBm

Table 3.3: Out-of-band spurious emission requirement

3.3 RADIO FREQUENCY TOLERANCE

The transmitted initial center frequency accuracy must be ±75 kHz from F_c. The initial frequency accuracy is defined as being the frequency accuracy before any information is transmitted. Note that the frequency drift requirement is not included in the ±75 kHz.

The transmitter center frequency drift in a packet is specified in Table 3.4. The different packets are defined in the Baseband Specification.

Type of Packet	Frequency Drift
One-slot packet	±25 kHz
Three-slot packet	±40 kHz
Five-slot packet	±40 kHz
Maximum drift rate[1)]	400 Hz/µs

Table 3.4: Frequency drift in a package

Note 1. The maximum drift rate is allowed anywhere in a packet.

4 RECEIVER CHARACTERISTICS

In order to measure the bit error rate performance; the equipment must have a “loop back” facility. The equipment sends back the decoded information. This facility is specified in the Test Mode Specification.

The reference sensitivity level referred to in this chapter equals -70 dBm.

4.1 ACTUAL SENSITIVITY LEVEL

The actual sensitivity level is defined as the input level for which a raw bit error rate (BER) of 0.1% is met. The requirement for a Bluetooth receiver is an actual sensitivity level of –70 dBm or better. The receiver must achieve the –70 dBm sensitivity level with any Bluetooth transmitter compliant to the transmitter specification specified in Section 3 on page 21.

4.2 INTERFERENCE PERFORMANCE

The interference performance on Co-channel and adjacent 1 MHz and 2 MHz are measured with the wanted signal 10 dB over the reference sensitivity level. On all other frequencies the wanted signal shall be 3 dB over the reference sensitivity level. Should the frequency of an interfering signal lie outside of the band 2400-2497 MHz, the out-of-band blocking specification (see Section 4.3 on page 26) shall apply. The interfering signal shall be Bluetooth-modulated (see section 4.8 on page 28). The BER shall be ≤ 0.1%. The signal to interference ratio shall be:

Requirement	Ratio
Co-Channel interference, $C/I_{co\text{-}channel}$	11 dB[1)]
Adjacent (1 MHz) interference, C/I_{1MHz}	0 dB[1]
Adjacent (2 MHz) interference, C/I_{2MHz}	-30 dB
Adjacent (≥3 MHz) interference, $C/I_{\geq 3MHz}$	-40 dB
Image frequency Interference[2) 3)], C/I_{Image}	-9 dB[1]
Adjacent (1 MHz) interference to in-band image frequency, $C/I_{Image\pm 1MHz}$	-20 dB[1]

Table 4.1: Interference performance

Note 1. These specifications are tentative and will be fixed within 18 months after the release of the Bluetooth specification version 1.0. Implementations have to fulfil the final specification after a 3-years’ convergence period starting at the release of the Bluetooth specification version 1.0. During the convergence period, devices need to achieve a co-channel interference resistance of +14 dB, an ACI (@1MHz) resistance of +4 dB, Image frequency interference resistance of –6 dB and an ACI to in-band image frequency resistance of –16 dB.

Note 2. In-band image frequency

Note 3. If the image frequency ≠ n*1 MHz, than the image reference frequency is defined as the closest n*1 MHz frequency.

Note 4. If two adjacent channel specifications from Table 4.1 are applicable to the same channel, the more relaxed specification applies.

These specifications are only to be tested at nominal temperature conditions with a receiver hopping on one frequency, meaning that the synthesizer must change frequency between receive slot and transmit slot, but always return to the same receive frequency.

Frequencies where the requirements are not met are called spurious response frequencies. Five spurious response frequencies are allowed at frequencies with a distance of ≥2 MHz from the wanted signal. On these spurious response frequencies a relaxed interference requirement C/I = -17 dB shall be met.

4.3 OUT-OF-BAND BLOCKING

The Out of band blocking is measured with the wanted signal 3 dB over the reference sensitivity level. The interfering signal shall be a continuous wave signal. The BER shall be ≤ 0.1%. The Out of band blocking shall fulfil the following requirements:

Interfering Signal Frequency	Interfering Signal Power Level
30 MHz - 2000 MHz	-10 dBm
2000 - 2399 MHz	-27 dBm
2498 – 3000 MHz	-27 dBm
3000 MHz – 12.75 GHz	-10 dBm

Table 4.2: Out of Band blocking requirements

24 exceptions are permitted which are dependent upon the given receive channel frequency and are centered at a frequency which is an integer multiple of 1 MHz. At 19 of these spurious response frequencies a relaxed power level -50 dBm of the interferer may used to achieve a BER of 0.1%. At the remaining 5 spurious response frequencies the power level is arbitrary.

4.4 INTERMODULATION CHARACTERISTICS

The reference sensitivity performance, BER = 0.1%, shall be met under the following conditions.

- The wanted signal at frequency f_0 with a power level 6 dB over the reference sensitivity level.
- A static sine wave signal at f_1 with a power level of –39 dBm
- A Bluetooth modulated signal (see Section 4.8 on page 28) at f_2 with a power level of -39 dBm

Such that $f_0=2f_1-f_2$ and $|f_2-f_1|=n*1$ MHz, where n can be 3, 4, or 5. The system must fulfil one of the three alternatives.

4.5 MAXIMUM USABLE LEVEL

The maximum usable input level the receiver shall operate at shall be better than –20 dBm. The BER shall be less or equal to 0,1% at –20* dBm input power.

4.6 SPURIOUS EMISSIONS

The spurious emission for a Bluetooth receiver shall not be more than:

Frequency Band	Requirement
30 MHz - 1 GHz	-57 dBm
1 GHz – 12.75 GHz	-47 dBm

Table 4.3: Out-of-band spurious emission

The measured power should be measured in a 100 kHz bandwidth.

4.7 RECEIVER SIGNAL STRENGTH INDICATOR (OPTIONAL)

A transceiver that wishes to support power-controlled links must be able to measure the strength of the received signal and determine if the transmitter on the other side of the link should increase or decrease its output power level. A Receiver Signal Strength Indicator (RSSI) makes this possible.

The RSSI measurement compares the received signal power with two threshold levels, which define the Golden Receive Power Range. The lower threshold level corresponds to a received power between -56 dBm and 6 dB above the actual sensitivity of the receiver. The upper threshold level is 20 dB above the lower threshold level to an accuracy of +/- 6 dB (see Figure 4.1 on page 27).

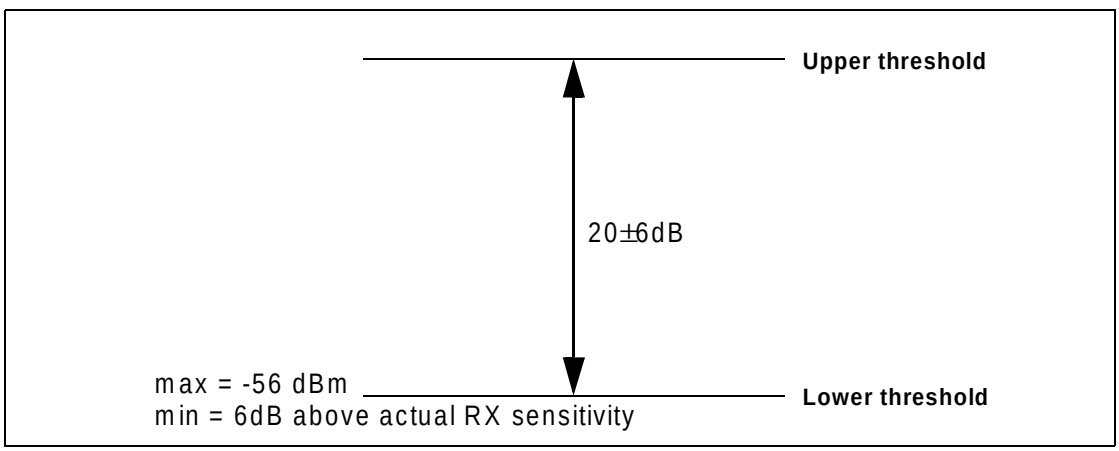


Figure 4.1: RSSI dynamic range and accuracy

4.8 REFERENCE SIGNAL DEFINITION

A Bluetooth modulated interfering signal is defined as:

Modulation = GFSK

Modulation index = 0.32±1%

BT= 0.5±1%

Bit Rate = 1 Mbps ±1 ppm

Modulating Data for wanted signal = PRBS9

Modulating Data for interfering signal = PRBS 15

Frequency accuracy better than ±1 ppm.

5 APPENDIX A

5.1 NOMINAL TEST CONDITIONS (NTC)

5.1.1 Nominal temperature

The nominal temperature conditions for tests shall be +15 to +35 ^{o}C. When it is impractical to carry out the test under this condition a note to this effect, stating the ambient temperature, shall be recorded. The actual value during the test shall be recorded in the test report.

5.1.2 Nominal Power source

5.1.2.1 Mains Voltage

The nominal test voltage for equipment to be connected to the mains shall be the nominal mains voltage. The nominal voltage shall be declared voltage or any of the declared voltages for which the equipment was designed. The frequency of the test power source corresponding to the AC mains shall be within 2% of the nominal frequency.

5.1.2.2 Lead-acid battery power sources used in vehicles

When radio equipment is intended for operation from the alternator-fed lead-acid battery power sources which are standard in vehicles, then the nominal test voltage shall be 1.1 times the nominal voltage of the battery (6V, 12V, etc.).

5.1.2.3 Other power sources

For operation from other power sources or types of battery (primary or secondary), the nominal test voltage shall be as declared by the equipment manufacturer. This shall be recorded in the test report.

5.2 EXTREME TEST CONDITIONS

5.2.1 Extreme temperatures

The extreme temperature range is defined as the largest temperature range given by the combination of:

- · The minimum temperature range 0 °C to +35 °C
- · The product operating temperature range declared by the manufacturer.

This extreme temperature range and the declared operating temperature range shall be recorded in the test report.

5.2.2 Extreme power source voltages

Tests at extreme power source voltages specified below are not required when the equipment under test is designed for operation as part of and powered by another system or piece of equipment. Where this is the case, the limit values of the host system or host equipment shall apply. The appropriate limit values shall be declared by the manufacturer and recorded in the test report.

5.2.2.1 Mains voltage

The extreme test voltage for equipment to be connected to an AC mains source shall be the nominal mains voltage ±10%.

5.2.2.2 Lead-acid battery power source used on vehicles

When radio equipment is intended for operation from the alternator-fed lead-acid battery power sources which are standard in vehicles, then extreme test voltage shall be 1.3 and 0.9 times the nominal voltage of the battery (6V, 12V etc.)

5.2.2.3 Power sources using other types of batteries

The lower extreme test voltage for equipment with power sources using the following types of battery, shall be

a) for Leclanché, alkaline, or lithium type battery: 0.85 times the nominal voltage of the battery

b) for the mercury or nickel-cadmium types of battery: 0.9 times the nominal voltage of the battery.

In both cases, the upper extreme test voltage shall be 1.15 times the nominal voltage of the battery.

5.2.2.4 Other power sources

For equipment using other power sources, or capable of being operated from a variety of power sources (primary or secondary), the extreme test voltages shall be those declared by the manufacturer. These shall be recorded in the test report.

6 APPENDIX B

The Radio parameters shall be tested in the following conditions

Parameter	Temperature	Power source
Output Power	ETC	ETC
Power control	NTC	NTC
Modulation index	ETC	ETC
Initial Carrier Frequency accuracy	ETC	ETC
Carrier Frequency drift	ETC	ETC
In-band spurious emissions	ETC	ETC
Out-of-band Spurious Emissions	ETC	ETC
Sensitivity	ETC	ETC
Interference Performance	NTC	NTC
Intermodulation Characteristics	NTC	NTC
Out-of-band blocking	NTC	NTC
Maximum Usable Level	NTC	NTC
Receiver Signal Strength Indicator	NTC	NTC

ETC = Extreme Test Conditions
NTC = Nominal Test Conditions

Part B

BASEBAND SPECIFICATION

This document describes the specifications of the Bluetooth link controller which carries out the baseband protocols and other low-level link routines.

CONTENTS

1 General Description 41

2 Physical Channel 43

2.1 Channel Definition 43

2.2 Time Slots 43

3 Physical Links 45

3.1 General 45

3.2 SCO Link 45

3.3 ACL Link 46

4 Packets 47

4.1 General Format 47

4.2 Access Code 48

4.2.1 Access code types 48

4.2.2 Preamble 49

4.2.3 Sync Word 49

4.2.4 Trailer 50

4.3 Packet Header 51

4.3.1 AM_ADDR 51

4.3.2 TYPE 51

4.3.3 FLOW 52

4.3.4 ARQN 52

4.3.5 SEQN 52

4.3.6 HEC 53

4.4 Packet Types 54

4.4.1 Common packet types 55

4.4.1.1 ID packet 55

4.4.1.2 NULL packet 55

4.4.1.3 POLL packet 55

4.4.1.4 FHS packet 56

4.4.1.5 DM1 packet 58

4.4.2 SCO packets 58

4.4.2.1 HV1 packet 58

4.4.2.2 HV2 packet 59

4.4.2.3 HV3 packet 59

4.4.2.4 DV packet 59

4.4.3 ACL packets 60
4.4.3.1 DM1 packet 60
4.4.3.2 DH1 packet 60
4.4.3.3 DM3 packet 60
4.4.3.4 DH3 packet 61
4.4.3.5 DM5 packet 61
4.4.3.6 DH5 packet 61
4.4.3.7 AUX1 packet 61
4.5 Payload Format 62
4.5.1 Voice field 62
4.5.2 Data field 62
4.6 Packet Summary 65
5 Error Correction 67
5.1 FEC Code: Rate 1/3 67
5.2 FEC Code: Rate 2/3 67
5.3 ARQ Scheme 68
5.3.1 Unnumbered ARQ 68
5.3.2 Retransmit filtering 70
5.3.3 Flushing payloads 71
5.3.4 Multi-slave considerations 71
5.3.5 Broadcast packets 72
5.4 Error Checking 73
6 Logical Channels 76
6.1 LC Channel (Link Control) 76
6.2 LM Channel (Link Manager) 76
6.3 UA/UI Channel (User Asynchronous/Isochronous data) 76
6.4 US Channel (User Synchronous data) 77
6.5 Channel Mapping 77
7 Data Whitening 78
8 Transmit/Receive Routines 79
8.1 TX Routine 79
8.1.1 ACL traffic 80
8.1.2 SCO traffic 81
8.1.3 Mixed data/voice traffic 81
8.1.4 Default packet types 82
8.2 RX Routine 82
8.3 Flow Control 83
8.3.1 Destination control 83
8.3.2 Source control 83
8.4 Bitstream Processes 84

9 Transmit/Receive Timing 85
9.1 Master/Slave Timing Synchronization 85
9.2 Connection State 86
9.3 Return From Hold Mode 88
9.4 Park and sniff Modes Wake-up 89
9.5 Page State 89
9.6 FHS Packet 90
9.7 Multi-slave Operation 91
10 Channel Control 93
10.1 Scope 93
10.2 Master-Slave Definition 93
10.3 Bluetooth Clock 93
10.4 Overview of States 95
10.5 Standby State 96
10.6 Access Procedures 97
10.6.1 General 97
10.6.2 Page scan 97
10.6.3 Page 99
10.6.4 Page response procedures 101
10.6.4.1 Slave response 103
10.6.4.2 Master response 105
10.7 Inquiry Procedures 106
10.7.1 General 106
10.7.2 Inquiry scan 107
10.7.3 Inquiry 108
10.7.4 Inquiry response 109
10.8 Connection State 111
10.8.1 Active mode 111
10.8.2 Sniff mode 112
10.8.3 Hold mode 113
10.8.4 Park mode 113
10.8.4.1 Beacon channel 114
10.8.4.2 Beacon access window 115
10.8.4.3 Parked slave synchronization 117
10.8.4.4 Parking 118
10.8.4.5 Master-activated unparking 118
10.8.4.6 Slave-activated unparking 119
10.8.4.7 Broadcast scan window 119
10.8.5 Polling schemes 120
10.8.5.1 Polling in active mode 120
10.8.5.2 Polling in park mode 120

10.8.6 Slot reservation scheme 120
10.8.7 Broadcast scheme 120
10.9 Scatternet 121
10.9.1 General 121
10.9.2 Inter-piconet communications 121
10.9.3 Master-slave switch 122
10.10 Power Management 125
10.10.1 Packet handling 125
10.10.2 Slot occupancy 125
10.10.3 Low-power modes 125
10.11 Link Supervision 125
11 Hop Selection 127
11.1 General Selection Scheme 127
11.2 Selection Kernel 129
11.2.1 First addition operation 130
11.2.2 XOR operation 130
11.2.3 Permutation operation 131
11.2.4 Second addition operation 133
11.2.5 Register bank 133
11.3 Control Word 133
11.3.1 Page scan and Inquiry scan substates 135
11.3.2 Page substate 135
11.3.3 Page response 136
11.3.3.1 Slave response 136
11.3.3.2 Master response 137
11.3.4 Inquiry substate 137
11.3.5 Inquiry response 138
11.3.6 Connection state 138
12 Bluetooth Audio 139
12.1 LOG PCM CODEC 139
12.2 CVSD CODEC 139
12.3 Error Handling 142
12.4 General Audio Requirements 142
12.4.1 Signal levels 142
12.4.2 CVSD audio quality 142

13 Bluetooth Addressing 143
13.1 Bluetooth Device Address (BD_ADDR) 143
13.2 Access Codes 143
13.2.1 Synchronization word definition 144
13.2.2 Pseudo-random noise sequence generation 146
13.2.3 Reserved addresses for GIAC and DIAC 147
13.3 Active Member Address (AM_ADDR) 147
13.4 Parked Member Address (PM_ADDR) 148
13.5 Access Request Address (AR_ADDR) 148

14 Bluetooth Security 149
14.1 Random Number Generation 150
14.2 Key Management 151
14.2.1 Key types 151
14.2.2 Key generation and initialization 153
14.2.2.1 Generation of initialization key, ... 154
14.2.2.2 Authentication 154
14.2.2.3 Generation of a unit key 154
14.2.2.4 Generation of a combination key 155
14.2.2.5 Generating the encryption key 156
14.2.2.6 Point-to-multipoint configuration 157
14.2.2.7 Modifying the link keys 157
14.2.2.8 Generating a master key 158
14.3 Encryption 159
14.3.1 Encryption key size negotiation 160
14.3.2 Encryption modes 161
14.3.3 Encryption concept 161
14.3.4 Encryption algorithm 163
14.3.4.1 The operation of the cipher 165
14.3.5 LFSR initialization 165
14.3.6 Key stream sequence 168
14.4 Authentication 169
14.4.1 Repeated attempts 170
14.5 The Authentication And Key-Generating Functions 171
14.5.1 The authentication function E1 171
14.5.2 The functions Ar and A’r 173
14.5.2.1 The round computations 173
14.5.2.2 The substitution boxes “e” and “l” 174
14.5.2.3 Key scheduling 175
14.5.3 E2-Key generation function for authentication 175
14.5.4 E3-Key generation function for encryption 177

15 List of Figures 179

16 List of Tables 183

1 GENERAL DESCRIPTION

Bluetooth is a short-range radio link intended to replace the cable(s) connecting portable and/or fixed electronic devices. Key features are robustness, low complexity, low power, and low cost.

Bluetooth operates in the unlicensed ISM band at 2.4 GHz. A frequency hop transceiver is applied to combat interference and fading. A shaped, binary FM modulation is applied to minimize transceiver complexity. The symbol rate is 1 Ms/s. A slotted channel is applied with a nominal slot length of 625 µs. For full duplex transmission, a Time-Division Duplex (TDD) scheme is used. On the channel, information is exchanged through packets. Each packet is transmitted on a different hop frequency. A packet nominally covers a single slot, but can be extended to cover up to five slots.

The Bluetooth protocol uses a combination of circuit and packet switching. Slots can be reserved for synchronous packets. Bluetooth can support an asynchronous data channel, up to three simultaneous synchronous voice channels, or a channel which simultaneously supports asynchronous data and synchronous voice. Each voice channel supports a 64 kb/s synchronous (voice) channel in each direction. The asynchronous channel can support maximal 723.2 kb/s asymmetric (and still up to 57.6 kb/s in the return direction), or 433.9 kb/s symmetric.

The Bluetooth system consists of a radio unit (see Radio Specification), a link control unit, and a support unit for link management and host terminal interface functions, see Figure 1.1 on page 41. The current document describes the specifications of the Bluetooth link controller, which carries out the baseband protocols and other low-level link routines. Link layer messages for link set-up and control are defined in the Link Manager Protocol on page 183.

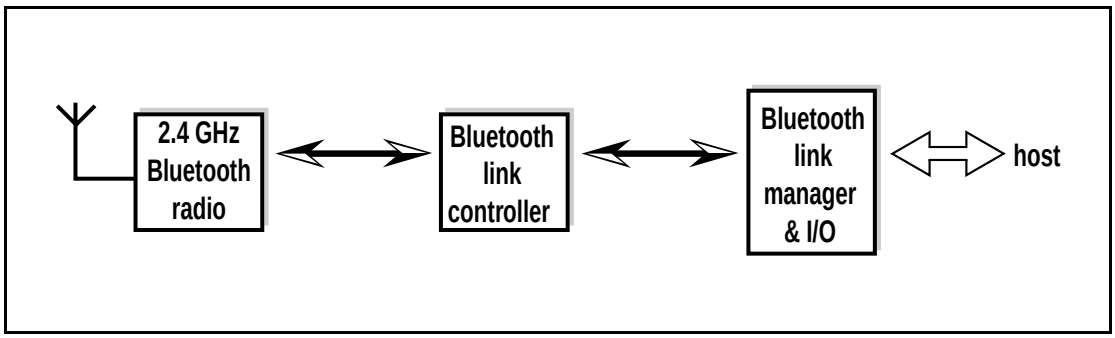


Figure 1.1: Different functional blocks in the Bluetooth system

The Bluetooth system provides a point-to-point connection (only two Bluetooth units involved), or a point-to-multipoint connection, see Figure 1.2 on page 42. In the point-to-multipoint connection, the channel is shared among several Bluetooth units. Two or more units sharing the same channel form a ***piconet***. One Bluetooth unit acts as the master of the piconet, whereas the other unit(s)

acts as slave(s). Up to seven slaves can be active in the piconet. In addition, many more slaves can remain locked to the master in a so-called parked state. These parked slaves cannot be active on the channel, but remain synchronized to the master. Both for active and parked slaves, the channel access is controlled by the master.

Multiple piconets with overlapping coverage areas form a ***scatternet***. Each piconet can only have a single master. However, slaves can participate in different piconets on a time-division multiplex basis. In addition, a master in one piconet can be a slave in another piconet. The piconets shall not be frequency-synchronized. Each piconet has its own hopping channel.

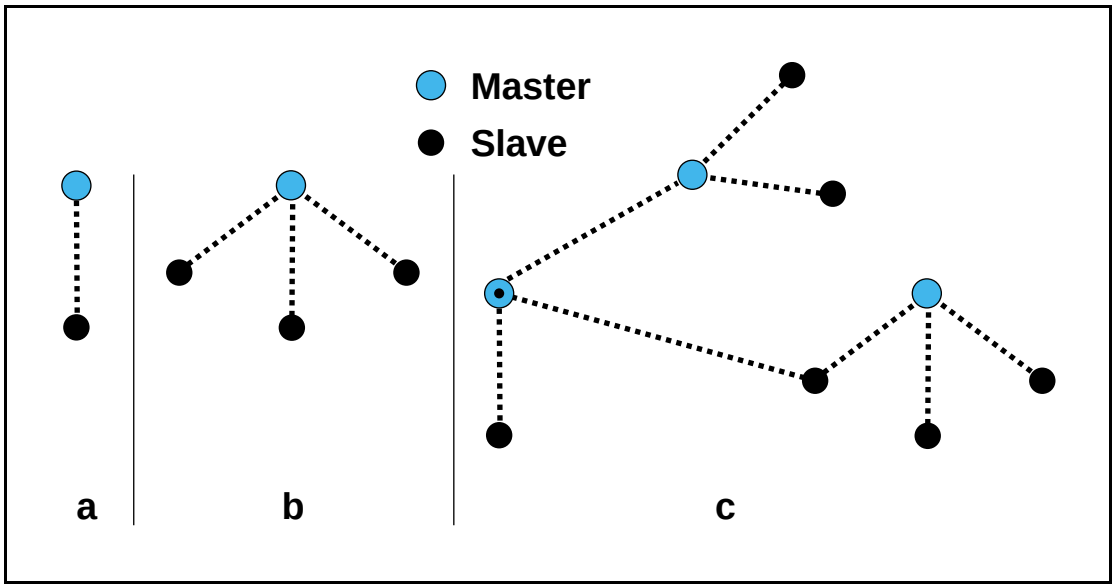


Figure 1.2: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c).

2 PHYSICAL CHANNEL

2.1 CHANNEL DEFINITION

The channel is represented by a pseudo-random hopping sequence hopping through the 79 or 23 RF channels. The hopping sequence is unique for the piconet and is determined by the Bluetooth device address of the master; the phase in the hopping sequence is determined by the Bluetooth clock of the master. The channel is divided into time slots where each slot corresponds to an RF hop frequency. Consecutive hops correspond to different RF hop frequencies. The nominal hop rate is 1600 hops/s. All Bluetooth units participating in the piconet are time- and hop-synchronized to the channel.

2.2 TIME SLOTS

The channel is divided into time slots, each 625 µs in length. The time slots are numbered according to the Bluetooth clock of the piconet master. The slot numbering ranges from 0 to 2^{27}-1 and is cyclic with a cycle length of 2^{27}.

In the time slots, master and slave can transmit packets.

A TDD scheme is used where master and slave alternatively transmit, see Figure 2.1 on page 44. The master shall start its transmission in even-numbered time slots only, and the slave shall start its transmission in odd-numbered time slots only. The packet start shall be aligned with the slot start. Packets transmitted by the master or the slave may extend over up to five time slots.

The RF hop frequency shall remain fixed for the duration of the packet. For a single packet, the RF hop frequency to be used is derived from the current Bluetooth clock value. For a multi-slot packet, the RF hop frequency to be used for the entire packet is derived from the Bluetooth clock value in the first slot of the packet. The RF hop frequency in the first slot after a multi-slot packet shall use the frequency as determined by the current Bluetooth clock value. Figure 2.2 on page 44 illustrates the hop definition on single- and multi-slot packets. If a packet occupies more than one time slot, the hop frequency applied shall be the hop frequency as applied in the time slot where the packet transmission was started.

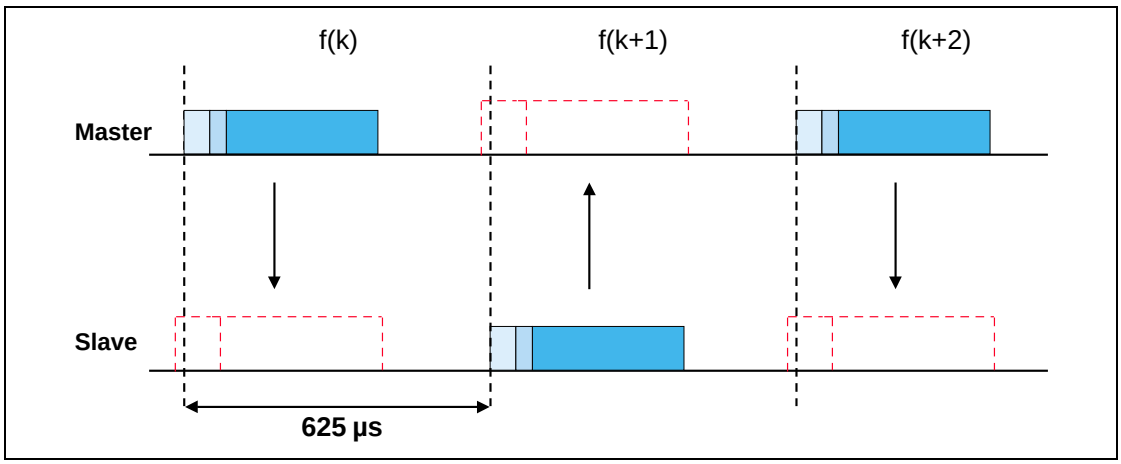


Figure 2.1: TDD and timing

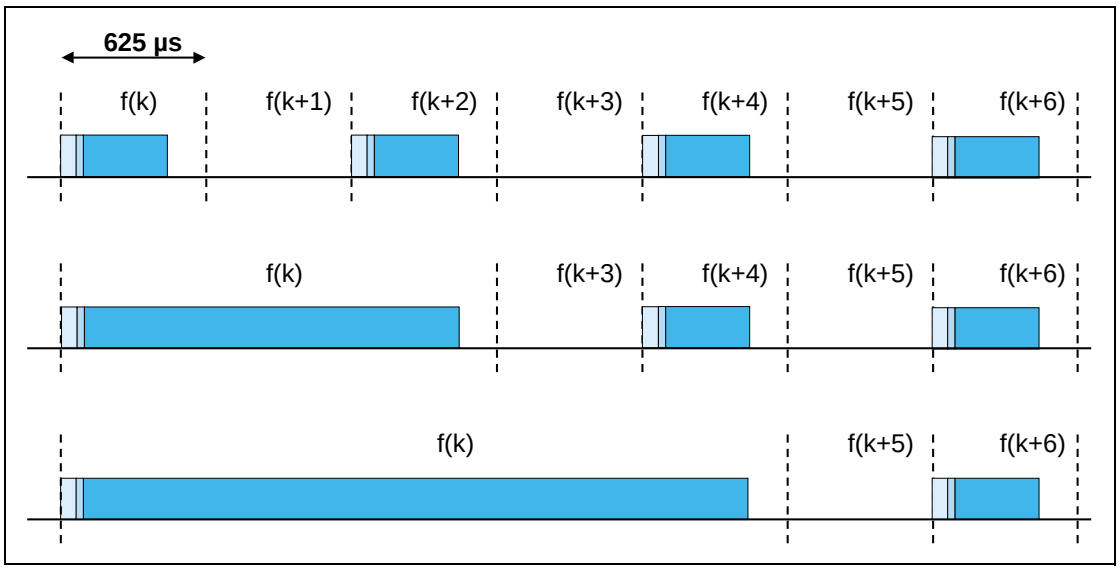


Figure 2.2: Multi-slot packets

3 PHYSICAL LINKS

3.1 GENERAL

Between master and slave(s), different types of links can be established. Two link types have been defined:

- Synchronous Connection-Oriented (SCO) link
- Asynchronous Connection-Less (ACL) link

The SCO link is a point-to-point link between a master and a single slave in the piconet. The master maintains the SCO link by using reserved slots at regular intervals. The ACL link is a point-to-multipoint link between the master and all the slaves participating on the piconet. In the slots not reserved for the SCO link(s), the master can establish an ACL link on a per-slot basis to any slave, including the slave(s) already engaged in an SCO link.

3.2 SCO LINK

The SCO link is a symmetric, point-to-point link between the master and a specific slave. The SCO link reserves slots and can therefore be considered as a circuit-switched connection between the master and the slave. The SCO link typically supports time-bounded information like voice. The master can support up to three SCO links to the same slave or to different slaves. A slave can support up to three SCO links from the same master, or two SCO links if the links originate from different masters. SCO packets are never retransmitted.

The master will send SCO packets at regular intervals, the so-called SCO interval T_{SCO} (counted in slots) to the slave in the reserved master-to-slave slots. The SCO slave is always allowed to respond with an SCO packet in the following slave-to-master slot unless a different slave was addressed in the previous master-to-slave slot. If the SCO slave fails to decode the slave address in the packet header, it is still allowed to return an SCO packet in the reserved SCO slot.

The SCO link is established by the master sending an SCO setup message via the LM protocol. This message will contain timing parameters such as the SCO interval T_{SCO} and the offset D_{SCO} to specify the reserved slots.

In order to prevent clock wrap-around problems, an initialization flag in the LMP setup message indicates whether initialization procedure 1 or 2 is being used. The slave shall apply the initialization method as indicated by the initialization flag. The master uses initialization 1 when the MSB of the current master clock (CLK_{27}) is 0; it uses initialization 2 when the MSB of the current master clock (CLK_{27}) is 1. The master-to-slave SCO slots reserved by the master and the slave shall be initialized on the slots for which the clock satisfies the following equation:

$CLK_{27-1} \bmod T_{SCO} = D_{SCO}$ for initialization 1

$(\overline{CLK_{27}}, CLK_{26-1}) \bmod T_{SCO} = D_{SCO}$ for initialization 2

The slave-to-master SCO slots shall directly follow the reserved master-to-slave SCO slots. After initialization, the clock value CLK(k+1) for the next master-to-slave SCO slot is found by adding the fixed interval T_{SCO} to the clock value of the current master-to-slave SCO slot:

$$CLK(k+1) = CLK(k) + T_{SCO}$$

3.3 ACL LINK

In the slots not reserved for SCO links, the master can exchange packets with any slave on a per-slot basis. The ACL link provides a packet-switched connection between the master and all active slaves participating in the piconet. Both asynchronous and isochronous services are supported. Between a master and a slave only a single ACL link can exist. For most ACL packets, packet retransmission is applied to assure data integrity.

A slave is permitted to return an ACL packet in the slave-to-master slot if and only if it has been addressed in the preceding master-to-slave slot. If the slave fails to decode the slave address in the packet header, it is not allowed to transmit.

ACL packets not addressed to a specific slave are considered as broadcast packets and are read by every slave. If there is no data to be sent on the ACL link and no polling is required, no transmission shall take place.

4 PACKETS

4.1 GENERAL FORMAT

The bit ordering when defining packets and messages in the *Baseband Specification*, follows the *Little Endian format*, i.e., the following rules apply:

- The *least significant bit* (LSB) corresponds to b_0;
- The LSB is the first bit sent over the air;
- In illustrations, the LSB is shown on the left side;

The linkcontroller interprets the first bit arriving from a higher software layer as b_0; i.e. this is the first bit to be sent over the air. Furthermore, data fields generated internally at baseband level, such as the packet header fields and payload header length, are transmitted with the LSB first. For instance, a 3-bit parameter X=3 is sent as $b_0 b_1 b_2$= 110 where 1 is sent first and 0 is sent last.

The data on the piconet channel is conveyed in packets. The general packet format is shown in Figure 4.1 on page 47. Each packet consists of 3 entities: the access code, the header, and the payload. In the figure, the number of bits per entity is indicated.

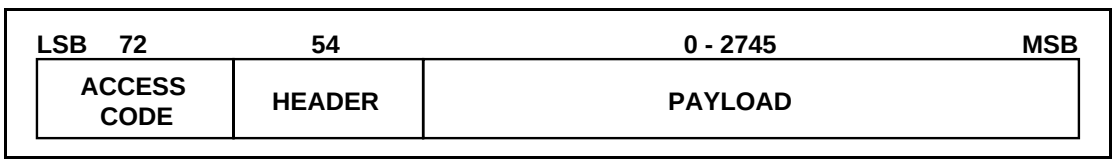


Figure 4.1: Standard packet format.

The access code and header are of fixed size : 72 bits and 54 bits respectively. The payload can range from zero to a maximum of 2745 bits. Different packet types have been defined. Packets may consist of the (shortened) access code only (see ID packet on page 55), of the access code – header, or of the access code – header – payload.

4.2 ACCESS CODE

Each packet starts with an access code. If a packet header follows, the access code is 72 bits long, otherwise the access code is 68 bits long. This access code is used for synchronization, DC offset compensation and identification. The access code identifies all packets exchanged on the channel of the piconet: all packets sent in the same piconet are preceded by the same channel access code. In the receiver of the Bluetooth unit, a sliding correlator correlates against the access code and triggers when a threshold is exceeded. This trigger signal is used to determine the receive timing.

The access code is also used in paging and inquiry procedures. In this case, the access code itself is used as a signalling message and neither a header nor a payload is present.

The access code consists of a preamble, a sync word, and possibly a trailer, see Figure 4.2 on page 48. For details see Section 4.2.1 on page 48.

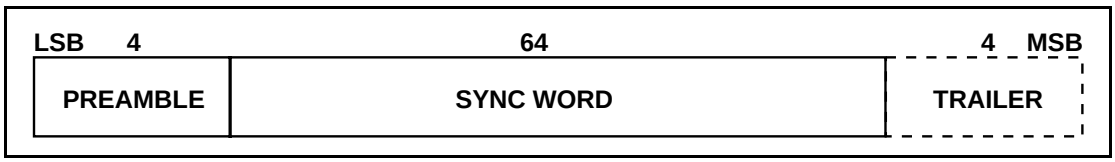


Figure 4.2: Access code format

4.2.1 Access code types

There are three different types of access codes defined:

- Channel Access Code (CAC)
- Device Access Code (DAC)
- Inquiry Access Code (IAC)

The respective access code types are used for a Bluetooth unit in different operating modes. The channel access code identifies a piconet. This code is included in all packets exchanged on the piconet channel. The device access code is used for special signalling procedures, e.g., paging and response to paging. For the inquiry access code there are two variations. A general inquiry access code (GIAC) is common to all devices. The GIAC can be used to discover which other Bluetooth units are in range. The dedicated inquiry access code (DIAC) is common for a dedicated group of Bluetooth units that share a common characteristic. The DIAC can be used to discover only these dedicated Bluetooth units in range.

The CAC consists of a **preamble**, **sync word**, and **trailer** and its total length is 72 bits. When used as self-contained messages without a header, the DAC and IAC do not include the trailer bits and are of length 68 bits.

The different access code types use different Lower Address Parts (LAPs) to construct the sync word. The LAP field of the BD address is explained in Section 13.1 on page 142. A summary of the different access code types can be found in Table 4.1 on page 49.

Code type	LAP	Code length	Comments
CAC	Master	72	See also Section 13.2 on page 142
DAC	Paged unit	68/72*	
GIAC	Reserved	68/72*	
DIAC	Dedicated	68/72*	

Table 4.1: Summary of access code types.

*. length 72 is only used in combination with FHS packets

4.2.2 Preamble

The preamble is a fixed zero-one pattern of 4 symbols used to facilitate DC compensation. The sequence is either 1010 or 0101, depending whether the LSB of the following sync word is 1 or 0, respectively. The preamble is shown in Figure 4.3 on page 49.

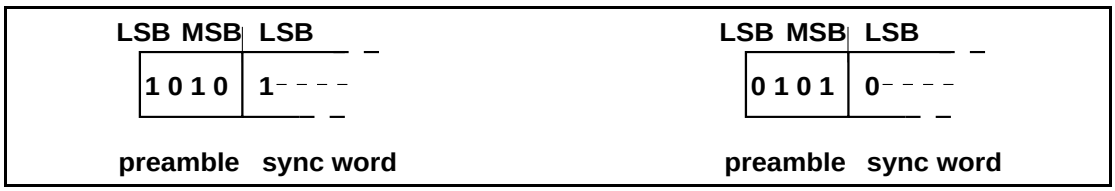


Figure 4.3: Preamble

4.2.3 Sync Word

The sync word is a 64-bit code word derived from a 24 bit address (LAP); for the CAC the master's LAP is used; for the GIAC and the DIAC, reserved, dedicated LAPs are used; for the DAC, the slave unit LAP is used. The construction guarantees large Hamming distance between sync words based on different LAPs. In addition, the good auto correlation properties of the sync word improve on the timing synchronization process. The derivation of the sync word is described in Section 13.2 on page 142

4.2.4 Trailer

The trailer is appended to the sync word as soon as the packet header follows the access code. This is typically the case with the CAC, but the trailer is also used in the DAC and IAC when these codes are used in FHS packets exchanged during page response and inquiry response procedures.

The trailer is a fixed zero-one pattern of four symbols. The trailer together with the three MSBs of the syncword form a 7-bit pattern of alternating ones and zeroes which may be used for extended DC compensation. The trailer sequence is either 1010 or 0101 depending on whether the MSB of the sync word is 0 or 1, respectively. The choice of trailer is illustrated in Figure 4.4 on page 50.

Figure 4.4: Trailer in CAC when MSB of sync word is 0 (a), and when MSB of sync word is 1 (b).

4.3 PACKET HEADER

The header contains link control (LC) information and consists of 6 fields:

- AM_ADDR: 3- bit active member address
- TYPE: 4-bit type code
- FLOW: 1-bit flow control
- ARQN: 1-bit acknowledge indication
- SEQN: 1-bit sequence number
- HEC: 8-bit header error check

The total header, including the HEC, consists of 18 bits, see Figure 4.5 on page 51, and is encoded with a rate 1/3 FEC (not shown but described in Section 5.1 on page 66) resulting in a 54-bit header. Note that the AM_ADDR and TYPE fields are sent with their LSB first. The function of the different fields will be explained next.

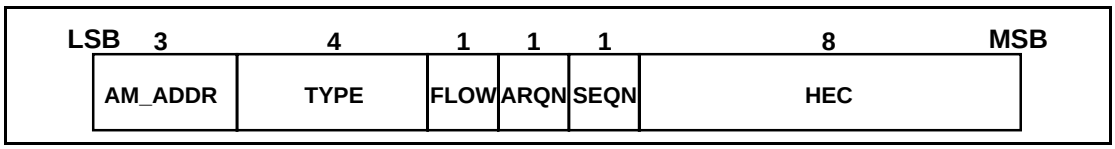


Figure 4.5: Header format.

4.3.1 AM_ADDR

The AM_ADDR represents a member address and is used to distinguish between the active members participating on the piconet. In a piconet, one or more slaves are connected to a single master.To identify each slave separately, each slave is assigned a temporary 3-bit address to be used when it is active. Packets exchanged between the master and the slave all carry the AM_ADDR of this slave; that is, the AM_ADDR of the slave is used in both master-to-slave packets and in the slave-to-master packets. The all-zero address is reserved for broadcasting packets from the master to the slaves. An exception is the FHS packet which may use the all-zero member address but is *not* a broadcast message (Section 4.4.1.4 on page 56). Slaves that are disconnected or parked give up their AM_ADDR. A new AM_ADDR has to be assigned when they re-enter the piconet.

4.3.2 TYPE

Sixteen different types of packets can be distinguished. The 4-bit TYPE code specifies which packet type is used. Important to note is that the interpretation of the TYPE code depends on the physical link type associated with the packet. First, it shall be determined whether the packet is sent on an SCO link or an ACL link. Then it can be determined which type of SCO packet or ACL packet has been received. The TYPE code also reveals how many slots the current packet will occupy. This allows the non-addressed receivers to refrain from listening to the channel for the duration of the remaining slots. In Section 4.4 on page 54, each packet type will be described in more detail.

4.3.3 FLOW

This bit is used for flow control of packets over the ACL link. When the RX buffer for the ACL link in the recipient is full and is not emptied, a STOP indication (FLOW=0) is returned to stop the transmission of data temporarily. Note, that the STOP signal only concerns ACL packets. Packets including only link control information (ID, POLL and NULL packets) or SCO packets can still be received. When the RX buffer is empty, a GO indication (FLOW=1) is returned. When no packet is received, or the received header is in error, a GO is assumed implicitly. In this case, the slave can receive a new packet with CRC although its RX buffer is still not emptied. The slave shall then return a NAK in response to this packet even if the packet passed the CRC check.

4.3.4 ARQN

The 1-bit acknowledgment indication ARQN is used to inform the source of a successful transfer of payload data with CRC, and can be positive acknowledge ACK or negative acknowledge NAK. If the reception was successful, an ACK (ARQN=1) is returned, otherwise a NAK (ARQN=0) is returned. When no return message regarding acknowledge is received, a NAK is assumed implicitly. NAK is also the default return information.

The ARQN is piggy-backed in the header of the return packet. The success of the reception is checked by means of a cyclic redundancy check (CRC) code. An unnumbered ARQ scheme which means that the ARQN relates to the latest received packet from the same source, is used. See Section 5.3 on page 67 for initialization and usage of this bit.

4.3.5 SEQN

The SEQN bit provides a sequential numbering scheme to order the data packet stream. For each new transmitted packet that contains data with CRC, the SEQN bit is inverted. This is required to filter out retransmissions at the destination; if a retransmission occurs due to a failing ACK, the destination receives the same packet twice. By comparing the SEQN of consecutive packets, correctly received retransmissions can be discarded. See section 5.3.2 on page 69 for initialization and usage of the SEQN bit. For broadcast packets, a modified sequencing method is used, see Section 5.3.5 on page 71.

4.3.6 HEC

Each header has a header-error-check to check the header integrity. The HEC consists of an 8-bit word generated by the polynomial 647 (octal representation). Before generating the HEC, the HEC generator is initialized with an 8-bit value. For FHS packets sent in **master page response** state, the slave upper address part (UAP) is used. For FHS packets sent in **inquiry response**, the default check initialization (DCI, see Section 5.4 on page 72) is used. In all other cases, the UAP of the master device is used. For the definition of Bluetooth device addresses, see Section 13.1 on page 142.

After the initialization, a HEC is calculated for the 10 header bits. Before checking the HEC, the receiver must initialize the HEC check circuitry with the proper 8-bit UAP (or DCI). If the HEC does not check, the entire packet is disregarded. More information can be found in Section 5.4 on page 72.

4.4 PACKET TYPES

The packets used on the piconet are related to the physical links they are used in. Up to now, two physical links are defined: the SCO link and the ACL link. For each of these links, 12 different packet types can be defined. Four control packets will be common to all link types: their TYPE code is unique irrespective of the link type.

To indicate the different packets on a link, the 4-bit TYPE code is used. The packet types have been divided into four segments. The first segment is reserved for the four control packets common to all physical link types; all four packet types have been defined. The second segment is reserved for packets occupying a single time slot; six packet types have been defined. The third segment is reserved for packets occupying three time slots; two packet types have been defined. The fourth segment is reserved for packets occupying five time slots; two packet types have been defined. The slot occupancy is reflected in the segmentation and can directly be derived from the type code. Table 4.2 on page 54 summarizes the packets defined so far for the SCO and ACL link types.

Segment	TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
1	0000	1	NULL	NULL
	0001	1	POLL	POLL
	0010	1	FHS	FHS
	0011	1	DM1	DM1
2	0100	1	undefined	DH1
	0101	1	HV1	undefined
	0110	1	HV2	undefined
	0111	1	HV3	undefined
	1000	1	DV	undefined
	1001	1	undefined	AUX1
3	1010	3	undefined	DM3
	1011	3	undefined	DH3
	1100	3	undefined	undefined
	1101	3	undefined	undefined
4	1110	5	undefined	DM5
	1111	5	undefined	DH5

Table 4.2: Packets defined for SCO and ACL link types

4.4.1 Common packet types

There are five common packets. In addition to the types listed in segment 1 of the previous table, there is the ID packet not listed. Each packet will now be examined in more detail.

4.4.1.1 ID packet

The identity or ID packet consists of the device access code (DAC) or inquiry access code (IAC). It has a fixed length of 68 bits. It is a very robust packet since the receiver uses a bit correlator to match the received packet to the known bit sequence of the ID packet. The packet is used, for example, in paging, inquiry, and response routines.

4.4.1.2 NULL packet

The NULL packet has no payload and therefore consists of the channel access code and packet header only. Its total (fixed) length is 126 bits. The NULL packet is used to return link information to the source regarding the success of the previous transmission (ARQN), or the status of the RX buffer (FLOW). The NULL packet itself does not have to be acknowledged.

4.4.1.3 POLL packet

The POLL packet is very similar to the NULL packet. It does not have a payload either. In contrast to the NULL packet, it requires a confirmation from the recipient. It is not a part of the ARQ scheme. The POLL packet does not affect the ARQN and SEQN fields. Upon reception of a POLL packet the slave must respond with a packet. This return packet is an implicit acknowledgement of the POLL packet. This packet can be used by the master in a piconet to poll the slaves, which must then respond even if they do not have information to send.

4.4.1.4 FHS packet

The FHS packet is a special control packet revealing, among other things, the Bluetooth device address and the clock of the sender. The payload contains 144 information bits plus a 16-bit CRC code. The payload is coded with a rate 2/3 FEC which brings the gross payload length to 240 bits. The FHS packet covers a single time slot.

Figure 4.6 on page 56 illustrates the format and contents of the FHS payload. The payload consists of eleven fields. The FHS packet is used in page master response, inquiry response and in master slave switch. In page master response or master slave switch, it is retransmitted until its reception is acknowledged or a timeout has exceeded. In inquiry response, the FHS packet is not acknowledged. The FHS packet contains real-time clock information. This clock information is updated before each retransmission. The retransmission of the FHS payload is thus somewhat different from the retransmission of ordinary data payloads where the same payload is used for each retransmission. The FHS packet is used for frequency hop synchronization before the piconet channel has been established, or when an existing piconet changes to a new piconet. In the former case, the recipient has not been assigned an active member address yet, in which case the AM_ADDR field in the FHS packet header is set to all-zeroes; however, the FHS packet should not be considered as a broadcast packet. In the latter case the slave already has an AM_ADDR in the existing piconet, which is then used in the FHS packet header.

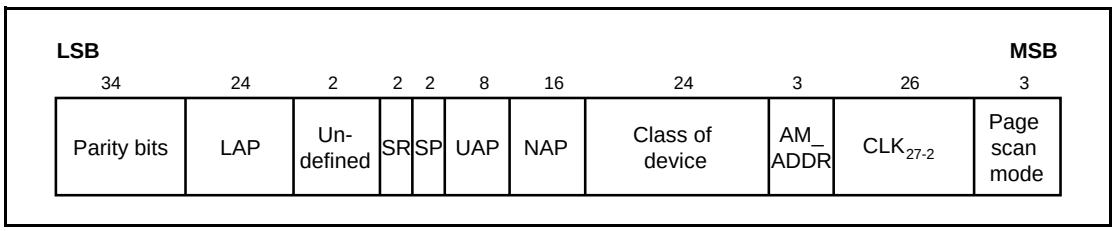


Figure 4.6: Format of the FHS payload

Each field is described in more detail below:

Parity bits	This 34-bit field contains the parity bits that form the first part of the sync word of the access code of the unit that sends the FHS packet. These bits are derived from the LAP as described in section 13.2 on page 142.
LAP	This 24-bit field contains the lower address part of the unit that sends the FHS packet.
Undefined	This 2-bit field is reserved for future use and shall be set to zero.
SR	This 2-bit field is the scan repetition field and indicates the interval between two consecutive page scan windows, see also Table 4.4 and Table 10.1 on page 97

Table 4.3: Description of the FHS payload

SP	This 2-bit field is the scan period field and indicates the period in which the mandatory page scan mode is applied after transmission of an inquiry response message, see also Table 4.5 and Table 10.6 on page 109.
UAP	This 8-bit field contains the upper address part of the unit that sends the FHS packet.
NAP	This 16-bit field contains the non-significant address part of the unit that sends the FHS packet (see also section 13.1 on page 142 for LAP, UAP, and NAP).
Class of device	This 24-bit field contains the class of device of the unit that sends the FHS packet. The field is defined in Bluetooth Assigned Numbers (http://www.bluetooth.org/assigned-numbers.htm).
AM_ADDR	This 3-bit field contains the member address the recipient shall use if the FHS packet is used at call setup or master-slave switch. A slave responding to a master or a unit responding to an inquiry request message shall include an all-zero AM_ADDR field if it sends the FHS packet.
CLK_{27-2}	This 26-bit field contains the value of the native system clock of the unit that sends the FHS packet, sampled at the beginning of the transmission of the access code of this FHS packet. This clock value has a resolution of 1.25ms (two-slot interval). For each new transmission, this field is updated so that it accurately reflects the real-time clock value.
Page scan mode	This 3-bit field indicates which scan mode is used by default by the sender of the FHS packet. The interpretation of the page scan mode is illustrated in Table 4.6. Currently, the standard supports one mandatory scan mode and up to three optional scan modes (see also "Appendix VII" on page1025).

Table 4.3: Description of the FHS payload

SR bit format b_1b_0	**SR mode**
00	R0
01	R1
10	R2
11	reserved

Table 4.4: Contents of SR field

SP bit format b_1b_0	**SP mode**
00	P0
01	P1
10	P2
11	reserved

Table 4.5: Contents of SP field

Bit format $b_2b_1b_0$	Page scan mode
000	Mandatory scan mode
001	Optional scan mode I
010	Optional scan mode II
011	Optional scan mode III
100	Reserved for future use
101	Reserved for future use
110	Reserved for future use
111	Reserved for future use

Table 4.6: Contents of page scan mode field

The LAP, UAP, and NAP together form the 48-bit IEEE address of the unit that sends the FHS packet. Using the parity bits and the LAP, the recipient can directly construct the channel access code of the sender of the FHS packet.

4.4.1.5 DM1 packet

DM1 serves as part of segment 1 in order to support control messages in any link type. However, it can also carry regular user data. Since the DM1 packet is recognized on the SCO link, it can interrupt the synchronous information to send control information. Since the DM1 packet can be regarded as an ACL packet, it will be discussed in Section 4.4.3 on page 60.

4.4.2 SCO packets

SCO packets are used on the synchronous SCO link. The packets do not include a CRC and are never retransmitted. SCO packets are routed to the synchronous I/O (voice) port. Up to now, three pure SCO packets have been defined. In addition, an SCO packet is defined which carries an asynchronous data field in addition to a synchronous (voice) field. The SCO packets defined so far are typically used for 64 kb/s speech transmission.

4.4.2.1 HV1 packet

The **HV1** packet carries 10 information bytes. The bytes are protected with a rate 1/3 FEC. No CRC is present. The payload length is fixed at 240 bits. There is no payload header present.

HV packets are used for transmission of voice and transparent synchronous data (see also Link Manager Protocol, section 3.21.1 on page 227. HV stands for High-quality Voice. The voice packets are never retransmitted and need no CRC.

An HV1 packet can carries 1.25ms of speech at a 64 kb/s rate.An HV1 packet has therefore to be sent every two time slots (T_{SCO}=2).

4.4.2.2 HV2 packet

The **HV2** packet carries 20 information bytes. The bytes are protected with a rate 2/3 FEC. No CRC is present. The payload length is fixed at 240 bits. There is no payload header present.

An HV2 packet carries 2.5ms of speech at a 64 kb/s rate. An HV2 packet has therefore to be sent every four time slots (T_{SCO}=4).

4.4.2.3 HV3 packet

The **HV3** packet carries 30 information bytes. The bytes are not protected by FEC. No CRC is present. The payload length is fixed at 240 bits. There is no payload header present.

An HV3 packet carries 3.75ms of speech at a 64 kb/s rate. An HV3 packet has therefore to be sent every six time slots (T_{SCO}=6).

4.4.2.4 DV packet

The **DV** packet is a combined data - voice packet. The payload is divided into a voice field of 80 bits and a data field containing up to 150 bits, see Figure 4.7. The voice field is not protected by FEC. The data field contains up to 10 information bytes (including the 1-byte payload header) and includes a 16-bit CRC. The data field is encoded with a rate 2/3 FEC. If necessary, extra zeroes are appended to assure that the total number of payload bits is a multiple of 10 prior to FEC encoding. Since the **DV** packet has to be sent at regular intervals due to its synchronous (voice) contents, it is listed under the SCO packet types. The voice and data fields are treated completely separate. The voice field is handled like normal SCO data and is never retransmitted; that is, the voice field is always new. The data field is checked for errors and is retransmitted if necessary.

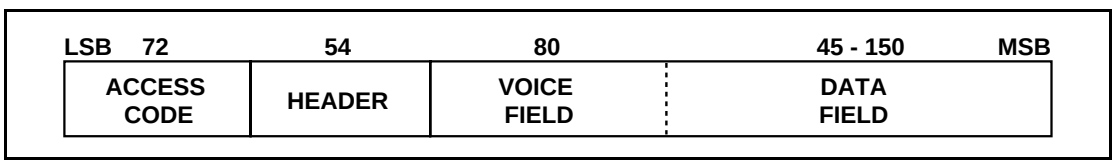


Figure 4.7: DV packet format

4.4.3 ACL packets

ACL packets are used on the asynchronous links. The information carried can be user data or control data. Including the DM1 packet, seven ACL packets have been defined. Six of the ACL packets contain a CRC code and retransmission is applied if no acknowledgement of proper reception is received (except in case a flush operation is carried out, see Section 5.3.3 on page 70). The 7th ACL packet, the AUX1 packet, has no CRC and is not retransmitted.

4.4.3.1 DM1 packet

The DM1 packet is a packet that carries data information only. DM stands for Data – Medium rate. The payload contains up to 18 information bytes (including the 1-byte payload header) plus a 16-bit CRC code. The DM1 packet may cover up to a single time slot. The information plus CRC bits are coded with a rate 2/3 FEC which adds 5 parity bits to every 10-bit segment. If necessary, extra zeros are appended after the CRC bits to get the total number of bits (information bits, CRC bits, and tail bits) equal a multiple of 10. The payload header in the DM1 packet is only 1 byte long, see Figure 4.8 on page 62. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).

4.4.3.2 DH1 packet

This packet is similar to the DM1 packet, except that the information in the payload is not FEC encoded. As a result, the DH1 frame can carry up to 28 information bytes (including the 1 byte payload header) plus a 16-bit CRC code. DH stands for Data – High rate. The DH1 packet may cover up to a single time slot.

4.4.3.3 DM3 packet

The DM3 packet is a DM1 packet with an extended payload. The DM3 packet may cover up to three time slots. The payload contains up to 123 information bytes (including the 2-bytes payload header) plus a 16-bit CRC code. The payload header in the DM3 packet is 2 bytes long, see Figure 4.9 on page 62. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code). When a DM3 packet is sent or received, the RF hop frequency shall not change for a duration of three time slots (the first time slot being the slot where the channel access code was transmitted).

4.4.3.4 DH3 packet

This packet is similar to the DM3 packet, except that the information in the payload is not FEC encoded. As a result, the DH3 packet can carry up to 185 information bytes (including the two bytes payload header) plus a 16-bit CRC code. The DH3 packet may cover three time slots. When a DH3 packet is sent or received, the hop frequency shall not change for a duration of three time slots (the first time slot being the slot where the channel access code was transmitted).

4.4.3.5 DM5 packet

The DM5 packet is a DM1 packet with an extended payload. The DM5 packet may cover up to five time slots. The payload contains up to 226 information bytes (including the 2-bytes payload header) plus a 16-bit CRC code. The payload header in the DM5 packet is 2 bytes long. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code). When a DM5 packet is sent or received, the hop frequency shall not change for a duration of five time slots (the first time slot being the slot where the channel access code was transmitted).

4.4.3.6 DH5 packet

This packet is similar to the DM5 packet, except that the information in the payload is not FEC encoded. As a result, the DH5 packet can carry up to 341 information bytes (including the two bytes payload header) plus a 16-bit CRC code. The DH5 packet may cover five time slots. When a DH5 packet is sent or received, the hop frequency shall not change for a duration of five time slots (the first time slot being the slot where the channel access code was transmitted).

4.4.3.7 AUX1 packet

This packet resembles a DH1 packet but has no CRC code. The AUX1 packet can carry up to 30 information bytes (including the 1-byte payload header). The AUX1 packet may cover up to a single time slot.

4.5 PAYLOAD FORMAT

In the previous packet overview, several payload formats were considered. In the payload, two fields are distinguished: the (synchronous) voice field and the (asynchronous) data field. The ACL packets only have the data field and the SCO packets only have the voice field – with the exception of the DV packets which have both.

4.5.1 Voice field

The voice field has a fixed length. For the HV packets, the voice field length is 240 bits; for the DV packet the voice field length is 80 bits. No payload header is present.

4.5.2 Data field

The data field consists of three segments: a payload header, a payload body, and possibly a CRC code (only the AUX1 packet does not carry a CRC code).

1. Payload header

Only data fields have a payload header. The payload header is one or two bytes long. Packets in segments one and two have a 1-byte payload header; packets in segments three and four have a 2-bytes payload header. The payload header specifies the logical channel (2-bit L_CH indication), controls the flow on the logical channels (1-bit FLOW indication), and has a payload length indicator (5 bits and 9 bits for 1-byte and 2-bytes payload header, respectively). In the case of a 2-byte payload header, the length indicator is extended by four bits into the next byte. The remaining four bits of the second byte are reserved for future use and shall be set to zero. The formats of the 1-byte and 2-bytes payload headers are shown in Figure 4.8 on page 62 and Figure 4.9 on page 62.

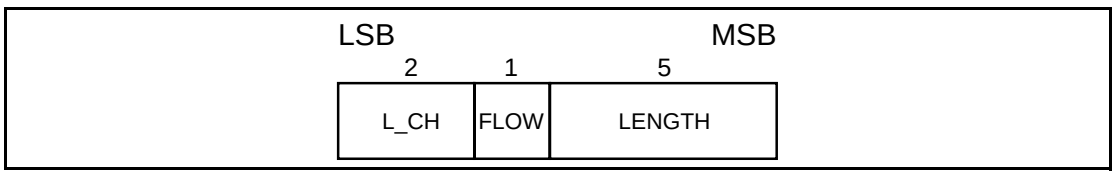


Figure 4.8: Payload header format for single-slot packets.

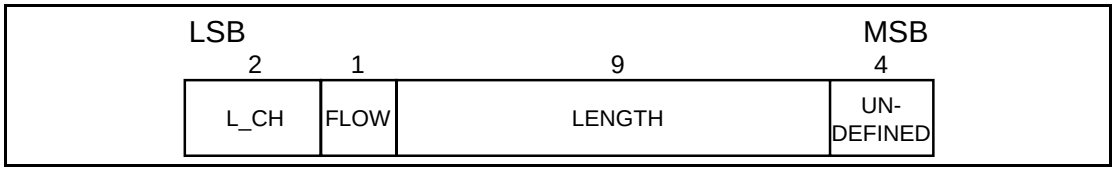


Figure 4.9: Payload header format for multi-slot packets.

The L_CH field is transmitted first, the length field last. In Table 4.7 on page 63, more details about the contents of the L_CH field are listed.

L_CH code b_1b_0	Logical Channel	Information
00	NA	undefined
01	UA/UI	Continuation fragment of an L2CAP message
10	UA/UI	Start of an L2CAP message or no fragmentation
11	LM	LMP message

Table 4.7: Logical channel L_CH field contents

An L2CAP message can be fragmented into several packets. Code 10 is used for an L2CAP packet carrying the first fragment of such a message; code 01 is used for continuing fragments. If there is no fragmentation, code 10 is used for every packet. Code 11 is used for LMP messages. Code 00 is reserved for future use.

The flow indicator in the payload is used to control the flow at the L2CAP level. It is used to control the flow per logical channel (when applicable). FLOW=1 means flow-on (“OK to send") and FLOW=0 means flow-off ("stop"). After a new connection has been established the flow indicator should be set to FLOW=1. When a Bluetooth unit receives a payload header with the flow bit set to "stop" (FLOW=0), it shall stop the transmission of ACL packets before an additional amount of payload data is sent. This amount can be defined as the flow control lag, expressed with a number of bytes. The shorter the flow control lag, the less buffering the other Bluetooth device must dedicate to this function. The flow control lag shall not exceed 1792 bytes (7*256 bytes), but in order to allow units to optimize the selection of packet length and buffer space, the flow control lag of a given implementation is provided in the LMP_features_res message.

If the packets containing the payload flow bit of "stop" is received with a valid packet header but bad payload, the payload flow control bit will not be recognized. The packet level ACK contained in the packet header will be received and a further ACL packet can be transmitted. Each occurrence of this situation allows a further ACL packet to be sent in spite of the flow control request being sent via the payload header flow control bit. It is recommended that Bluetooth units that use the payload header flow bit should ensure that no further ACL packets are sent until the payload flow bit has been correctly received. This can be accomplished by simultaneously turning on the flow bit in the packet header and keeping it on until an ACK is received back (ARQN=1). This will typically be only one round trip time. Since they lack a payload CRC, AUX1 packets should not be used with a payload flow bit of "stop".

The link manager is responsible for setting and processing the flow bit in the payload header. Real-time flow control is carried out at the packet level by the link controller via the flow bit in the packet header (see Section 4.3.3 on page 52). With the payload flow bit, traffic from the remote end can be controlled. It is allowed to generate and send an ACL packet with payloadlength zero irrespective of flow status. L2CAP start- and continue-fragment

indications (L_CH=10 and L_CH=01) also retain their meaning when the payload length is equal to zero (i.e. an empty start-fragment should not be sent in the middle of an on-going L2CAP packet transmission). It is always safe to send an ACL packet with payload length=0 and L_CH=01. The payload flow bit has its own meaning for each logical channel (UA/I or LM), see Table 4.8 on page 64. On the LM channel, no flow control is applied and the payload flow bit is always set at one.

L_CH code b_1b_0	Usage and semantics of the ACL payload header FLOW bit
00	Not defined, reserved for future use.
01 or 10	Flow control of the UA/I channels (which are used to send L2CAP messages)
11	Always set FLOW=1 on transmission and ignore the bit on reception

Table 4.8: Use of payload header flow bit on the logical channels.

The length indicator indicates the number of bytes (i.e. 8-bit words) in the payload excluding the payload header and the CRC code; i.e. the payload body only. With reference to Figure 4.8 and Figure 4.9, the MSB of the length field in a 1-byte header is the last (right-most) bit in the payload header; the MSB of the length field in a 2-byte header is the fourth bit (from left) of the second byte in the payload header.

2. Payload body

The payload body includes the user host information and determines the effective user throughput. The length of the payload body is indicated in the length field of the payload header.

3. CRC code generation

The 16-bit cyclic redundancy check code in the payload is generated by the CRC-CCITT polynomial `210041` (octal representation). It is generated in a way similar to the HEC. Before determining the CRC code, an 8-bit value is used to initialize the CRC generator. For the CRC code in the FHS packets sent in **master page response** state, the UAP of the slave is used. For the FHS packet sent in **inquiry response** state, the DCI (see Section 5.4 on page 72) is used. For all other packets, the UAP of the master is used.

The 8 bits are loaded into the 8 least significant (left-most) positions of the LFSR circuit, see Figure 5.10 on page 74. The other 8 bits are at the same time reset to zero. Subsequently, the CRC code is calculated over the information. Then the CRC code is appended to the information; the UAP (or DCI) is disregarded. At the receive side, the CRC circuitry is in the same way initialized with the 8-bit UAP (DCI) before the received information is checked. More information can be found in Section 5.4 on page 72.

4.6 PACKET SUMMARY

A summary of the packets and their characteristics is shown in Table 4.9, Table 4.10 and Table 4.11. The user payload represents the packet payload excluding FEC, CRC, and payload header.

Type	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate	Asymmetric Max. Rate
ID	na	na	na	na	na
NULL	na	na	na	na	na
POLL	na	na	na	na	na
FHS	18	2/3	yes	na	na

Table 4.9: Link control packets

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)	Asymmetric Max. Rate (kb/s)	
						Forward	Reverse
DM1	1	0-17	2/3	yes	108.8	108.8	108.8
DH1	1	0-27	no	yes	172.8	172.8	172.8
DM3	2	0-121	2/3	yes	258.1	387.2	54.4
DH3	2	0-183	no	yes	390.4	585.6	86.4
DM5	2	0-224	2/3	yes	286.7	477.8	36.3
DH5	2	0-339	no	yes	433.9	723.2	57.6
AUX1	1	0-29	no	no	185.6	185.6	185.6

Table 4.10: ACL packets

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)
HV1	na	10	1/3	no	64.0
HV2	na	20	2/3	no	64.0
HV3	na	30	no	no	64.0
DV*	1 D	10+(0-9) D	2/3 D	yes D	64.0+57.6 D

Table 4.11: SCO packets

*. Items followed by ‘D’ relate to data field only.

5 ERROR CORRECTION

There are three error correction schemes defined for Bluetooth:

- 1/3 rate FEC
- 2/3 rate FEC
- ARQ scheme for the data

The purpose of the FEC scheme on the data payload is to reduce the number of retransmissions. However, in a reasonable error-free environment, FEC gives unnecessary overhead that reduces the throughput. Therefore, the packet definitions given in Section 4 on page 47 have been kept flexible to use FEC in the payload or not, resulting in the **DM** and **DH** packets for the ACL link and the **HV** packets for the SCO link. The packet header is always protected by a 1/3 rate FEC; it contains valuable link information and should be able to sustain more bit errors.

Correction measures to mask errors in the voice decoder are not included in this section. This matter is discussed in section Section 12.3 on page 141.

5.1 FEC CODE: RATE 1/3

A simple 3-times repetition FEC code is used for the header. The repetition code is implemented by repeating the bit three times, see the illustration in Figure 5.1 on page 66. The 3-bit repetition code is used for the entire header, and also for the voice field in the **HV1** packet.

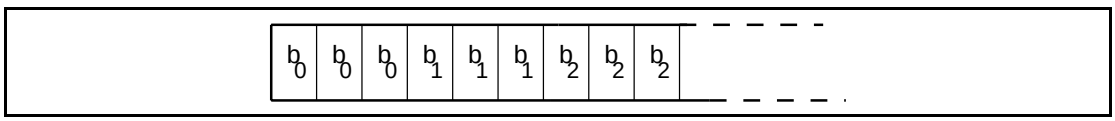


Figure 5.1: Bit-repetition encoding scheme.

5.2 FEC CODE: RATE 2/3

The other FEC scheme is a (15,10) shortened Hamming code. The generator polynomial is $g(D) = (D + 1)(D^4 + D + 1)$. This corresponds to 65 in octal notation. The LFSR generating this code is depicted in Figure 5.2 on page 67. Initially all register elements are set to zero. The 10 information bits are sequentially fed into the LFSR with the switches S1 and S2 set in position 1. Then, after the final input bit, the switches S1 and S2 are set in position 2, and the five parity bits are shifted out. The parity bits are appended to the information bits. Consequently, each block of 10 information bits is encoded into a 15 bit codeword. This code can correct all single errors and detect all double errors in each codeword. This 2/3 rate FEC is used in the **DM** packets, in the data field of the **DV** packet, in the **FHS** packet, and in the **HV2** packet. Since the encoder operates with information segments of length 10, tail bits with value zero may have to be appended after the CRC bits.

The total number of bits to encode, i.e., payload header, user data, CRC, and tail bits, must be a multiple of 10. Thus, the number of tail bits to append is the least possible that achieves this (i.e., in the interval 0...9). These tail bits are not included in the payload length indicator.

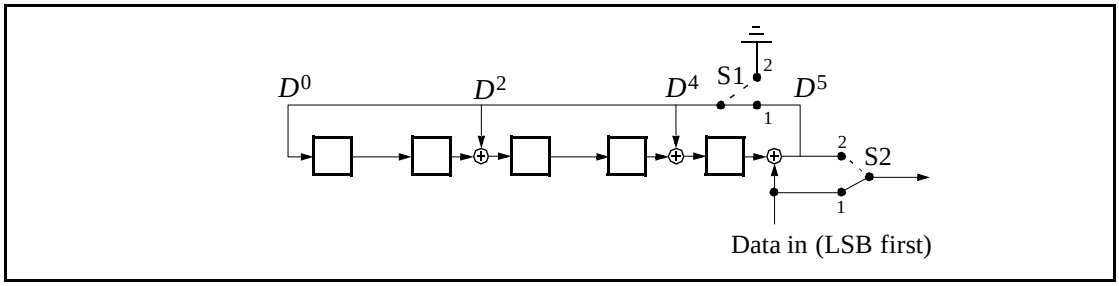


Figure 5.2: LFSR generating the (15,10) shortened Hamming code.

5.3 ARQ SCHEME

With an automatic repeat request scheme, **DM, DH** and the data field of **DV** packets are transmitted and retransmitted until acknowledgement of a successful reception is returned by the destination (or timeout is exceeded). The acknowledgement information is included in the header of the return packet, so-called piggy-backing. To determine whether the payload is correct or not, a cyclic redundancy check (CRC) code is added to the packet. The ARQ scheme only works on the payload in the packet (only that payload which has a CRC). The packet header and the voice payload are not protected by the ARQ scheme.

5.3.1 Unnumbered ARQ

Bluetooth uses a fast, unnumbered acknowledgment scheme: an ACK (ARQN=1) or a NAK (ARQN=0) is returned in response to the receipt of previously received packet. The slave will respond in the slave-to-master slot directly following the master-to-slave slot; the master will respond at the next event it will address the same slave (the master may have addressed other slaves between the last received packet from the considered slave and the master response to this packet). For a packet reception to be successful, at least the HEC must check. In addition, the CRC must check if present.

At the start of a new connection which may be the result of a page, page scan, master-slave switch or unpark, the master sends a POLL packet to verify the connection. In this packet the master initializes the ARQN bit to NAK. The response packet sent by the slave also has the ARQN bit set to NAK. The subsequent packets use the following rules.

The ARQ bit is affected by data packets containing CRC and empty slots only. As shown in Fig. 5.3 on pa ge68, upon successful reception of a CRC packet, the ARQN bit is set to ACK. If, in any receive slot in the slave, or, in a receive slot in the master following transmission of a packet, one of these events applies:

1. no access code is detected,
2. the HEC fails,
3. the CRC fails,

then the ARQN bit is set to NAK.

Packets that have correct HEC but that are addressed to other slaves, or packets other than DH, DM, or DV packets, do not affect the ARQN bit. In these cases the ARQN bit is left as it was prior to reception of the packet. If a CRC packet with a correct header has the same SEQN as the previously received CRC packet, the ARQN bit is set to ACK and the payload is disregarded without checking the CRC.

The ARQ bit in the FHS packet is not meaningful. Contents of the ARQN bit in the FHS packet should not be checked.

Broadcast packets are checked on errors using the CRC, but no ARQ scheme is applied. Broadcast packets are never acknowledged.

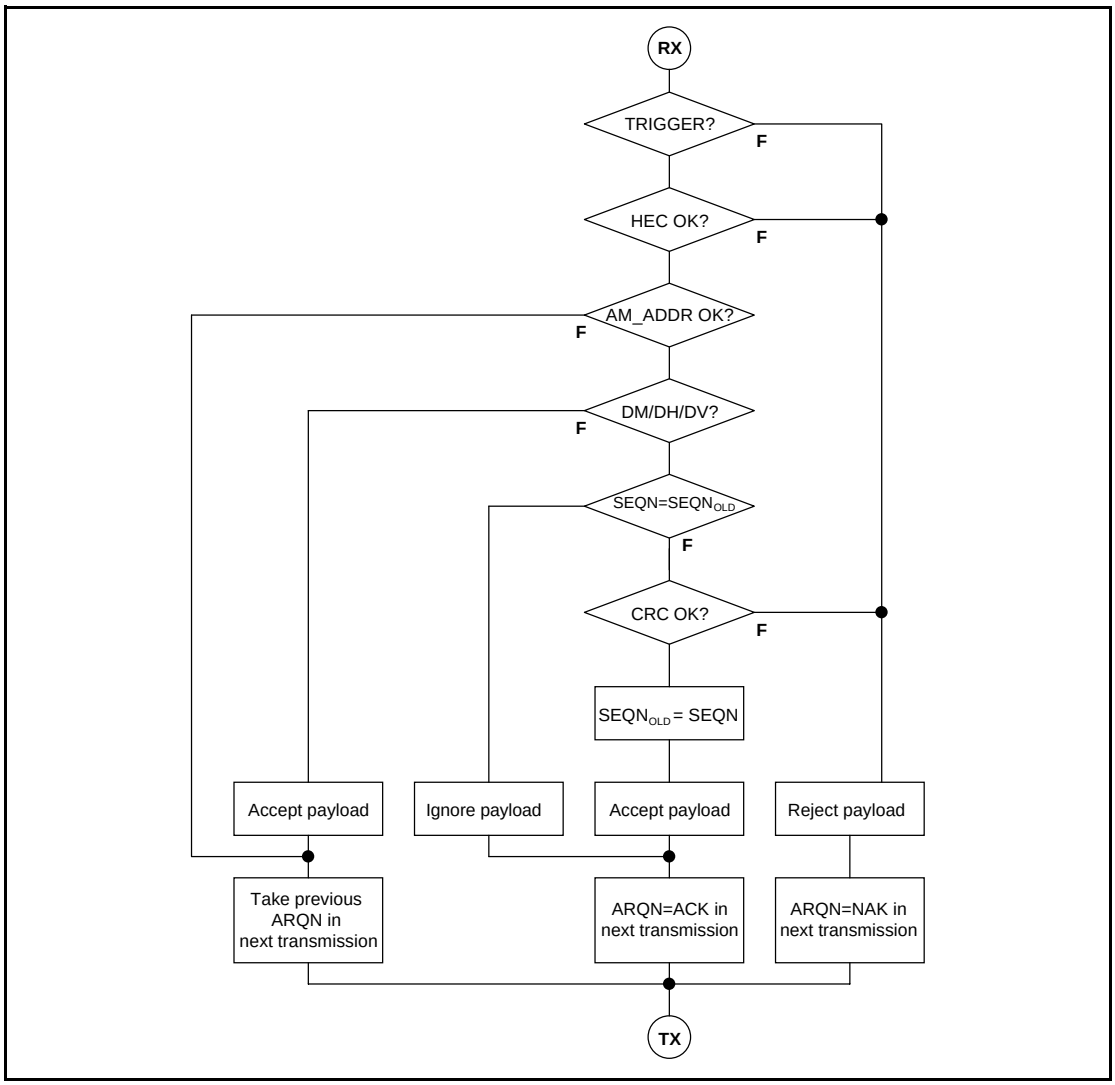


Figure 5.3: Receive protocol for determining the ARQN bit.

5.3.2 Retransmit filtering

The data payload is retransmitted until a positive acknowledgment is received (or a timeout is exceeded). A retransmission is carried out either because the packet transmission itself failed, or because the piggy-backed acknowledgment transmitted in the return packet failed (note that the latter has a lower failure probability since the header is more heavily coded). In the latter case, the destination keeps receiving the same payload over and over again. In order to filter out the retransmissions in the destination, the SEQN bit is added in the header. Normally, this bit is alternated for every new CRC data payload transmission. In case of a retransmission, this bit is not changed. So the destination can compare the SEQN bit with the previous SEQN value. If different, a new data payload has arrived; otherwise it is the same data payload and can be discarded. Only new data payloads are transferred to the link manager. Note that CRC data payloads can be carried only by **DM, DH** or **DV** packets.

At the start of a new connection which may be the result of a page, page scan, master slave switch or unpark, the master sends a POLL packet to verify the connection. The slave responds with a packet. The SEQN bit of the first CRC data packet, on both the master and the slave sides, is set to 1. The subsequent packets use the rules given below.

The SEQN bit is affected only by the CRC data packets as shown in Figure 5.4. It is inverted every time a new CRC data packet is sent. The CRC data packet is retransmitted with the same SEQN number until an ACK is received or the packet is flushed. When an ACK is received, the SEQN bit is inverted and a new payload is sent. When the packet is flushed (see Section 5.3.3 on page 70), a new payload is sent. The SEQN bit is not necessarily inverted. However, if an ACK is received before the new packet is sent, the SEQN bit is inverted. This procedure prevents loss of the first packet of a message (after the **flush** command has been given) due to the retransmit filtering.

If a device decides to flush, and it has not received an acknowledgement for the current packet, it replaces the current packet with an ACL L2CAP continuation packet with length zero. It transmits this packet with the same sequence number as the packet it is trying to flush until it does get an ACK. Only then can it move on to the next packet. If a flush is needed for the next packet in the transmit queue before the zero-length packet has been transmitted, that next packet can be removed from the queue directly without it also being replaced by a zero-length packet.The described flushing procedure is considered optional, although strongly recommended.

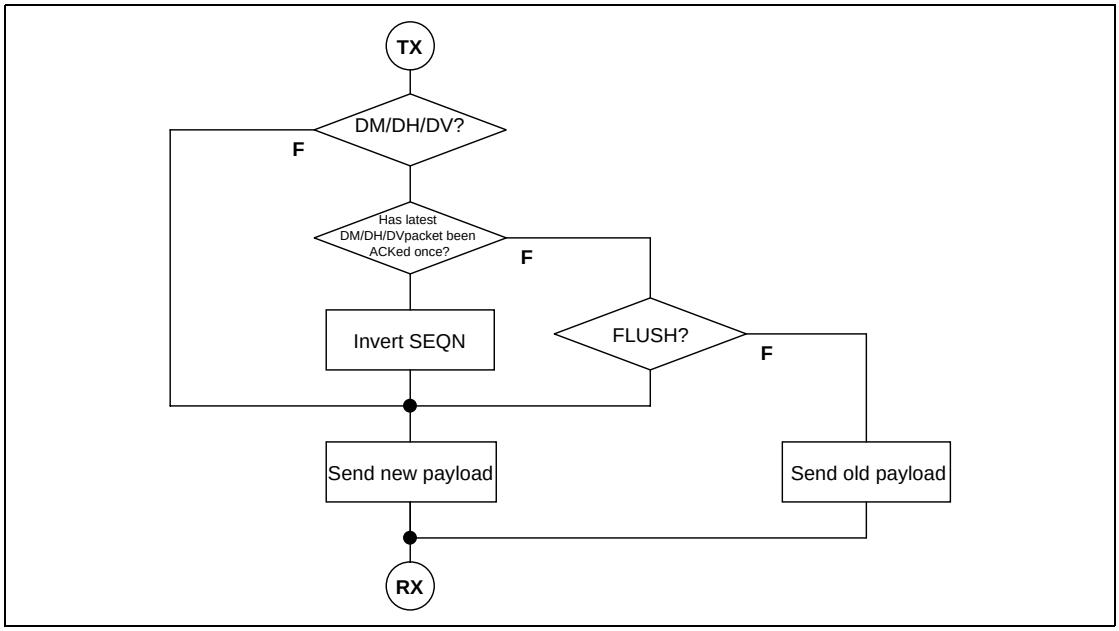


Figure 5.4: Retransmit filtering for packets with CRC.

The SEQN bit in the FHS packet is not meaningful. This bit can be set to any value. Contents of the SEQN bit in the FHS packet should not be checked. During transmission of all other packets the SEQN bit remains the same as it was in the previous packet.

5.3.3 Flushing payloads

The ARQ scheme can cause variable delay in the traffic flow since retransmissions are inserted to assure error-free data transfer. For certain communication links, only a limited amount of delay is allowed: retransmissions are allowed up to a certain limit at which the current payload must be disregarded and the next payload must be considered. This data transfer is indicated as **isochronous traffic**. This means that the retransmit process must be overruled in order to continue with the next data payload. Aborting the retransmit scheme is accomplished by *flushing* the old data and forcing the Bluetooth controller to take the next data instead.

Flushing results in loss of remaining portions of an L2CAP message. Therefore, the packet following the flush will have a start packet indication of L_CH = 10 in the payload header for the next L2CAP message. This informs the destination of the flush. (see Section 4.5 on page 62). Flushing will not necessarily result in a change in the SEQN bit value, see the previous section.

5.3.4 Multi-slave considerations

In case of a piconet with multiple slaves, the master carries out the ARQ protocol independently to each slave.

5.3.5 Broadcast packets

Broadcast packets are packets transmitted by the master to all the slaves simultaneously. A broadcast packet is indicated by the all-zero AM_ADDR (note; the FHS packet is the only packet which may have an all-zero address but is not a broadcast packet). Broadcast packets are not acknowledged (at least not at the LC level).

Since broadcast messages are not acknowledged, each broadcast packet is repeated for a fixed number of times. A broadcast packet is repeated N_{BC} times before the next broadcast packet of the same broadcast message is repeated, see Figure 5.5 on page 71. However, time - critical broadcast information may abort the ongoing broadcast repetition train. For instance, unpark messages sent at beacon instances may do this, see Section 10.8.4.5 on page 117.

Broadcast packets with a CRC have their own sequence number. The SEQN of the first broadcast packet with a CRC is set to SEQN = 1 by the master and it is inverted for each new broadcast packet with CRC thereafter. Broadcast packets without a CRC have no influence on the sequence number. The slave accepts the SEQN of the first broadcast packet it receives in a connection and checks for change in SEQN for consequent broadcast packets. Since there is no acknowledgement of broadcast messages and there is no end packet indication, it is important to receive the start packets correctly. To ensure this, repetitions of the broadcast packets that are L2CAP start packets and LMP packets will not be filtered out. These packets are indicated by L_CH=1X in the payload header as explained in section 4.5 on page 62. Only repetitions of the L2CAP continuation packets will be filtered out.

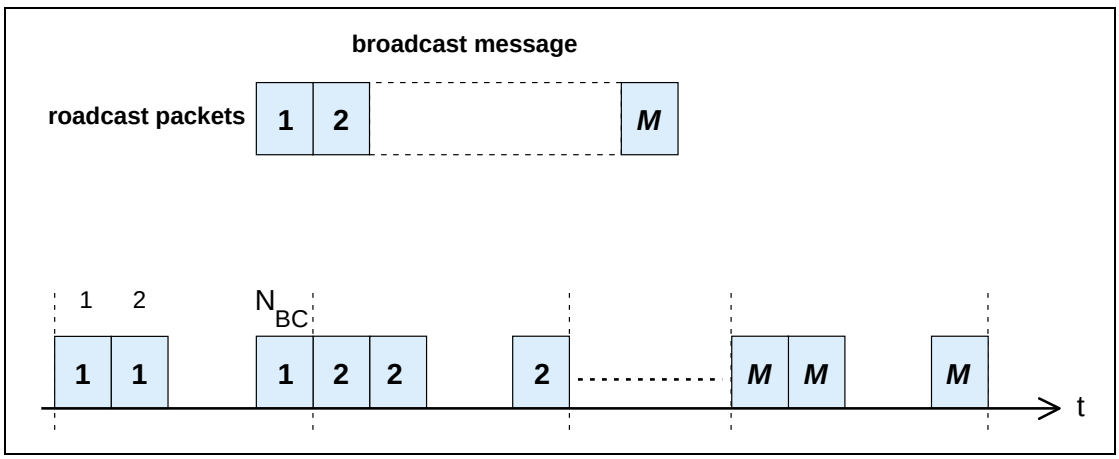


Figure 5.5: Broadcast repetition scheme

5.4 ERROR CHECKING

We can check the packet for errors or wrong delivery using the channel access code, the HEC in the header, and the CRC in the payload. At packet reception, first the access code is checked. Since the 64-bit sync word in the channel access code is derived from the 24-bit master LAP, this checks if the LAP is correct, and prevents the receiver from accepting a packet of another piconet.

The HEC and CRC are used to check both on errors and on a wrong address: to increase the address space with 8 bits, the UAP is normally included in the HEC and CRC checks. Then, even when a packet with the same access code – i.e., an access code of a device owning the same LAP but different UAP – passes the access code test, it will be discarded after the HEC and CRC tests when the UAP bits do not match. However, there is an exception where no common UAP is available in the transmitter and receiver. This is the case when the HEC and CRC are generated for the FHS packet in **inquiry response** state. In this case the default check initialization (DCI) value is used. The DCI is defined to be `0x00` (hexadecimal).

The generation and check of the HEC and CRC are summarized in Figure 5.8 on page 73 and Figure 5.11 on page 74. Before calculating the HEC or CRC, the shift registers in the HEC/CRC generators are initialized with the 8-bit UAP (or DCI) value. Then the header and payload information is shifted into the HEC and CRC generators, respectively (with the LSB first).

The HEC generating LFSR is depicted in Figure 5.6 on page 72. The generator polynomial is $g(D) = (D + 1)(D^7 + D^4 + D^3 + D^2 + 1) = D^8 + D^7 + D^5 + D^2 + D + 1$ Initially this circuit is pre-loaded with the 8-bit UAP such that the LSB of the UAP (denoted UAP_0) goes to the left-most shift register element, and, UAP_7 goes to the right-most element. The initial state of the HEC LFSR is depicted in Figure 5.7 on page 73. Then the data is shifted in with the switch S set in position 1. When the last data bit has been clocked into the LFSR, the switch S is set in position 2, and, the HEC can be read out from the register. The LFSR bits are read out from right to left (i.e., the bit in position 7 is the first to be transmitted, followed by the bit in position 6, etc.).

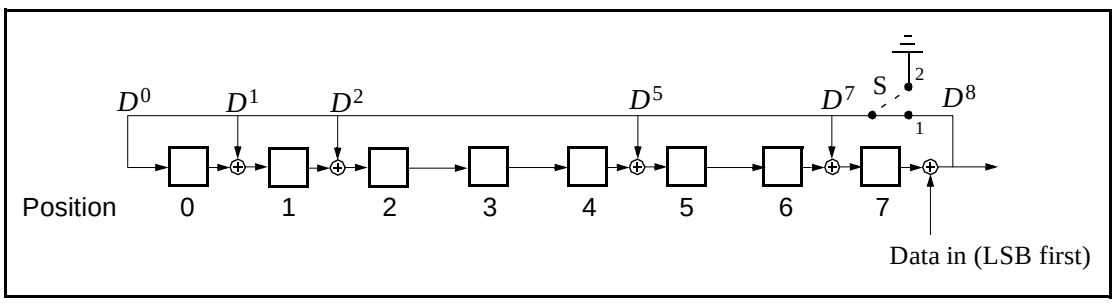


Figure 5.6: The LFSR circuit generating the HEC.

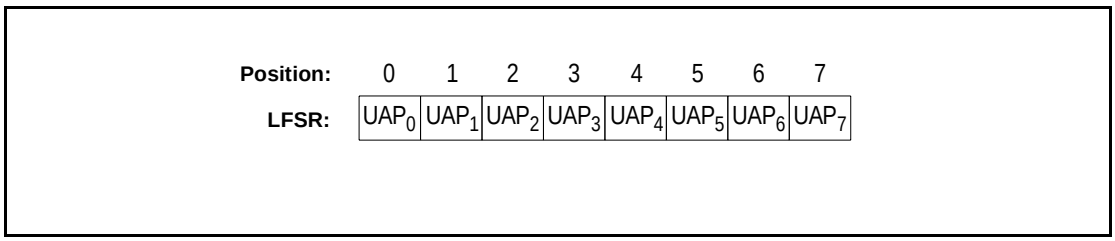


Figure 5.7: Initial state of the HEC generating circuit.

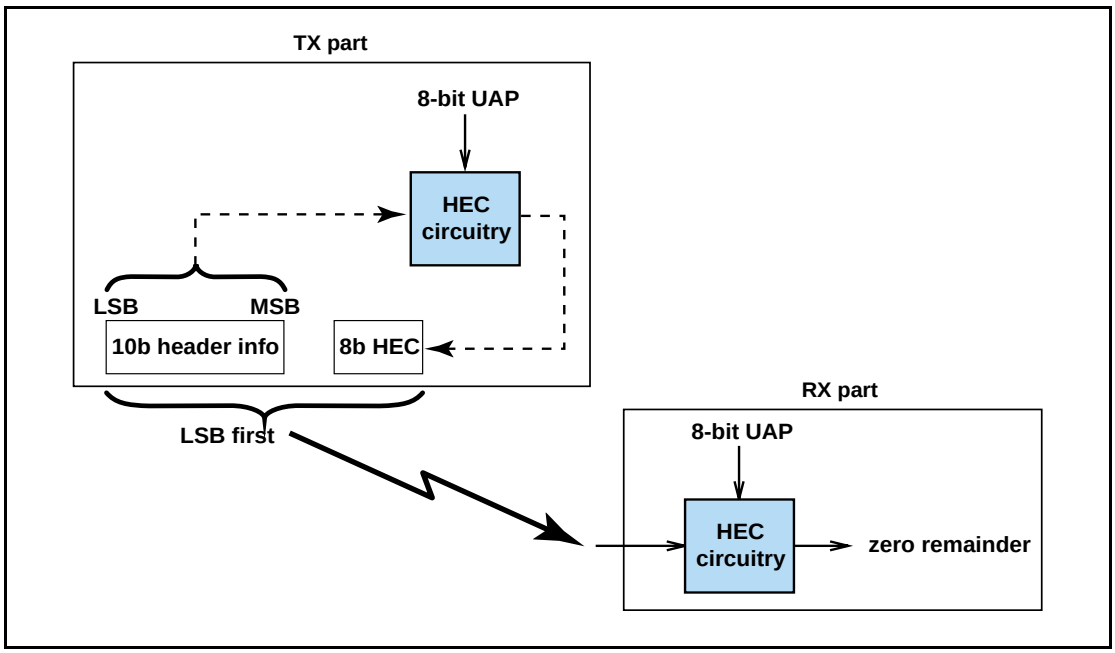


Figure 5.8: HEC generation and checking.

The 16 bit LFSR for the CRC is constructed similarly using the CRC-CCITT generator polynomial $g(D) = D^{16} + D^{12} + D^5 + 1$ (see Figure 5.9 on page 73). For this case, the 8 left-most bits are initially loaded with the 8-bit UAP (UAP_0 to the left and UAP_7 to the right) while the 8 right-most bits are reset to zero. The initial state of the 16 bit LFSR is depicted in Figure 5.10 on page 74. The switch S is set in position 1 while the data is shifted in. After the last bit has entered the LFSR, the switch is set in position 2, and, the register's contents are transmitted, from right to left (i.e., starting with position 15, then position 14, etc.).

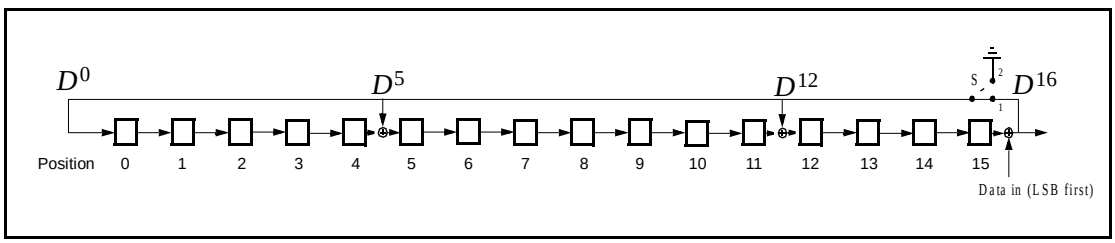


Figure 5.9: The LFSR circuit generating the CRC.

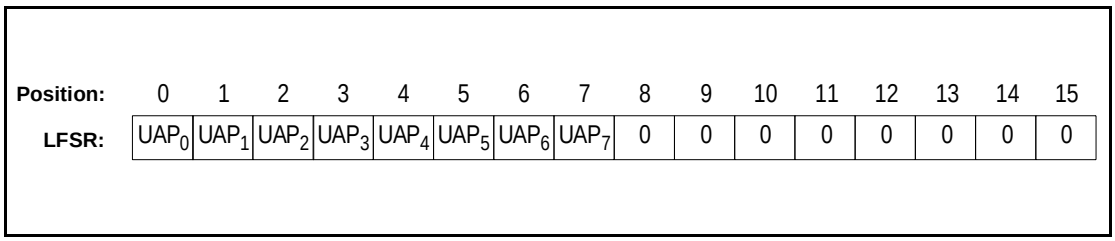


Figure 5.10: Initial state of the CRC generating circuit.

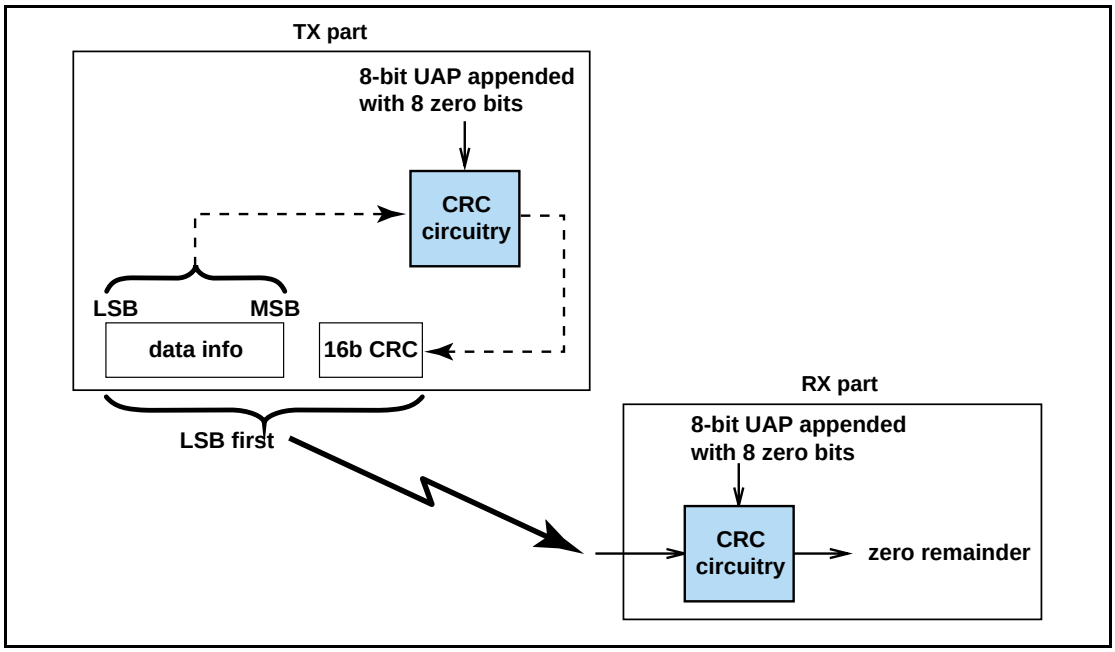


Figure 5.11: CRC generation and checking

6 LOGICAL CHANNELS

In the Bluetooth system, five logical channels are defined:

- LC control channel
- LM control channel
- UA user channel
- UI user channel
- US user channel

The control channels LC and LM are used at the link control level and link manager level, respectively. The user channels UA, UI, and US are used to carry asynchronous, isochronous, and synchronous user information, respectively. The LC channel is carried in the packet header; all other channels are carried in the packet payload. The LM, UA, and UI channels are indicated in the L_CH field in the payload header. The US channel is carried by the SCO link only; the UA and UI channels are normally carried by the ACL link; however, they can also be carried by the data in the DV packet on the SCO link. The LM channel can be carried either by the SCO or the ACL link.

6.1 LC CHANNEL (Link Control)

The LC control channel is mapped onto the packet header. This channel carries low level link control information like ARQ, flow control, and payload characterization. The LC channel is carried in every packet except in the **ID** packet which has no packet header.

6.2 LM CHANNEL (Link Manager)

The LM control channel carries control information exchanged between the link managers of the master and the slave(s). Typically, the LM channel uses protected **DM** packets. The LM channel is indicated by the L_CH code 11 in the payload header.

6.3 UA/UI CHANNEL (User Asynchronous/Isochronous Data)

The UA channel carries L2CAP transparent asynchronous user data. This data may be transmitted in one or more baseband packets. For fragmented messages, the start packet uses an L_CH code of 10 in the payload header. Remaining continuation packets use L_CH code 01. If there is no fragmentation, all packets use the L2CAP start code 10.

Isochronous data channel is supported by timing start packets properly at higher levels. At the baseband level, the L_CH code usage is the same as the UA channel.

6.4 US CHANNEL (User Synchronous Data)

The US channel carries transparent synchronous user data. This channel is carried over the SCO link.

6.5 CHANNEL MAPPING

The LC channel is mapped onto the packet header. All other channels are mapped onto the payload. The US channel can only be mapped onto the SCO packets. All other channels are mapped on the ACL packets, or possibly the SCO **DV** packet. The LM, UA, and UI channels may interrupt the US channel if it concerns information of higher priority.

7 DATA WHITENING

Before transmission, both the header and the payload are scrambled with a data whitening word in order to randomize the data from highly redundant patterns and to minimize DC bias in the packet. The scrambling is performed prior to the FEC encoding.

At the receiver, the received data is descrambled using the same whitening word generated in the recipient. The descrambling is performed after FEC decoding.

The whitening word is generated with the polynomial $g(D) = D^7 + D^4 + 1$ (i.e., 221 in octal representation) and is subsequently EXORed with the header and the payload. The whitening word is generated with the linear feedback shift register shown in Figure 7.1 on page 77. Before each transmission, the shift register is initialized with a portion of the master Bluetooth clock, $CLK_{6\text{-}1}$, extended with an MSB of value one. This initialization is carried out with CLK_1 written to position 0, CLK_2 written to position 1, etc. An exception forms the FHS packet sent during frequency hop acquisition, where initialization of the whitening register is carried out differently. Instead of the master clock, the X-input used in the **inquiry** or **page response** (depending on current state) routine is used, see Table 11.3 and Table 11.4 for the 79-hop and 23-hop systems, respectively. In case of a 79-hop system, the 5-bit values is extended with two MSBs of value one. In case of a 23-hop system, the 4-bit value is extended with three bits; the two MSBs are set to one and the third most significant bit is set to zero. During register initialization, the LSB of X (i.e., X_0) is written to position 0, X_1 is written to position 1, etc.

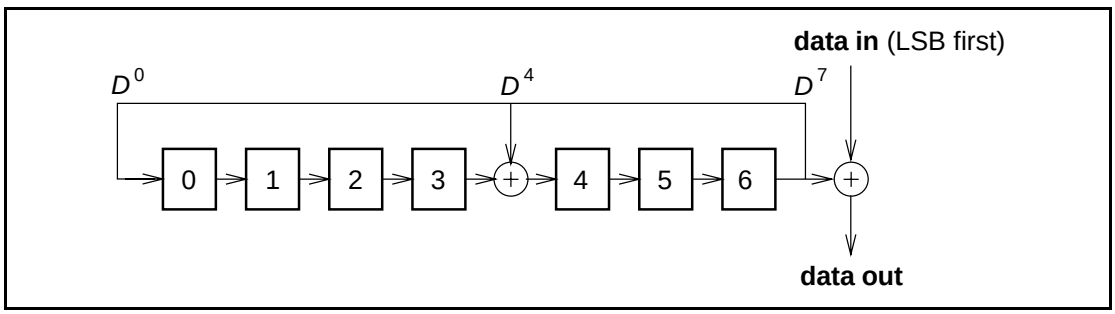


Figure 7.1: Data whitening LFSR.

After initialization, the packet header and the payload (including the CRC) are scrambled. The payload whitening continues from the state the whitening LFSR had at the end of HEC. There is no re-initialization of the shift register between packet header and payload. The first bit of the “Data In” sequence is the LSB of the packet header.

8 TRANSMIT/RECEIVE ROUTINES

This section describes the way to use the packets as defined in Section 4 on page 47 in order to support the traffic on the ACL and SCO links. Both single-slave and multi-slave configurations are considered. In addition, the use of buffers for the TX and RX routines are described.

The TX and RX routines described in sections 8.1 and 8.2 are of an informative character only. The final implementation may be carried out differently.

8.1 TX ROUTINE

The TX routine is carried out separately for each ACL link and each SCO link. Figure 8.1 on page 78 shows the ACL and SCO buffers as used in the TX routine. In this figure, only a single TX ACL buffer and a single TX SCO buffer are shown. In the master, there is a separate TX ACL buffer for each slave. In addition there may be one or more TX SCO buffers for each SCO slave (different SCO links may either reuse the same TX SCO buffer, or each have their own TX SCO buffer). Each TX buffer consists of two FIFO registers: one **current** register which can be accessed and read by the Bluetooth controller in order to compose the packets, and one **next** register that can be accessed by the Bluetooth Link Manager to load new information. The positions of the switches S1 and S2 determine which register is current and which register is next; the switches are controlled by the Bluetooth Link Controller. The switches at the input and the output of the FIFO registers can never be connected to the same register simultaneously.

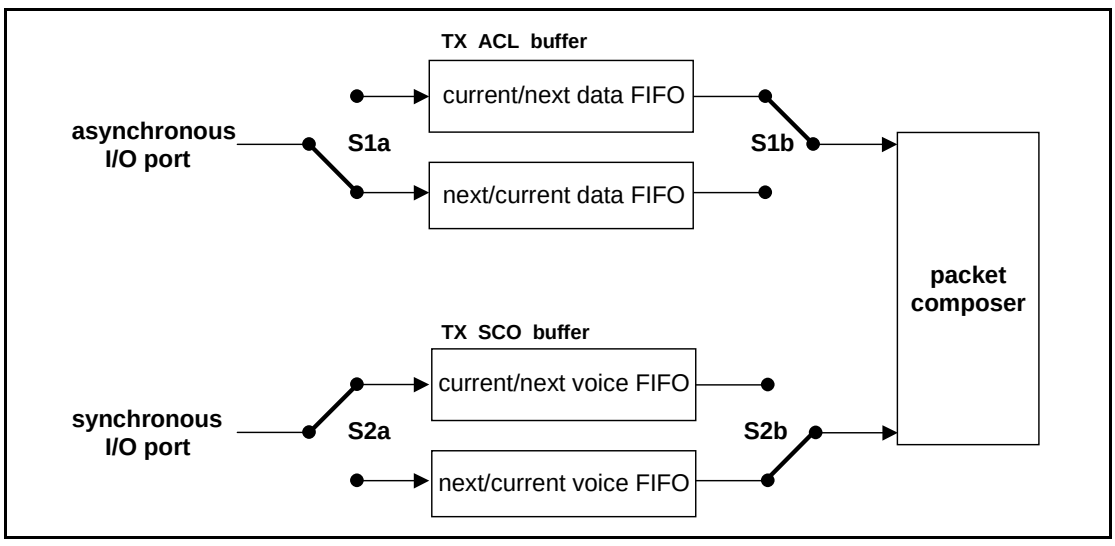


Figure 8.1: Functional diagram of TX buffering.

Of the packets common on the ACL and SCO links (**ID**, **NULL**, **POLL**, **FHS**, **DM1**) only the **DM1** packet carries a payload that is exchanged between the Link Controller and the Link Manager; this common packet makes use of the

ACL buffer. All ACL packets make use of the ACL buffer. All SCO packets make use of the SCO buffer except for the **DV** packet where the voice part is handled by the SCO buffer and the data part is handled by the ACL buffer. In the next sections, the operation for ACL traffic, SCO traffic, and combined data-voice traffic on the SCO link will be considered.

8.1.1 ACL traffic

In the case of pure (asynchronous) data, only the TX ACL buffer in Figure 8.1 on page 78 has to be considered. In this case, only packet types **DM** or **DH** are used, and these can have different lengths. The length is indicated in the payload header. The selection of high-rate data or medium-rate data shall depend on the quality of the link. When the quality is good, the FEC in the data payload can be omitted, resulting in a **DH** packet. Otherwise, **DM** packets must be used.

The default TYPE in pure data traffic is **NULL**. This means that, if there is no data to be sent (the data traffic is asynchronous, and therefore pauses occur in which no data is available) or no slaves need to be polled, **NULL** packets are sent instead – in order to send link control information to the other Bluetooth unit (e.g. ACK/STOP information for received data). When no link control information is available either (no need to acknowledge and/or no need to stop the RX flow) no packet is sent at all.

The TX routine works as follows. The Bluetooth Link Manager loads new data information in the register to which the switch S1a points. Next, it gives a **flush** command to the Bluetooth Link Controller , which forces the switch S1 to change (both S1a and S1b switch in synchrony). When the payload needs to be sent, the packet composer reads the current register and, depending on the packet TYPE, builds a payload which is appended to the channel access code and the header and is subsequently transmitted. In the response packet (which arrives in the following RX slot if it concerned a master transmission, or may be postponed until some later RX slot if it concerned a slave transmission), the result of the transmission is reported back. In case of an ACK, the switch S1 changes position; if a NAK (explicit or implicit) is received instead, the switch S1 will not change position. In that case, the same payload is retransmitted at the next TX occasion.

As long as the Link Manager keeps loading the registers with new information, the Bluetooth Link Controller will automatically transmit the payload; in addition, retransmissions are performed automatically in case of errors. The Link Controller will send **NULL** or nothing when no new data is loaded. If no new data has been loaded in the **next** register, during the last transmission, the packet composer will be pointing to an empty register after the last transmission has been acknowledged and the **next** register becomes the **current** register. If new data is loaded in the **next** register, a **flush** command is required to switch the S1 switch to the proper register. As long as the Link Manager keeps loading the data and type registers before each TX slot, the data is automatically processed by the Link Controller since the S1 switch is controlled by the

ACK information received in response. However, if the traffic from the Link Manager is interrupted once and a default packet is sent instead, a **flush** command is required to continue the flow in the Link Controller.

The **flush** command can also be used in case of time-bounded (isochronous) data. In case of a bad link, many retransmission are necessary. In certain applications, the data is time-bounded: if a payload is retransmitted all the time because of link errors, it may become outdated, and the system might decide to continue with more recent data instead and skip the payload that does not come through. This is accomplished by the **flush** command as well. With the **flush**, the switch S1 is forced to change and the Link Controller is forced to consider the next data payload and overrules the ACK control.

8.1.2 SCO traffic

In case of an SCO link, we only use **HV** packet types. The synchronous port continuously loads the **next** register in the SCO buffer. The S2 switches are changed according to the T_{SCO} interval. This T_{SCO} interval is negotiated between the master and the slave at the time the SCO link is established.

For each new SCO slot, the packet composer reads the **current** register after which the S2 switch is changed. If the SCO slot has to be used to send control information with high priority concerning a control packet between the master and the considered SCO slave, or a control packet between the master and any other slave, the packet composer will discard the SCO information and use the control information instead. This control information must be sent in a DM1 packet. Data or link control information can also be exchanged between the master and the SCO slave by using the **DV** or **DM1** packets. Any ACL type of packet can be used to sent data or link control information to any other ACL slave. This is discussed next.

8.1.3 Mixed data/voice traffic

In Section 4.4.2 on page 58, a **DV** packet has been defined that can support both data and voice simultaneously on a single SCO link. When the TYPE is **DV**, the Link Controller reads the data register to fill the data field and the voice register to fill the voice field. Thereafter, the switch S2 is changed. However, the position of S1 depends on the result of the transmission like on the ACL link: only if an ACK has been received will the S1 switch change its position. In each **DV** packet, the voice information is new, but the data information might be retransmitted if the previous transmission failed. If there is no data to be sent, the SCO link will automatically change from **DV** packet type to the current **HV** packet type used before the mixed data/voice transmission. Note that a **flush** command is required when the data stream has been interrupted and new data has arrived.

Combined data-voice transmission can also be accomplished by using separate ACL links in addition to the SCO link(s) if channel capacity permits this.

8.1.4 Default packet types

On the ACL links, the default type is always **NULL** both for the master and the slave. This means that if no user information needs to be send, either a **NULL** packet is sent if there is **ACK** or **STOP** information, or no packet is sent at all. The **NULL** packet can be used by the master to allocate the next slave-to-master slot to a certain slave (namely the one addressed). However, the slave is not forced to respond to the **NULL** packet from the master. If the master requires a response, it has to send a **POLL** packet.

The SCO packet type is negotiated at the LM level when the SCO link is established. The agreed packet type is also the default packet type for the SCO slots.

8.2 RX ROUTINE

The RX routine is carried out separately for the ACL link and the SCO link. However, in contrast to the master TX ACL buffer, a single RX buffer is shared among all slaves. For the SCO buffer, it depends how the different SCO links are distinguished whether extra SCO buffers are required or not. Figure 8.2 on page 81 shows the ACL and SCO buffers as used in the RX routine. The RX ACL buffer consists of two FIFO registers: one register that can be accessed and loaded by the Bluetooth Link Controller with the payload of the latest RX packet, and one register that can be accessed by the Bluetooth Link Manager to read the previous payload. The RX SCO buffer also consists of two FIFO registers: one register which is filled with newly arrived voice information, and one register which can be read by the voice processing unit.

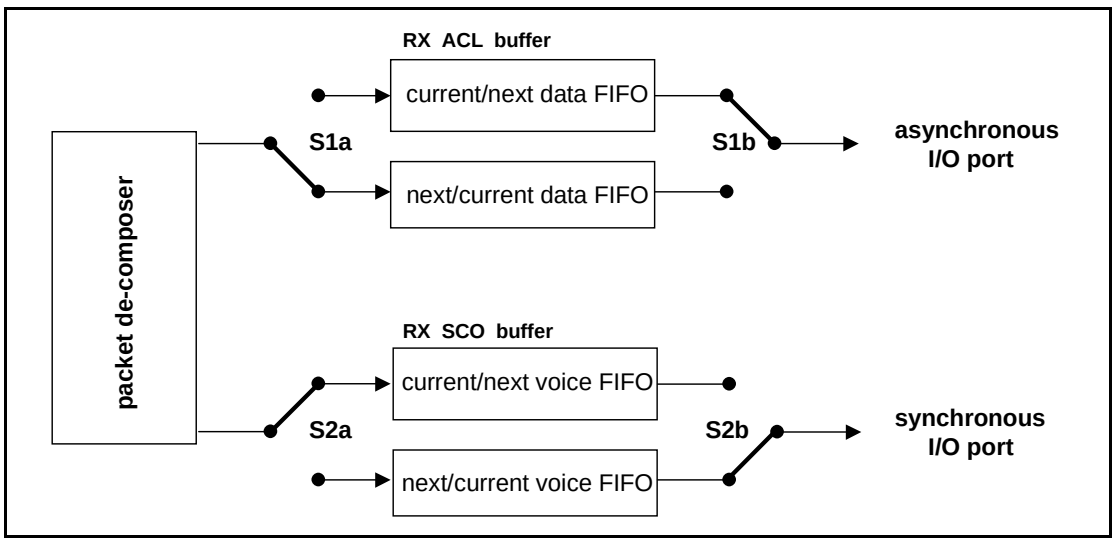


Figure 8.2: Functional diagram of RX buffering

Since the TYPE indication in the header of the received packet indicates whether the payload contains data and/or voice, the packet de-composer can automatically direct the traffic to the proper buffers. The switch S1 changes

every time the Link Manager has read the old register. If the next payload arrives before the RX register is emptied, a STOP indication must be included in the packet header of the next TX packet that is returned. The STOP indication is removed again as soon as the RX register is emptied. The SEQN field is checked before a new ACL payload is stored into the ACL register (flush indication in L_CH and broadcast messages influence the interpretation of the SEQN field see Section 5.3 on page 67).

The S2 switch is changed every T_{SCO}. If – due to errors in the header – no new voice payload arrives, the switch still changes. The voice processing unit then has to process the voice signal to account for the missing speech parts.

8.3 FLOW CONTROL

Since the RX ACL buffer can be full while a new payload arrives, flow control is required. As was mentioned earlier, the header field FLOW in the return TX packet can use STOP or GO in order to control the transmission of new data.

8.3.1 Destination control

As long as data cannot be received, a STOP indication is transmitted which is automatically inserted by the Link Controller into the header of the return packet. STOP is returned as long as the RX ACL buffer is not emptied by the Link Manager. When new data can be accepted again, the GO indication is returned. GO is the default value. Note that all packet types not including data can still be received. Voice communication for example is not affected by the flow control. Also note that although a Bluetooth unit cannot receive new information, it can still continue to transmit information: the flow control is separate for each direction.

8.3.2 Source control

On the reception of a STOP signal, the Link Controller will automatically switch to the default packet type. The ACL packet transmitted just before the reception of the STOP indication is kept until a GO signal is received. It is retransmitted as soon as a GO indication is received. Default packets are sent as long as the STOP indication is received. When no packet is received, GO is assumed implicitly. Note that the default packets contain link control information (in the header) for the receive direction (which may still be open) and may contain voice (**HV** packets). When a GO indication is received, the Link Controller resumes to transmit the data as is present in the TX ACL buffers.

In a multi-slave configuration, only the transmission to the slave that issued the STOP signal is stalled. This means that the previously described routine implemented in the master only concerns the TX ACL buffer that corresponds to the slave that cannot accept data momentarily.

8.4 BITSTREAM PROCESSES

Before the user information is sent over the air interface, several bit manipulations are performed in the transmitter to increase reliability and security. To the packet header, an HEC is added, the header bits are scrambled with a whitening word, and FEC coding is applied. In the receiver, the inverse processes are carried out. Figure 8.3 on page 83 shows the processes carried out for the packet header both at the transmit and the receive side. All header bit processes are mandatory.

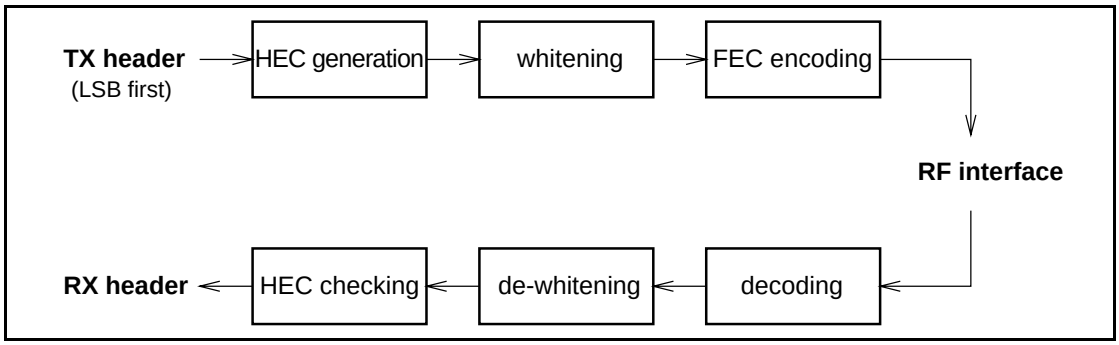


Figure 8.3: Header bit processes.

For the payload, similar processes are performed. It depends on the packet type, which processes are carried out. Figure 8.4 on page 83 shows the processes that may be carried out on the payload. In addition to the processes defined for the packet header, encryption can be applied on the payload. Only whitening and de-whitening, as explained in Section 7 on page 77, are mandatory for every payload; all other processes are optional and depend on the packet type and the mode enabled. In Figure 8.4 on page 83, optional processes are indicated by dashed blocks.

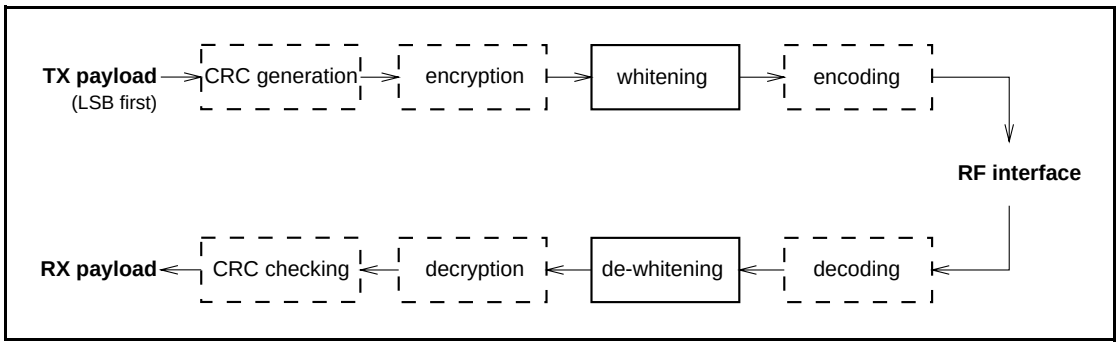


Figure 8.4: Payload bit processes.

9 TRANSMIT/RECEIVE TIMING

The Bluetooth transceiver applies a time-division duplex (TDD) scheme. This means that it alternately transmits and receives in a synchronous manner. It depends on the mode of the Bluetooth unit what the exact timing of the TDD scheme is. In the normal connection mode, *the master transmission shall always start at even numbered time slots (master CLK1=0) and the slave transmission shall always start at odd numbered time slots (master CLK1=1).* Due to packet types that cover more than a single slot, master transmission may continue in odd numbered slots and slave transmission may continue in even numbered slots.

All timing diagrams shown in this chapter are based on the signals as present at the antenna. The term “exact” when used to describe timing refers to an ideal transmission or reception and neglects timing jitter and clock frequency imperfections.

The average timing of master packet transmission must not drift faster than 20 ppm relative to the ideal slot timing of 625 µs. The instantaneous timing must not deviate more than 1 µs from the average timing. Thus, the absolute packet transmission timing t_k of slot boundary k must fulfill the equation:

$$t_k = \left(\sum_{i=1}^{k} (1 + d_i) T_N \right) + j_k + \text{offset}, \qquad \text{(EQ 1)}$$

where T_N is the nominal slot length (625 µs), j_k denotes jitter ($|j_k| \leq 1$ µs) at slot boundary k, and, d_k, denotes the drift ($|d_k| \leq 20$ ppm) within slot k. The jitter and drift may vary arbitrarily within the given limits for every slot, while “offset” is an arbitrary but fixed constant. For hold, park and sniff mode the drift and jitter parameters as described in Link Manager Protocol Section 3.9 on page 205 apply.

9.1 MASTER/SLAVE TIMING SYNCHRONIZATION

The piconet is synchronized by the system clock of the master. The master never adjusts its system clock during the existence of the piconet: it keeps an exact interval of Mx625 µs (where M is an even, positive integer larger than 0) between consecutive transmissions. The slaves adapt their native clocks with a timing offset in order to match the master clock. This offset is updated each time a packet is received from the master: by comparing the exact RX timing of the received packet with the estimated RX timing, the slaves correct the offset for any timing misalignments. Note that the slave RX timing can be corrected with any packet sent in the master-to-slave slot, since only the channel access code is required to synchronize the slave.

The slave TX timing shall be based on the most recent slave RX timing. The RX timing is based on the latest successful trigger during a master-to-slave slot. For ACL links, this trigger must have occurred in the master-to-slave slot directly pre-

ceding the current slave transmission; for SCO links, the trigger may have occurred several master-to-slave slots before since a slave is allowed to send an SCO packet even if no packet was received in the preceding master-to-slave slot. The slave shall be able to receive the packets and adjust the RX timing as long as the timing mismatch remains within the ±10 µs uncertainty window.

The master TX timing is strictly related to the master clock.The master shall keep an exact interval of Mx1250 µs (where M is a positive integer larger than 0) between the start of successive transmissions; the RX timing is based on this TX timing with a shift of exactly Nx625 µs (where N is an odd, positive integer larger than 0). During the master RX cycle, the master will also use the ±10µ uncertainty window to allow for slave misalignments. The master will adjust the RX processing of the considered packet accordingly, but will **not** adjust its RX/TX timing for the following TX and RX cycles. During periods when an active slave is not able to receive any valid channel access codes from the master, the slave may increase its receive uncertainty window and/or use predicted timing drift to increase the probability of receiving the master's bursts when reception resumes.

Timing behaviour may differ slightly depending on the current state of the unit. The different states are described in the next sections.

9.2 CONNECTION STATE

In the connection mode, the Bluetooth transceiver transmits and receives alternately, see Figure 9.1 on page 85 and Figure 9.2 on page 86 . In the figures, only single-slot packets are shown as an example. Depending on the type and the payload length, the packet size can be up to 366 µs. Each RX and TX transmission is at a different hop frequency. For multi-slot packets, several slots are covered by the same packet, and the hop frequency used in the first slot will be used throughout the transmission.

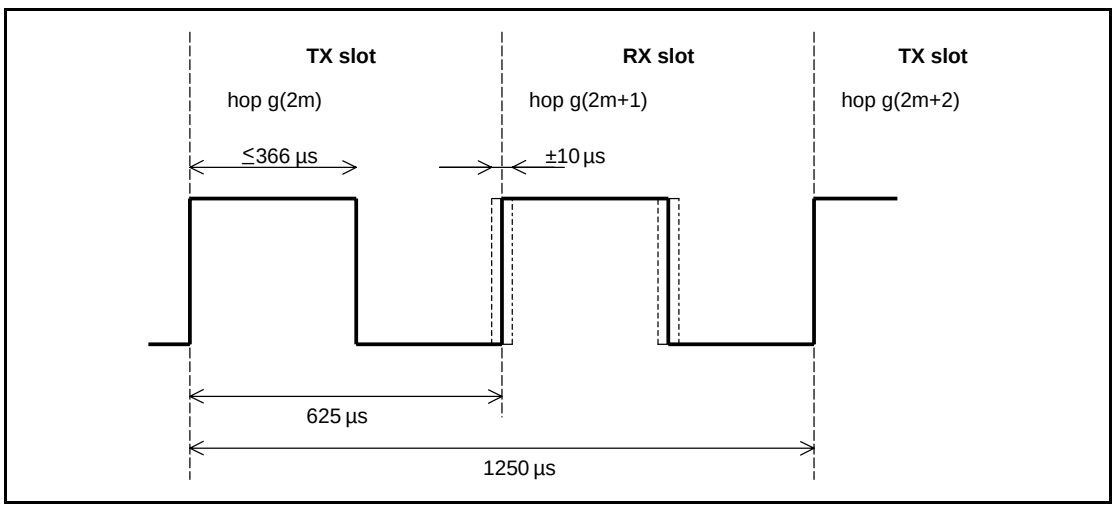


Figure 9.1: RX/TX cycle of Bluetooth master transceiver in normal mode for single-slot packets.

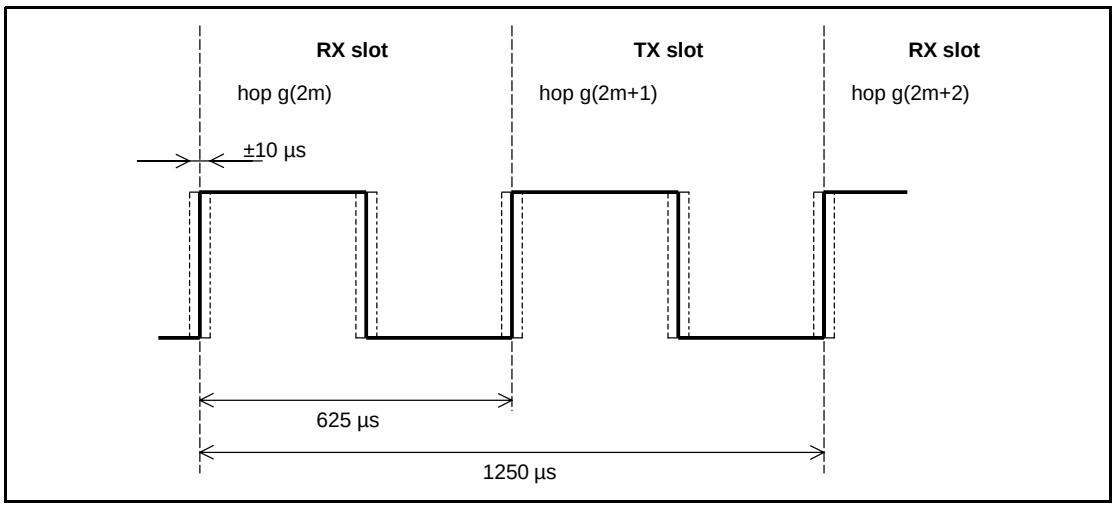


Figure 9.2: RX/TX cycle of Bluetooth slave transceiver in normal mode for single-slot packets.

The master TX/RX timing is shown in Figure 9.1 on page 85. In figures 9.1 through 9.6, f(k) is used for the frequencies of the page hopping sequence and f'(k) denotes the corresponding page response sequence frequencies. The channel hopping frequencies are indicated by g(m). After transmission, a return packet is expected Nx625 µs after the start of the TX burst where N is an odd, positive integer. N depends on the type of the transmitted packet. To allow for some time slipping, an uncertainty window is defined around the exact receive timing. During normal operation, the window length is 20 µs, which allows the RX burst to arrive up to 10 µs too early or 10 µs too late. During the beginning of the RX cycle, the access correlator searches for the correct channel access code over the uncertainty window. If no trigger event occurs, the receiver goes to sleep until the next RX event. If in the course of the search, it becomes apparent that the correlation output will never exceed the final threshold, the receiver may go to sleep earlier. If a trigger event does occur, the receiver remains open to receive the rest of the packet.

The current master transmission is based on the previous master transmission: it is scheduled Mx1250µs after the start of the previous master TX burst where M depends on the transmitted and received packet type. Note that the master TX timing is not affected by time drifts in the slave(s). If no transmission takes place during a number of consecutive slots, the master will take the TX timing of the latest TX burst as reference.

The slave's transmission is scheduled Nx625µs after the start of the slave's RX burst. If the slave's RX timing drifts, so will its TX timing. If no reception takes place during a number of consecutive slots, the slave will take the RX timing of the latest RX burst as reference.

9.3 RETURN FROM HOLD MODE

In the connection state, the Bluetooth unit can be placed in a **hold** mode, see Section 10.8 on page 110. In the **hold** mode, a Bluetooth transceiver neither transmits nor receives information. When returning to the normal operation after a **hold** mode in a slave Bluetooth unit, the slave must listen for the master before it may send information. In that case, the length of the search window in the slave unit may be increased from 20 µs to a larger value X µs as illustrated in Figure 9.3 on page 87. Note that only RX hop frequencies are used: the hop frequency used in the master-to-slave (RX) slot is also used in the uncertainty window extended into the preceding time interval normally used for the slave-to-master (TX) slot.

If the length of search window (X) exceeds 1250 µs, consecutive windows shall not be centered at the start of RX hops g(2m), g(2m+2), ... g(2m+2i) (where 'i' is an integer) to avoid ovelapping search windows. Consecutive windows should instead be centred at g(2m), g(2m+4), ... g(2m+4i), which gives a maximum value X=2500 µs, or even at g(2m), g(2m+6), ...g(2m+6i) which gives a maximum value X=3750 µs. The RX hop frequencies used shall correspond to the RX slot numbers.

It is recommended that single slot packets are used upon return from hold to minimize the synchronization time, especially after long hold periods that require search windows exceeding 625 µs.

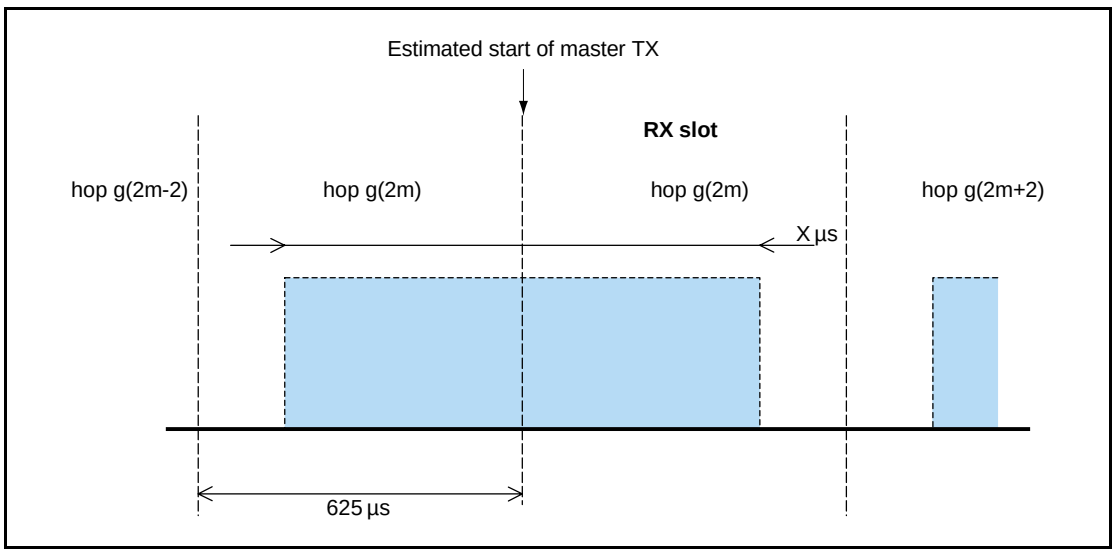


Figure 9.3: RX timing of slave returning from hold state.

9.4 PARK AND SNIFF MODES WAKE-UP

The **park** and **sniff**modes is similar to the **hold** mode. A slave in park or sniff mode periodically wakes up to listen to transmissions from the master and to re-synchronize its clock offset. As in the return from hold mode, a slave in park or sniff mode when waking up may increase the length of the search window from 20µs to a larger value X µs as illustrated in Figure 9.3 on page 87.

9.5 PAGE STATE

In the page state, the master transmits the device access code (ID packet) corresponding to the slave to be connected, rapidly on a large number of different hop frequencies. Since the ID packet is a very short packet, the hop rate can be increased from 1600 hops/s to 3200 hops/s. In a single TX slot interval, the paging master transmits on two different hop frequencies. In a single RX slot interval, the paging transceiver listens on two different hop frequencies; see Figure 9.4 on page 88. During the TX slot, the paging unit sends an ID packet at the TX hop frequencies *f(k)* and *f*(*k*+1*)*. In the RX slot, it listens for a response on the corresponding RX hop frequencies *f'(k)* and *f'*(*k*+1*)*. The listening periods are exactly timed 625 µs after the corresponding paging packets, and include a ±10 µs uncertainty window.

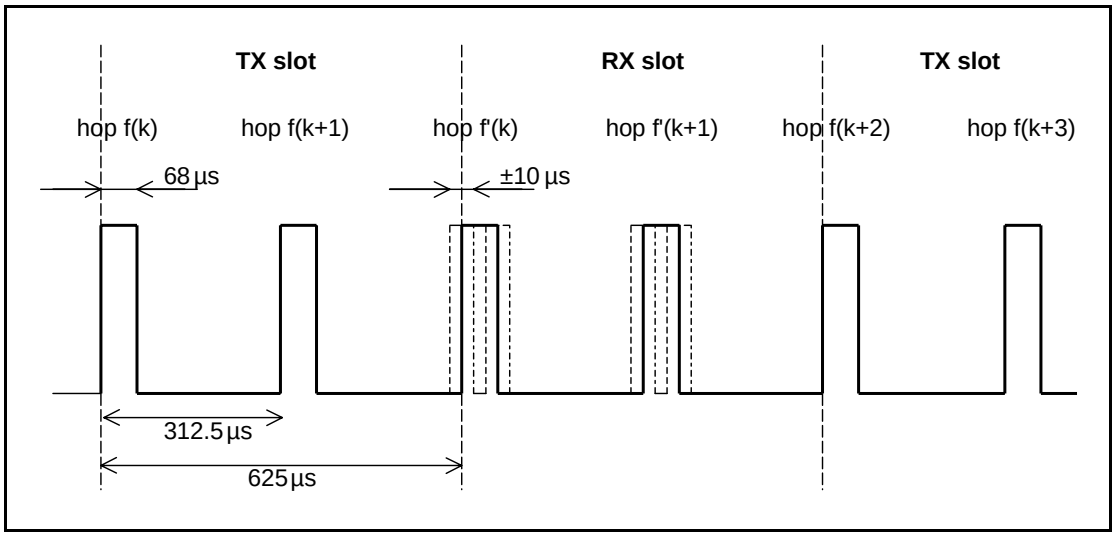


Figure 9.4: RX/TX cycle of Bluetooth transceiver in PAGE mode.

9.6 FHS PACKET

At connection setup and during a master-slave switch, an **FHS** packet is transferred from the master to the slave. This packet will establish the timing and frequency synchronization (see also Section 4.4.1.4 on page 56). After the slave unit has received the page message, it will return a response message which again consists of the ID packet and follows exactly 625 µs after the receipt of the page message. The master will send the FHS packet in the TX slot following the RX slot in which it received the slave response, according the RX/TX timing of the master. The time difference between the response and **FHS** message will depend on the timing of the page message the slave received. In Figure 9.5 on page 89, the slave receives the paging message sent **first** in the master-to-slave slot. It will then respond with an ID packet in the first half of the slave-to-master slot. The timing of the **FHS** packet is based on the timing of the page message sent first in the preceding master-to-slave slot: there is an exact 1250 µs delay between the first page message and the **FHS** packet. The packet is sent at the hop frequency *f(k+1)* which is the hop frequency following the hop frequency *f(k)* the page message was received in. In Figure 9.6 on page 90, the slave receives the paging message sent **secondly** in the master-to-slave slot. It will then respond with an ID packet in the second half of the slave-to-master slot exactly 625 µs after the receipt of the page message. The timing of the **FHS** packet is still based on the timing of the page message sent **first** in the preceding master-to-slave slot: there is an exact 1250 µs delay between the **first** page message and the **FHS** packet. The packet is sent at the hop frequency *f(k+2)* which is the hop frequency following the hop frequency *f(k+1)* the page message was received in.

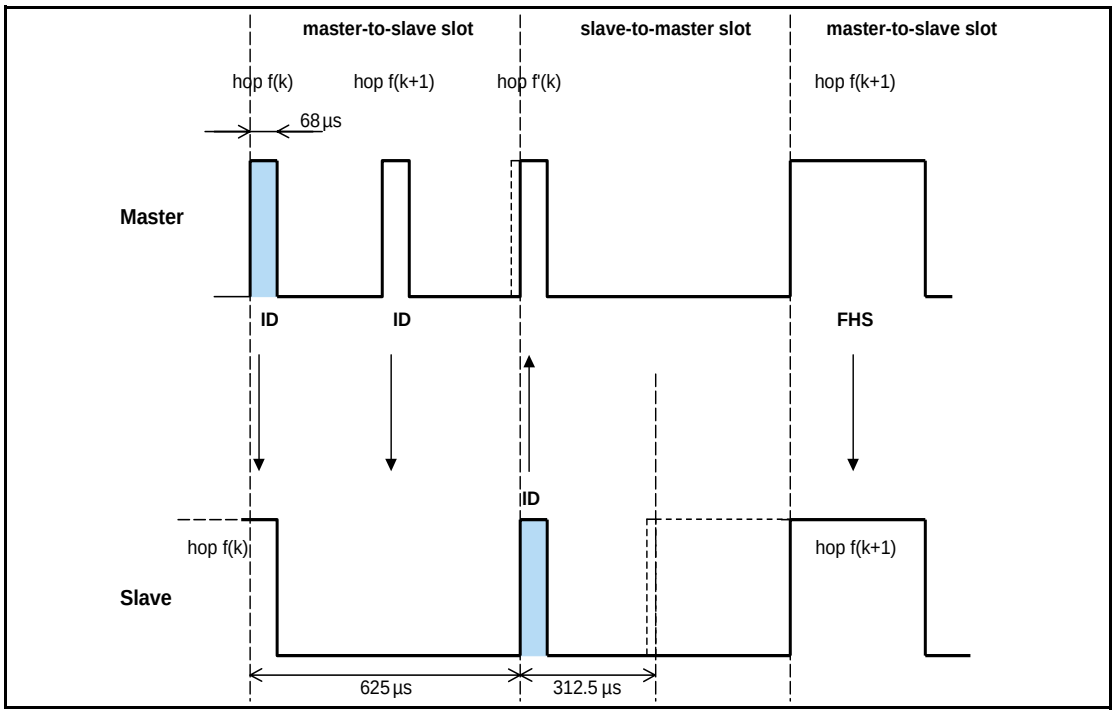


Figure 9.5: Timing of FHS packet on successful page in first half slot.

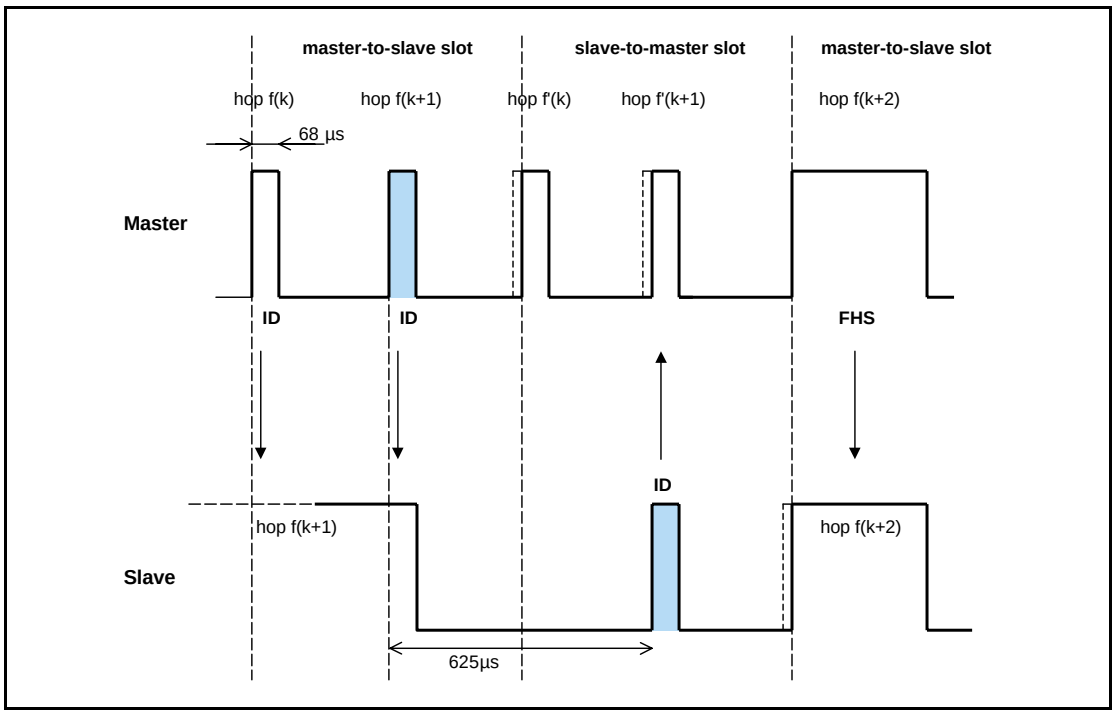


Figure 9.6: Timing of FHS packet on successful page in second half slot.

The slave will adjust its RX/TX timing according to the reception of the **FHS** packet (and not according to the reception of the page message). That is, the second response message that acknowledges the reception of the FHS packet is transmitted 625 µs after the start of the **FHS** packet.

9.7 MULTI-SLAVE OPERATION

As was mentioned in the beginning of this chapter, the master always starts the transmission in the even-numbered slots whereas the slaves start their transmission in the odd-numbered slots. This means that the timing of the master and the slave(s) is shifted by one slot (625 µs), see Figure 9.7 on page 91.

Only the slave that is addressed by its AM_ADDR can return a packet in the next slave-to-master slot. If no valid AM_ADDR is received, the slave may only respond if it concerns its reserved SCO slave-to-master slot. In case of a broadcast message, no slave is allowed to return a packet (an exception is found in the access window for access requests in the park mode, see Section 10.8.4 on page 112).

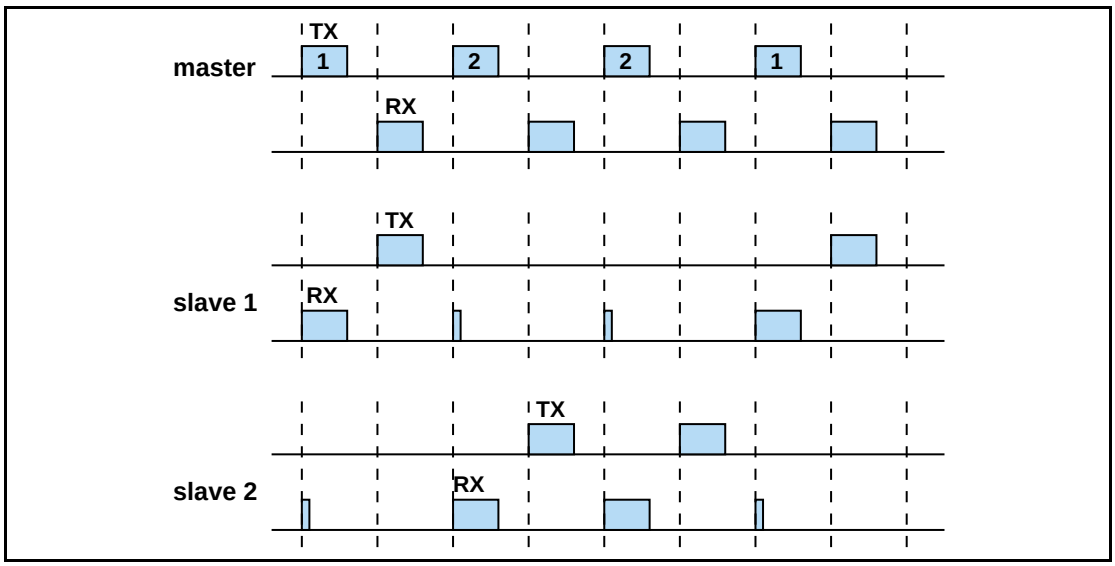


Figure 9.7: RX/TX timing in multi-slave configuration

10 CHANNEL CONTROL

10.1 SCOPE

This section describes how the channel of a piconet is established and how units can be added to and released from the piconet. Several states of operation of the Bluetooth units are defined to support these functions. In addition, the operation of several piconets sharing the same area, the so-called scatternet, is discussed. A special section is attributed to the Bluetooth clock which plays a major role in the FH synchronization.

10.2 MASTER-SLAVE DEFINITION

The channel in the piconet is characterized entirely by the master of the piconet. The Bluetooth device address (BD_ADDR) of the master determines the FH hopping sequence and the channel access code; the system clock of the master determines the phase in the hopping sequence and sets the timing. In addition, the master controls the traffic on the channel by a polling scheme.

By definition, the **master** is represented by the Bluetooth unit that initiates the connection (to one or more **slave** units). Note that the names ‘master’ and ‘slave’ only refer to the protocol on the channel: the Bluetooth units themselves are identical; that is, any unit can become a master of a piconet. Once a piconet has been established, master-slave roles can be exchanged. This is described in more detail in Section 10.9.3 on page 121.

10.3 BLUETOOTH CLOCK

Every Bluetooth unit has an internal system clock which determines the timing and hopping of the transceiver. The Bluetooth clock is derived from a free running native clock which is never adjusted and is never turned off. For synchronization with other units, only offsets are used that, added to the native clock, provide temporary Bluetooth clocks which are mutually synchronized. It should be noted that the Bluetooth clock has no relation to the time of day; it can therefore be initialized at any value. The Bluetooth clock provides the heart beat of the Bluetooth transceiver. Its resolution is at least half the TX or RX slot length, or 312.5 µs. The clock has a cycle of about a day. If the clock is implemented with a counter, a 28-bit counter is required that wraps around at 2^{28}-1. The LSB ticks in units of 312.5 µs, giving a clock rate of 3.2 kHz.

The timing and the frequency hopping on the channel of a piconet is determined by the Bluetooth clock of the master. When the piconet is established, the master clock is communicated to the slaves. Each slave adds an offset to its native clock to be synchronized to the master clock. Since the clocks are free-running, the offsets have to be updated regularly.

The clock determines critical periods and triggers the events in the Bluetooth receiver. Four periods are important in the Bluetooth system: 312.5 µs, 625 µs, 1.25 ms, and 1.28 s; these periods correspond to the timer bits CLK_0, CLK_1, CLK_2, and CLK_{12}, respectively, see Figure 10.1 on page 93. Master-to-slave transmission starts at the even-numbered slots when CLK_0 and CLK_1 are both zero.

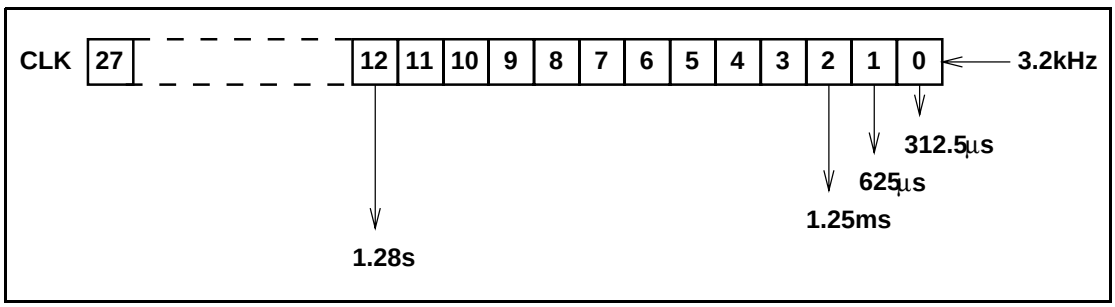


Figure 10.1: Bluetooth clock.

In the different modes and states a Bluetooth unit can reside in, the clock has different appearances:

- CLKN native clock
- CLKE estimated clock
- CLK master clock

CLKN is the free-running native clock and is the reference to all other clock appearances. In states with high activity, the native clock is driven by the reference crystal oscillator with worst case accuracy of +/-20ppm. In the low power states, like **STANDBY, HOLD, PARK** and **SNIFF**, the native clock may be driven by a low power oscillator (LPO) with relaxed accuracy (+/-250ppm).

CLKE and CLK are derived from the reference CLKN by adding an offset. CLKE is a clock estimate a paging unit makes of the native clock of the recipient; i.e. an offset is added to the CLKN of the pager to approximate the CLKN of the recipient, see Figure 10.2 on page 94. By using the CLKN of the recipient, the pager speeds up the connection establishment.

CLK is the master clock of the piconet. It is used for all timing and scheduling activities in the piconet. All Bluetooth devices use the CLK to schedule their transmission and reception. The CLK is derived from the native clock CLKN by adding an offset, see Figure 10.3 on page 94. The offset is zero for the master since CLK is identical to its own native clock CLKN. Each slave adds an appropriate offset to its CLKN such that the CLK corresponds to the CLKN of the master. Although all CLKNs in the Bluetooth devices run at the same nominal rate, mutual drift causes inaccuracies in CLK. Therefore, the offsets in the slaves must be regularly updated such that CLK is approximately CLKN of the master.

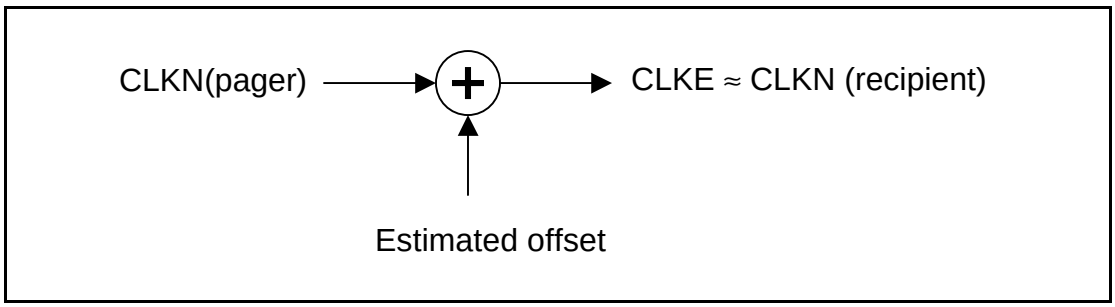


Figure 10.2: Derivation of CLKE

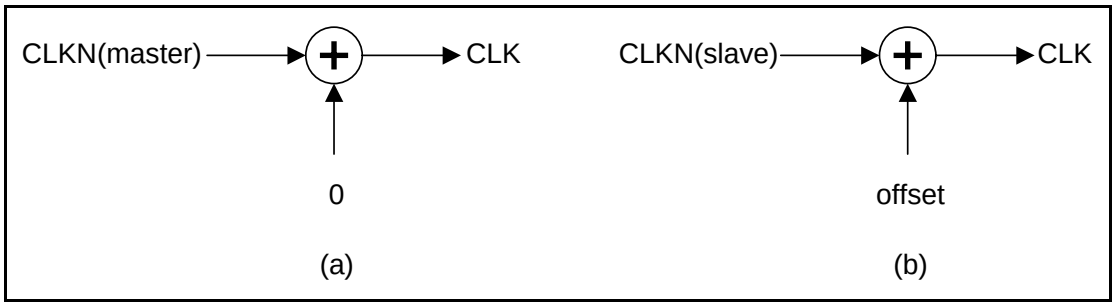


Figure 10.3: Derivation of CLK in master (a) and in slave (b).

10.4 OVERVIEW OF STATES

Figure 10.4 on page 95 shows a state diagram illustrating the different states used in the Bluetooth link controller. There are two major states: **STANDBY** and **CONNECTION**; in addition, there are seven substates, **page**, **page scan**, **inquiry**, **inquiry scan**, **master response**, **slave response**, and **inquiry response**. The substates are interim states that are used to add new slaves to a piconet. To move from one state to the other, either commands from the Bluetooth link manager are used, or internal signals in the link controller are used (such as the trigger signal from the correlator and the timeout signals).

.

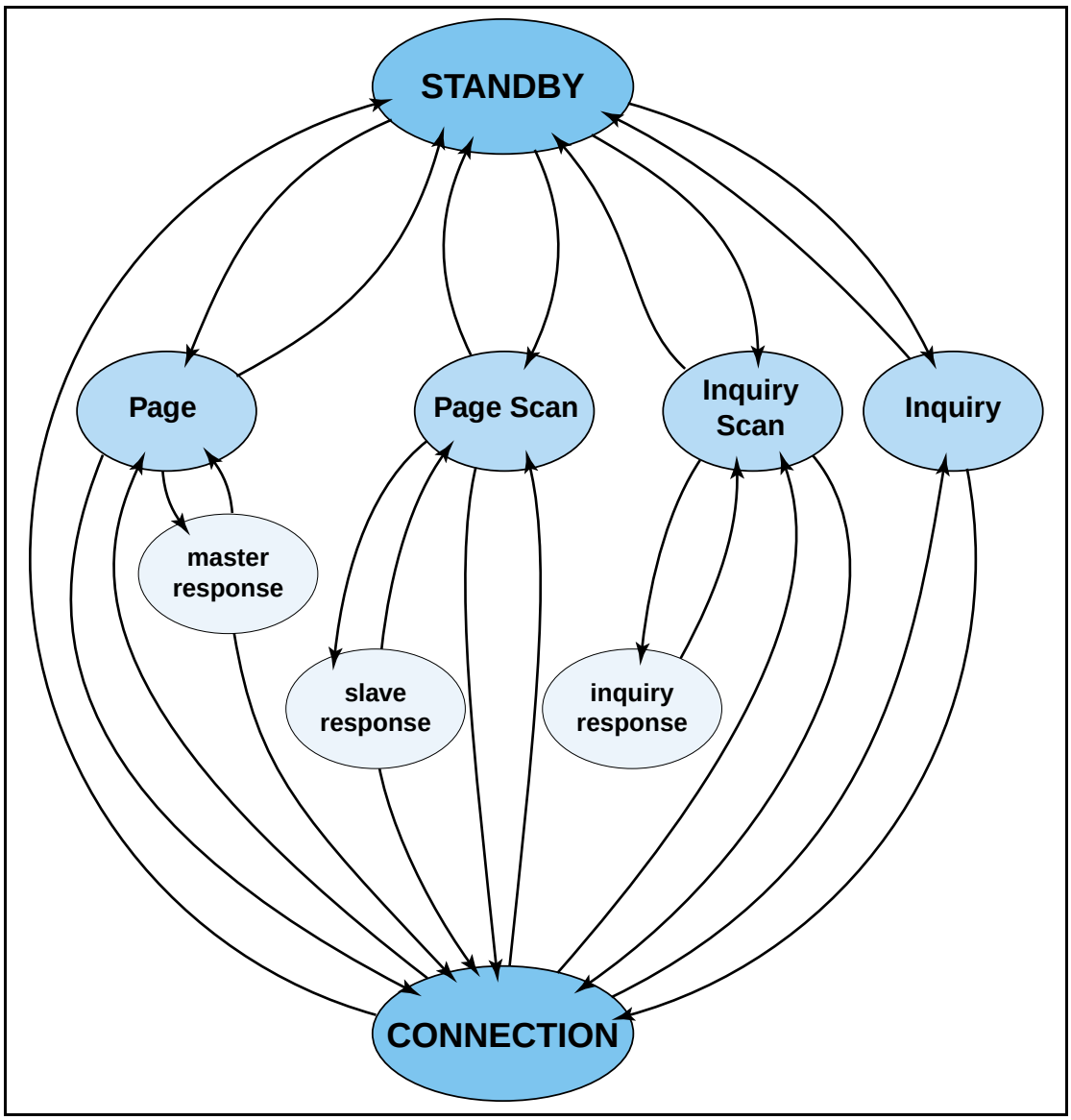


Figure 10.4: State diagram of Bluetooth link controller.

10.5 STANDBY STATE

The **STANDBY** state is the default state in the Bluetooth unit. In this state, the Bluetooth unit is in a low-power mode. Only the native clock is running at the accuracy of the LPO (or better).

The controller may leave the **STANDBY** state to scan for page or inquiry messages, or to page or inquiry itself. When responding to a page message, the unit will not return to the **STANDBY** state but enter the **CONNECTION** state as a slave. When carrying out a successful page attempt, the unit will enter the **CONNECTION** state as a master. The intervals with which scan activities can be carried out are discussed in Section 10.6.2 on page 96 and Section 10.7.2 on page 106.

10.6 ACCESS PROCEDURES

10.6.1 General

In order to establish new connections the procedures inquiry and paging are used. The inquiry procedure enables a unit to discover which units are in range, and what their device addresses and clocks are. With the paging procedure, an actual connection can be established. Only the Bluetooth device address is required to set up a connection. Knowledge about the clock will accelerate the setup procedure. A unit that establishes a connection will carry out a page procedure and will automatically be the master of the connection.

In the paging and inquiry procedures, the device access code (DAC) and the inquiry access code (IAC) are used, respectively. A unit in the **page scan** or **inquiry scan** substate correlates against these respective access codes with a matching correlator.

For the paging process, several paging schemes can be applied. There is one mandatory paging scheme which has to be supported by each Bluetooth device. This mandatory scheme is used when units meet for the first time, and in case the paging process directly follows the inquiry process. Two units, once connected using a mandatory paging/scanning scheme, may agree on an optional paging/scanning scheme. Optional paging schemes are discussed in "Appendix VII" on page 1025. In the current chapter, only the mandatory paging scheme is considered.

10.6.2 Page scan

In the **page scan** substate, a unit listens for its own device access code for the duration of the scan window $T_{w\ page\ scan}$. During the scan window, the unit listens at a single hop frequency, its correlator matched to its device access code. The scan window shall be long enough to completely scan 16 page frequencies.

When a unit enters the **page scan** substate, it selects the scan frequency according to the page hopping sequence corresponding to this unit, see Section 11.3.1 on page 134. This is a 32-hop sequence (or a 16-hop sequence in case of a reduced-hop system) in which each hop frequency is unique. The page hopping sequence is determined by the unit's Bluetooth device address (BD_ADDR). The phase in the sequence is determined by $CLKN_{16\text{-}12}$ of the unit's native clock ($CLKN_{15\text{-}12}$ in case of a reduced-hop system); that is, every 1.28s a different frequency is selected.

If the correlator exceeds the trigger threshold during the **page scan**, the unit will enter the **slave response** substate, which is described in Section 10.6.4.1 on page 102.

The **page scan** substate can be entered from the **STANDBY** state or the **CONNECTION** state. In the **STANDBY** state, no connection has been established and the unit can use all the capacity to carry out the **page scan**. Before entering the **page scan** substate from the **CONNECTION** state, the unit preferably reserves as much capacity for scanning. If desired, the unit may place ACL connections in the HOLD mode or even use the PARK mode, seeSection 10.8.3 on page 112 and Section 10.8.4 on page 112. SCO connections are preferably not interrupted by the **page scan**. In this case, the **page scan** may be interrupted by the reserved SCO slots which have higher priority than the **page scan**. SCO packets should be used requiring the least amount of capacity (**HV3** packets). The scan window shall be increased to minimize the setup delay. If one SCO link is present using **HV3** packets and T_{SCO}=6 slots, a total scan window $T_{w\ page\ scan}$ of at least 36 slots (22.5ms) is recommended; if two SCO links are present using **HV3** packets and T_{SCO}=6 slots, a total scan window of at least 54 slots (33.75ms) is recommended.

The scan interval $T_{page\ scan}$ is defined as the interval between the beginnings of two consecutive page scans. A distinction is made between the case where the scan interval is equal to the scan window $T_{w\ page\ scan}$ (continuous scan), the scan interval is maximal 1.28s, or the scan interval is maximal 2.56s. These three cases determine the behavior of the paging unit; that is, whether the paging unit shall use R0, R1 or R2, see also Section 10.6.3 on page 98. Table 10.1 illustrates the relationship between $T_{page\ scan}$ and modes R0, R1 and R2. Although scanning in the R0 mode is continuous, the scanning may be interrupted by for example reserved SCO slots. The scan interval information is included in the SR field in the FHS packet.

During page scan the Bluetooth unit may choose to use an optional scanning scheme. (An exception is the page scan after returning an inquiry response message. See Section 10.7.4 on page 108 for details.)

SR mode	$T_{page\ scan}$	N_{page}
R0	continuous	≥1
R1	≤ 1.28s	≥ 128
R2	≤ 2.56s	≥ 256
Reserved	-	-

Table 10.1: Relationship between scan interval, train repetition, and paging modes R0, R1 and R2.

10.6.3 Page

The **page** substate is used by the master (source) to activate and connect to a slave (destination) which periodically wakes up in the **page scan** substate. The master tries to capture the slave by repeatedly transmitting the slave's device access code (DAC) in different hop channels. Since the Bluetooth clocks of the master and the slave are not synchronized, the master does not know exactly when the slave wakes up and on which hop frequency. Therefore, it transmits a train of identical DACs at different hop frequencies, and listens in between the transmit intervals until it receives a response from the slave.

The page procedure in the master consists of a number of steps. First, the slave's device address is used to determine the page hopping sequence, see Section 11.3.2 on page 134. This is the sequence the master will use to reach the slave. For the phase in the sequence, the master uses an estimate of the slave's clock. This estimate can for example be derived from timing information that was exchanged during the last encounter with this particular device (which could have acted as a master at that time), or from an inquiry procedure. With this estimate CLKE of the slave's Bluetooth clock, the master can predict on which hop channel the slave will start page scan.

The estimate of the Bluetooth clock in the slave can be completely wrong. Although the master and the slave use the same hopping sequence, they use different phases in the sequence and will never meet each other.To compensate for the clock drifts, the master will send its page message during a short time interval on a number of wake-up frequencies. It will in fact transmit also on hop frequencies just before and after the current, predicted hop frequency. During each TX slot, the master sequentially transmits on two different hop frequencies. Since the page message is the ID packet which is only 68 bits in length, there is ample of time (224.5 µs minimal) to switch the synthesizer. In the following RX slot, the receiver will listen sequentially to two corresponding RX hops for ID packet. The RX hops are selected according to the page_response hopping sequence. The page_response hopping sequence is strictly related to the page hopping sequence; that is: for each page hop there is a corresponding page_response hop. The RX/TX timing in the **page** substate has been described in Section 9 on page 84, see also Figure 9.4 on page 88. In the next TX slot, it will transmit on two hop frequencies different from the former ones. The synthesizer hop rate is increased to 3200 hops/s.

A distinction must be made between the 79-hop systems and the 23-hop systems. First the 79-hop systems are considered. With the increased hopping rate as described above, the transmitter can cover 16 different hop frequencies in 16 slots or 10 ms. The page hopping sequence is divided over two paging trains **A** and **B** of 16 frequencies. Train **A** includes the 16 hop frequencies surrounding the current, predicted hop frequency f(k), where k is determined by the clock estimate $CLKE_{16\text{-}12}$. So the first train consists of hops

f(k-8), f(k-7),...,f(k),...,f(k+7)

When the difference between the Bluetooth clocks of the master and the slave is between -8x1.28 s and +7x1.28 s, one of the frequencies used by the master will be the hop frequency the slave will listen to. However, since the master does not know when the slave will enter the **page scan** substate, he has to repeat this train **A** N_{page} times or until a response is obtained. If the slave scan interval corresponds to R1, the repetition number is at least 128; if the slave scan interval corresponds to R2, the repetition number is at least 256.
Note that $CLKE_{16-12}$ changes every 1.28 s; therefore, every 1.28 s, the trains will include different frequencies of the page hopping set.

When the difference between the Bluetooth clocks of the master and the slave is less than -8x1.28 s or larger than +7x1.28 s, more distant hops must be probed. Since in total, there are only 32 dedicated wake-up hops, the more distant hops are the remaining hops not being probed yet. The remaining 16 hops are used to form the new 10 ms train **B.** The second train consists of hops

f(k-16), f(k-15),...,f(k-9),f(k+8)...,f(k+15)

Train **B** is repeated for N_{page} times. If still no response is obtained, the first train **A** is tried again N_{page} times. Alternate use of train A and train B is continued until a response is received or the timeout *pageTO* is exceeded. If during one of the listening occasions, a response is returned by the slave, the master unit enters the **master response** substate.

The description for paging and **page scan** procedures given here has been tailored towards the 79-hop systems used in the US and Europe. For the 23-hop systems as used in France, the procedure is slightly different. In the 23-hop case, the length of the page hopping sequence is reduced to 16. As a consequence, there is only a single train (train **A**) including all the page hopping frequencies. The phase to the page hopping sequence is not $CLKE_{16-12}$ but $CLKE_{15-12}$. An estimate of the slave's clock does not have to be made.

The **page** substate can be entered from the **STANDBY** state or the **CONNECTION** state. In the **STANDBY** state, no connection has been established and the unit can use all the capacity to carry out the page. Before entering the page substate from the CONNECTION state, the unit shall free as much capacity as possible for scanning. To ensure this, it is recommended that the ACL connections are put on hold or park. However, the SCO connections shall not be disturbed by the page. This means that the page will be interrupted by the reserved SCO slots which have higher priority than the page. In order to obtain as much capacity for paging, it is recommended to use the SCO packets which use the least amount of capacity (**HV3** packets). If SCO links are present, the repetition number N_{page} of a single train shall be increased, see Table 10.2. Here it has been assumed that the **HV3** packet are used with an interval T_{SCO}=6 slots, which would correspond to a 64 kb/s voice link.

SR mode	no SCO link	one SCO link (HV3)	two SCO links (HV3)
R0	$N_{page} \geq 1$	$N_{page} \geq 2$	$N_{page} \geq 3$
R1	$N_{page} \geq 128$	$N_{page} \geq 256$	$N_{page} \geq 384$
R2	$N_{page} \geq 256$	$N_{page} \geq 512$	$N_{page} \geq 768$

Table 10.2: Relationship between train repetition, and paging modes R0, R1 and R2 when SCO links are present.

The construction of the page train is independent on the presence of SCO links; that is, SCO packets are sent on the reserved slots but do not affect the hop frequencies used in the unreserved slots, see Figure 10.5 on page 100.

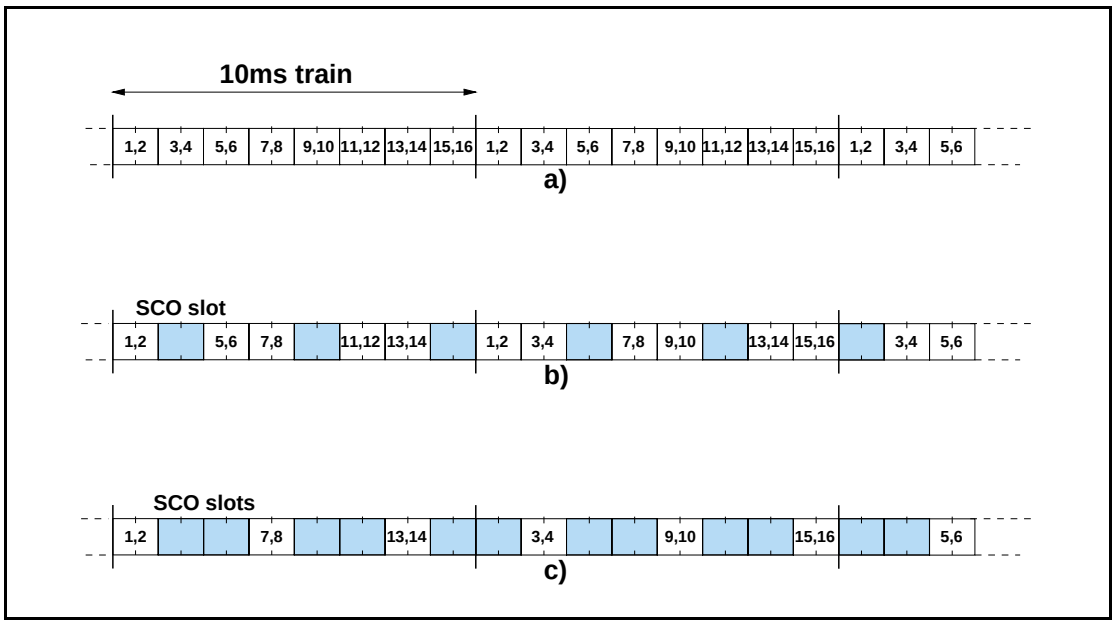


Figure 10.5: Conventional page (a), page while one SCO link present (b), page while two SCO links present (c).

For the descriptions of optional paging schemes see “Appendix VII” on page 1025.

10.6.4 Page response procedures

When a page message is successfully received by the slave, there is a coarse FH synchronization between the master and the slave. Both the master and the slave enter a response routine to exchange vital information to continue the connection setup. Important for the piconet connection is that both Bluetooth units use the same channel access code, use the same channel hopping sequence, and that their clocks are synchronized. These parameters are derived from the master unit. The unit that initializes the connection (starts paging) is defined as the master unit (which is thus only valid during the time the piconet exists). The channel access code and channel hopping sequence are

derived from the Bluetooth device address (BD_ADDR) of the master. The timing is determined by the master clock. An offset is added to the slave's native clock to temporarily synchronize the slave clock to the master clock. At start-up, the master parameters have to be transmitted from the master to the slave. The messaging between the master and the slave at start-up will be considered in this section.

The initial messaging between master and slave is shown in Table 10.3 on page 101 and in Figure 10.6 on page 102 and Figure 10.7 on page 102. In those two figures frequencies f (k), f(k+1), etc. are the frequencies of the page hopping sequence determined by the slave's BD_ADDR. The frequencies f'(k), f'(k+1), etc. are the corresponding page_response frequencies (slave-to-master). The frequencies g(m) belong to the channel hopping sequence.

Step	Message	Direction	Hopping Sequence	Access Code and Clock
1	slave ID	master to slave	page	slave
2	slave ID	slave to master	page response	slave
3	FHS	master to slave	page	slave
4	slave ID	slave to master	page response	slave
5	1st packet master	master to slave	channel	master
6	1st packet slave	slave to master	channel	master

Table 10.3: Initial messaging during start-up.

In step 1 (see Table 10.3 on page 101), the master unit is in **page** substate and the slave unit in the **page scan** substate. Assume in this step that the page message (= slave's device access code) sent by the master reaches the slave. On recognizing its device access code, the slave enters the **slave response** in step 2. The master waits for a reply from the slave and when this arrives in step 2, it will enter the **master response** in step 3. Note that during the initial message exchange, all parameters are derived from the slave's BD_ADDR, and that only the page hopping and page_response hopping sequences are used (which are also derived from the slave's BD_ADDR). Note that when the master and slave enter the response states, their clock input to the page and page_response hop selection is frozen as is described in Section 11.3.3 on page 135.

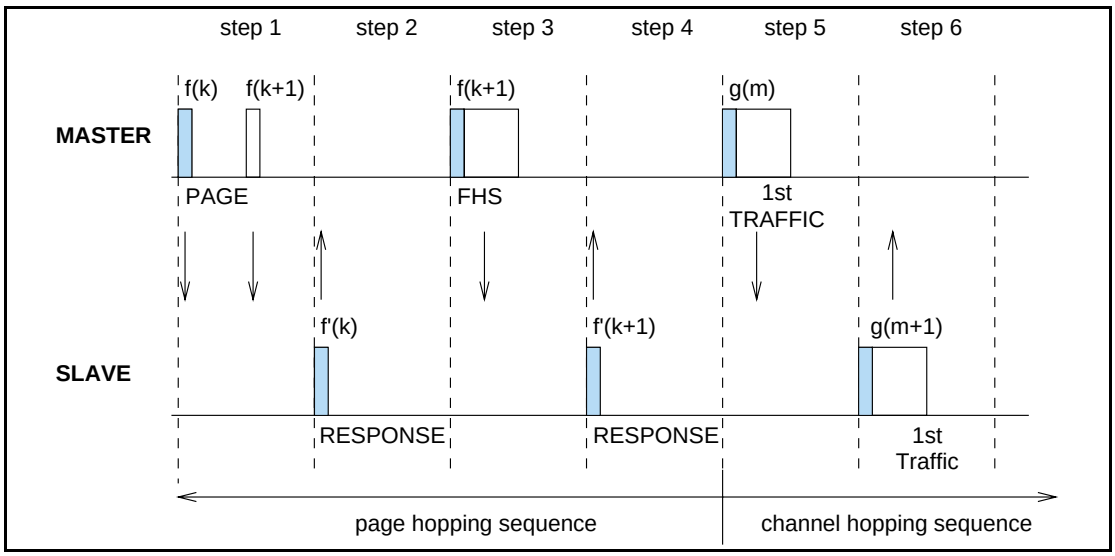


Figure 10.6: Messaging at initial connection when slave responds to first page message.

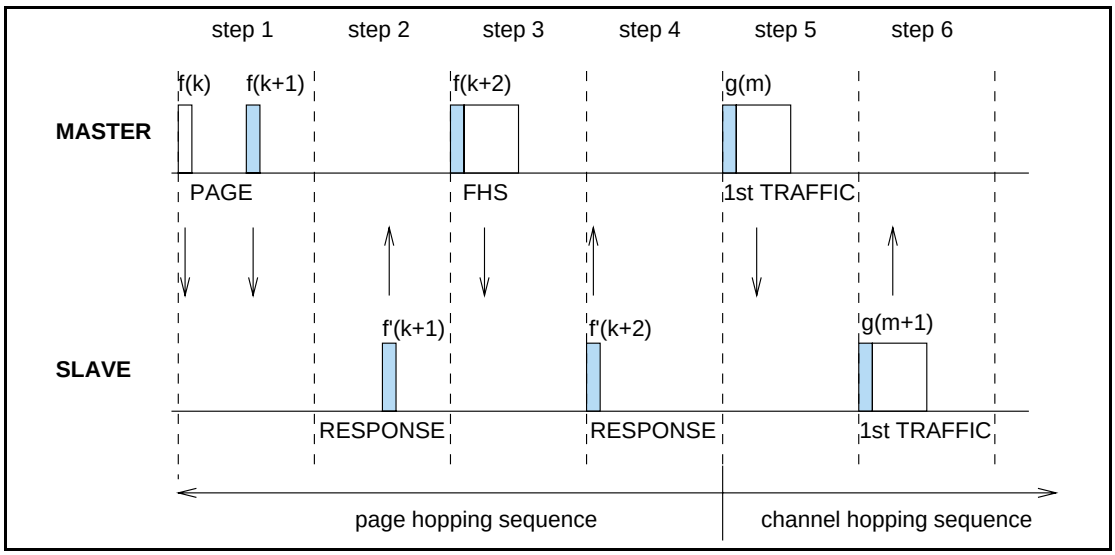


Figure 10.7: Messaging at initial connection when slave responds to second page message.

10.6.4.1 Slave response

After having received its own device access code in step 1, the slave unit transmits a response message in step 2. This response message again only consists of the slave's device access code. The slave will transmit this response 625 µs after the beginning of the received page message (slave ID packet) and at the response hop frequency that corresponds to the hop frequency in which the page message was received. The slave transmission is therefore time aligned to the master transmission. During initial messaging, the slave still uses the page response hopping sequence to return information to the master. The clock input $CLKN_{16\text{-}12}$ is frozen at the value it had at the time the page message was received.

After having sent the response message, the slave's receiver is activated (312.5 µs after the start of the response message) and awaits the arrival of a **FHS** packet. Note that a **FHS** packet can already arrive 312.5 µs after the arrival of the page message as shown in Figure 10.7 on page 102, and not after 625 µs as is usually the case in the RX/TX timing. More details about the timing can be found in Section 9.6 on page 89.

If the setup fails before the **CONNECTION** state has been reached, the following procedure is carried out. The slave will keep listening as long as no **FHS** packet is received until *pagerespTO* is exceeded. Every 1.25 ms, however, it will select the next master-to-slave hop frequency according to the page hop sequence. If nothing is received after *pagerespTO*, the slave returns back to the **page scan** substate for one scan period. Length of the scan period depends on the SCO slots present. If no page message is received during this additional scan period, the slave will resume scanning at its regular scan interval and return to the state it was in prior to the first page scan state.

If a **FHS** packet is received by the slave in the **slave response** substate, the slave returns a response (slave's device access code only) in step 4 to acknowledge the reception of the **FHS** packet (still using the page response hopping sequence). The transmission of this response packet is based on the reception of the **FHS** packet. Then the slave changes to the channel (master's) access code and clock as received from the **FHS** packet. Only the 26 MSBs of the master clock are transferred: the timing is assumed such that CLK_1 and CLK_0 are both zero at the time the **FHS** packet was received as the master transmits in even slots only. From the master clock in the **FHS** packet, the offset between the master's clock and the slave's clock is determined and reported to the slave's link manager.

Finally, the slave enters the **CONNECTION** state in step 5. From then on, the slave will use the master's clock and the master BD_ADDR to determine the channel hopping sequence and the channel access code. The connection mode starts with a POLL packet transmitted by the master. The slave responds with any type of packet. If the POLL packet is not received by the slave, or the response packet is not received by the master, within *newconnectionTO* number of slots after FHS packet acknowledgement, the master and the slave will return to page and page scan substates, respectively.See Section 10.8 on page 110

10.6.4.2 Master response

When the master has received a response message from the slave in step 2, it will enter the **master response** routine. It freezes the current clock input to the page hop selection scheme. Then the master will transmit a **FHS** packet in step 3 containing the master's real-time Bluetooth clock, the master's 48-bit BD_ADDR address, the BCH parity bits, and the class of device. The **FHS** packet contains all information to construct the channel access code without requiring a mathematical derivation from the master device address. The **FHS** packet is transmitted at the beginning of the master-to-slave slot following the slot in which the slave has responded. So the TX timing of the **FHS** is not based on the reception of the response packet from the slave. The **FHS** packet may therefore be sent 312.5 µs after the reception of the response packet like shown in Figure 10.7 on page 102 and not 625 µs after the received packet as is usual in the RX/TX timing, see also Section 9.6 on page 89.

After the master has sent its **FHS** packet, it waits for a second response from the slave in step 4 which acknowledges the reception of the **FHS** packet. Again this is only the slave's device access code. If no response is received, the master retransmits the **FHS** packet, but with an updated clock and still using the slave's parameters. It will retransmit (the clock is updated every retransmission) until a second slave response is received, or the timeout of *pagerespTO* is exceeded. In the latter case, the master turns back to the **page** substate and sends an error message to the link manager. During the retransmissions of the **FHS** packet, the master keeps using the page hopping sequence.

If the slave's response is indeed received, the master changes to the master parameters, so the channel access code and the master clock. The lower clock bits CLK_0 and CLK_1 are zero at the start of the **FHS** packet transmission and are not included in the **FHS** packet. Finally, the master enters the **CONNECTION** state in step 5. The master BD_ADDR is used to change to a new hopping sequence, the *channel hopping sequence*. The channel hopping sequence uses all 79 hop channels in a (pseudo) random fashion, see also Section 11.3.6 on page 137. The master can now send its first traffic packet in a hop determined with the new (master) parameters. This first packet will be a POLL packet. See Section 10.8 on page 110.

The master can now send its first traffic packet in a hop determined with the new (master) parameters. The first packet in this state is a POLL packet sent by the master. This packet will be sent within *newconnectionTO* number of slots after reception of the FHS packet acknowledgement. The slave will respond with any type of packet. If the POLL packet is not received by the slave or the POLL packet response is not received by the master within *newconnectionTO* number of slots, the master and the slave will return to page and page scan substates, respectively.

10.7 INQUIRY PROCEDURES

10.7.1 General

In the Bluetooth system, an inquiry procedure is defined which is used in applications where the destination's device address is unknown to the source. One can think of public facilities like printers or facsimile machines, or access points to a LAN. Alternatively, the inquiry procedure can be used to discover which other Bluetooth units are within range. During an **inquiry** substate, the discovering unit collects the Bluetooth device addresses and clocks of all units that respond to the inquiry message. It can then, if desired, make a connection to any one of them by means of the previously described page procedure.

The inquiry message broadcast by the source does not contain any information about the source. However, it may indicate which class of devices should respond. There is one general inquiry access code (GIAC) to inquire for any Bluetooth device, and a number of dedicated inquiry access codes (DIAC) that only inquire for a certain type of devices. The inquiry access codes are derived from reserved Bluetooth device addresses and are further described in Section 4.2.1 on page 48.

A unit that wants to discover other Bluetooth units enters an **inquiry** substate. In this substate, it continuously transmits the inquiry message (which is the ID packet, see Section 4.4.1.1 on page 55) at different hop frequencies. The **inquiry** hop sequence is always derived from the LAP of the GIAC. Thus, even when DIACs are used, the applied hopping sequence is generated from the GIAC LAP. A unit that allows itself to be discovered, regularly enters the **inquiry scan** substate to respond to inquiry messages. The following sections describe the message exchange and contention resolution during inquiry response. The inquiry response is optional: a unit is not forced to respond to an inquiry message.

10.7.2 Inquiry scan

The **inquiry scan** substate is very similar to the **page scan** substate. However, instead of scanning for the unit's device access code, the receiver scans for the inquiry access code long enough to completely scan for 16 inquiry frequencies. The length of this scan period is denoted $T_{w\_inquiry\_scan}$. The scan is performed at a single hop frequency. As in the page procedure, the inquiry procedure uses 32 dedicated inquiry hop frequencies according to the *inquiry hopping sequence*. These frequencies are determined by the general inquiry address. The phase is determined by the native clock of the unit carrying out the **inquiry scan**; the phase changes every 1.28s.

Instead or in addition to the general inquiry access code, the unit may scan for one or more dedicated inquiry access codes. However, the scanning will follow the inquiry scan hopping sequence which is determined by the general inquiry address. If an inquiry message is recognized during an inquiry wake-up period, the Bluetooth unit either performs a backoff in **CONNECTION** or **STANDBY** state before reentering the inquiry scan substate or enters the **inquiry response** substate if a random backoff was performed before entering the inquiry scan substate..

The **inquiry scan** substate can be entered from the **STANDBY** state or the **CONNECTION** state. In the **STANDBY** state, no connection has been established and the unit can use all the capacity to carry out the **inquiry scan**. Before entering the **inquiry scan** substate from the **CONNECTION** state, the unit preferably reserves as much capacity as possible for scanning. If desired, the unit may place ACL connections in the HOLD mode or even use the PARK mode, see Section 10.8.3 on page 112. SCO connections are preferably not interrupted by the **inquiry scan**. In this case, the **inquiry scan** may be interrupted by the reserved SCO slots which have higher priority than the **inquiry scan**. SCO packets should be used requiring the least amount of capacity (**HV3** packets). The scan window, $T_{w\ inquiry\ scan}$, shall be increased to increase the probability to respond to an inquiry message. If one SCO link is present using HV3 packets and T_{SCO}=6 slots, a total scan window of at least 36 slots (22.5ms) is recommended; if two SCO links are present using HV3 packets and T_{SCO}=6 slots, a total scan window of at least 54 slots (33.75ms) is recommended.

The scan interval $T_{inquiry\ scan}$ is defined as the interval between two consecutive inquiry scans. The **inquiry scan** interval shall be at most 2.56 s.

10.7.3 Inquiry

The **inquiry** substate is used by the unit that wants to discover new devices. This substate is very similar to the **page** substate, the same TX/RX timing is used as used for paging, see Section 9.6 on page 89 and Figure 9.4 on page 88. The TX and RX frequencies follow the inquiry hopping sequence and the inquiry response hopping sequence, and are determined by the general inquiry access code and the native clock of the discovering device. In between inquiry transmissions, the Bluetooth receiver scans for inquiry response messages. When found, the entire response packet (which is in fact a **FHS** packet) is read, after which the unit continues with the inquiry transmissions. So the Bluetooth unit in an **inquiry** substate does not acknowledge the inquiry response messages. It keeps probing at different hop channels and in between listens for response packets. Like in the **page** substate, two 10 ms trains **A** and **B** are defined, splitting the 32 frequencies of the inquiry hopping sequence into two 16-hop parts. A single train must be repeated for at least $N_{inquiry}$=256 times before a new train is used. In order to collect all responses in an error-free environment, at least three train switches must have taken place. As a result, the **inquiry** substate may have to last for 10.24 s unless the inquirer collects enough responses and determines to abort the inquiry substate earlier. If desired, the inquirer can also prolong the inquiry substate to increase the probability of receiving all responses in an error-prone environment. If an inquiry procedure is automatically initiated periodically (say a 10 s period every minute), then the interval between two inquiry instances must be determined randomly. This is done to avoid two Bluetooth units to synchronize their inquiry procedures.

The **inquiry** substate is continued until stopped by the Bluetooth link manager (when it decides that it has sufficient number of responses), or when a timeout has been reached (*inquiryTO*).

The **inquiry** substate can be entered from the **STANDBY** state or the **CONNECTION** state. In the **STANDBY** state, no connection has been established and the unit can use all the capacity to carry out the inquiry. Before entering the inquiry substate from the **CONNECTION** state, the unit shall free as much capacity as possible for scanning. To ensure this, it is recommended that the ACL connections are put on hold or park. However, the SCO connections shall not be disturbed by the inquiry. This means that the inquiry will be interrupted by the reserved SCO slots which have higher priority than the inquiry. In order to obtain as much capacity for inquiry, it is recommended to use the SCO packets which use the least amount of capacity (**HV3** packets). If SCO links are present, the repetition number $N_{inquiry}$ shall be increased, see Table 10.4 on page 108.
Here it has been assumed that the **HV3** packet are used with an interval T_{SCO}=6 slots, which would correspond to a 64 kb/s voice link.

	no SCO link	one SCO link (HV3)	two SCO links (HV3)
$N_{inquiry}$	≥ 256	≥ 512	≥ 768

Table 10.4: Increase of train repetition when SCO links are present.

10.7.4 Inquiry response

For the inquiry operation, there is only a slave response, no master response. The master listens between inquiry messages for responses, but after reading a response, it continues to transmit inquiry messages. The slave response routine for inquiries differs completely from the slave response routine applied for pages. When the inquiry message is received in the **inquiry scan** substate, a response message containing the recipient's address must be returned. This response message is a conventional **FHS** packet carrying the unit's parameters. However, a contention problem may arise when several Bluetooth units are in close proximity to the inquiring unit and all respond to an inquiry message at the same time. First of all, every Bluetooth unit has a free running clock; therefore, it is highly unlikely that they all use the same phase of the inquiry hopping sequence. However, in order to avoid collisions between units that do wake up in the same inquiry hop channel simultaneously, the following protocol in the slave's **inquiry response** is used. If the slave receives an inquiry message, it generates a random number RAND between 0 and 1023. The slave then returns to the **CONNECTION** or **STANDBY** state for the duration of RAND time slots. Before returning to the **CONNECTION** or **STANDBY** state, the unit may go through the page scan substate; this page scan must use the mandatory page scan scheme. After at least RAND slots, the unit will return to the **inquiry scan** substate. On the first inquiry message received in this substate the slave goes into the **inquiry response** substate and returns an **FHS** response packet to the master 625 µs after the inquiry message was received. If during the scan no trigger occurs within a timeout period of *inqrespTO*, the slave returns to the **STANDBY** or **CONNECTION** state. If the unit does receive an inquiry message and returns an FHS packet, it adds an offset of 1 to the phase in the inquiry hop sequence (the phase has a 1.28 s resolution) and enters the **inquiry scan** substate again. If the slave is triggered again, it repeats the procedure using a new RAND. The offset to the clock accumulates each time a **FHS** packet is returned. During a 1.28 s probing window, a slave on average responses 4 times, but on different frequencies and at different times. Possible SCO slots should have priority over response packets; that is, if a response packet overlaps with an SCO slot, it is not sent but the next inquiry message is awaited.

The messaging during the inquiry routines is summarized in Table 10.5 on page 109. In step 1, the master transmits an inquiry message using the inquiry access code and its own clock. The slave responds with the **FHS** packet which contains the slave's device address, native clock and other slave information. This **FHS** packet is returned at a semi-random time. The **FHS** packet is not acknowledged in the inquiry routine, but it is retransmitted at other times and frequencies as long as the master is probing with inquiry messages.

step	message	direction	hopping sequence	access code
1	ID	master to slave	inquiry	inquiry
2	FHS	slave to master	inquiry response	inquiry

Table 10.5: Messaging during inquiry routines.

If the scanning unit uses an optional scanning scheme, after responding to an inquiry with an FHS packet, it will perform page scan using the mandatory page scan scheme for $T_{mandatory\ pscan}$ period. Every time an inquiry response is sent the unit will start a timer with a timeout of $T_{mandatory\ pscan}$. The timer will be reset at each new inquiry response. Until the timer times out, when the unit performs page scan, it will use the mandatory page scanning scheme in the SR mode it uses for all its page scan intervals. Using the mandatory page scan scheme after the inquiry procedure enables all units to connect even if they do not support an optional paging scheme (yet). In addition to using the mandatory page scan scheme, an optional page scan scheme can be used in parallel for the $T_{mandatory\ pscan}$ period.

The $T_{mandatory\ pscan}$ period is included in the SP field of the FHS packet returned in the inquiry response routine, see Section 4.4.1.4 on page 56. The value of the period is indicated in the Table 10.6

SP mode	$T_{mandatory\ pscan}$
P0	≥20s
P1	≥ 40s
P2	≥ 60s
Reserved	-

Table 10.6: Mandatory scan periods for P0, P1, P2 scan period modes.

10.8 CONNECTION STATE

In the **CONNECTION** state, the connection has been established and packets can be sent back and forth. In both units, the channel (master) access code and the master Bluetooth clock are used. The hopping scheme uses the *channel hopping sequence*. The master starts its transmission in even slots (CLK_{1-0}=00), the slave starts its transmission in odd slots (CLK_{1-0}=10)

The **CONNECTION** state starts with a POLL packet sent by the master to verify the switch to the master's timing and channel frequency hopping. The slave can respond with any type of packet. If the slave does not receive the POLL packet or the master does not receive the response packet for *newconnectionTO* number of slots, both devices will return to **page**/**page scan** substates.

The first information packets in the **CONNECTION** state contain control messages that characterize the link and give more details regarding the Bluetooth units. These messages are exchanged between the link managers of the units. For example, it defines the SCO links and the sniff parameters. Then the transfer of user information can start by alternately transmitting and receiving packets.

The **CONNECTION** state is left through a **detach** or **reset** command. The **detach** command is used if the link has been disconnected in the normal way. All configuration data in the Bluetooth link controller is still valid. The **reset** command is a hard reset of all controller processes. After a reset, the controller has to be reconfigured.

The Bluetooth units can be in several modes of operation during the **CONNECTION** state: active mode, sniff mode, hold mode, and park mode. These modes are now described in more detail.

10.8.1 Active mode

In the active mode, the Bluetooth unit actively participates on the channel. The master schedules the transmission based on traffic demands to and from the different slaves. In addition, it supports regular transmissions to keep slaves synchronized to the channel. Active slaves listen in the master-to-slave slots for packets. If an active slave is not addressed, it may sleep until the next new master transmission. From the type indication in the packet, the number of slots the master has reserved for its transmission can be derived; during this time, the non-addressed slaves do not have to listen on the master-to-slave slots. A periodic master transmission is required to keep the slaves synchronized to the channel. Since the slaves only need the channel access code to synchronize with, any packet type can be used for this purpose.

10.8.2 Sniff mode

In the sniff mode, the duty cycle of the slave's listen activity can be reduced. If a slave participates on an ACL link, it has to listen in every ACL slot to the master traffic. With the sniff mode, the time slots where the master can start transmission to a specific slave is reduced; that is, the master can only start transmission in specified time slots. These so-called sniff slots are spaced regularly with an interval of T_{sniff}.

The slave starts listening at the sniff slots for $N_{sniff\ attempt}$ consecutive receive slots unless a packet with matching AM_ADDR is received. After every reception of a packet with matching AM_ADDR, the slave continues listening at the subsequent $N_{sniff\ timeout}$ or remaining of the receive slots, whichever is greater. So, for $N_{sniff\ timeout} > 0$, the slave continues listening as long as it receives packets with matching AM_ADDR.

Note that Receive slots here are every odd-numbered slots, in which the master may start sending a packet.

Note that $N_{sniff\ attempt} = 1$ and $N_{sniff\ timeout} = 0$ cause the slave to listen only at the first sniff slot, irrespective of packets received from the master.

Note that $N_{sniff\ attempt} = 0$ is not allowed.

To enter the sniff mode, the master or slave shall issue a sniff command via the LM protocol. This message will contain the sniff interval T_{sniff} and an offset D_{sniff}. The timing of the sniff mode is then determined similar as for the SCO links. In addition, an initialization flag indicates whether initialization procedure 1 or 2 is being used. The device uses initialization 1 when the MSB of the current master clock (CLK_{27}) is 0; it uses initialization 2 when the MSB of the current master clock (CLK_{27}) is 1. The slave shall apply the initialization method as indicated by the initialization flag irrespective of its clock bit value CLK_{27}. The master-to-slave sniff slots determined by the master and the slave shall be initialized on the slots for which the clock satisfies the following equation

$$CLK_{27-1} \bmod T_{sniff} = D_{sniff} \qquad \text{for initialization 1}$$

$$(\overline{CLK_{27}}, CLK_{26-1}) \bmod T_{sniff} = D_{sniff} \qquad \text{for initialization 2}$$

The slave-to-master sniff slot determined by the master and the slave shall be initialized on the slots after the master-to-slave sniff slot defined above. After initialization, the clock value CLK(k+1) for the next master-to-slave SNIFF slot is found by adding the fixed interval T_{sniff} to the clock value of the current master-to-slave sniff slot:

$$CLK(k+1) = CLK(k) + T_{sniff}$$

10.8.3 Hold mode

During the **CONNECTION** state, the ACL link to a slave can be put in a **hold** mode. This means that the slave temporarily does not support ACL packets on the channel any more (note: possible SCO links will still be supported). With the **hold** mode, capacity can be made free to do other things like scanning, paging, inquiring, or attending another piconet. The unit in **hold** mode can also enter a low-power sleep mode. During the **hold** mode, the slave unit keeps its active member address (AM_ADDR).

Prior to entering the hold mode, master and slave agree on the time duration the slave remains in the hold mode. A timer is initialized with the *holdTO* value. When the timer is expired, the slave will wake up, synchronize to the traffic on the channel and will wait for further master instructions.

10.8.4 Park mode

When a slave does not need to participate on the piconet channel, but still wants to remain synchronized to the channel, it can enter the park mode which is a low-power mode with very little activity in the slave. In the park mode, the slave gives up its active member address AM_ADDR. Instead, it receives two new addresses to be used in the park mode

- PM_ADDR: 8-bit Parked Member Address
- AR_ADDR: 8-bit Access Request Address

The PM_ADDR distinguishes a parked slave from the other parked slaves. This address is used in the master-initiated unpark procedure. In addition to the PM_ADDR, a parked slave can also be unparked by its 48-bit BD_ADDR. The all-zero PM_ADDR is a reserved address: if a parked unit has the all-zero PM_ADDR it can only be unparked by the BD_ADDR. In that case, the PM_ADDR has no meaning. The AR_ADDR is used by the slave in the slave-initiated unpark procedure. All messages sent to the parked slaves have to be carried by broadcast packets (the all-zero AM_ADDR) because of the missing AM_ADDR.

The parked slave wakes up at regular intervals to listen to the channel in order to re-synchronize and to check for broadcast messages. To support the synchronization and channel access of the parked slaves, the master supports a beacon channel described in the next section. The beacon structure is communicated to the slave when it is being parked. When the beacon structure changes, the parked slaves are updated through broadcast messages.

In addition for using it for low power consumption, the park mode is used to connect more than seven slaves to a single master. At any one time, only seven slaves can be active. However, by swapping active and parked slaves out respectively in the piconet, the number of slave virtually connected can be much larger (255 if the PM_ADDR is used, and even a larger number if the BD_ADDR is used). There is no limitation to the number of slaves that can be parked.

10.8.4.1 Beacon channel

To support parked slaves, the master establishes a beacon channel when one or more slaves are parked. The beacon channel consists of one beacon slot or a train of equidistant beacon slots which is transmitted periodically with a constant time interval. The beacon channel is illustrated in Figure 10.8 on page 114. A train of N_B ($N_B \geq 1$) beacon slots is defined with an interval of T_B slots. The beacon slots in the train are separated by Δ_B. The start of the first beacon slot is referred to as the **beacon instant** and serves as the beacon timing reference. The beacon parameters N_B and T_B are chosen such that there are sufficient beacon slots for a parked slave to synchronize to during a certain time window in an error-prone environment.

When parked, the slave will receive the beacon parameters through an LMP command. In addition, the timing of the beacon instant is indicated through the offset D_B. Like for the SCO link (see Section 3.2 on page 45), two initialization procedures 1 or 2 are used. The master uses initialization 1 when the MSB of the current master clock (CLK_{27}) is 0; it uses initialization 2 when the MSB of the current master clock (CLK_{27}) is 1. The chosen initialization procedure is also carried by an initialization flag in the LMP command. The slave shall apply the initiations method as indicated by the initialization flag irrespective of its clock bit CLK_{27}. The master-to-slave slot positioned at the beacon instant shall be initialized on the slots for which the clock satisfies the following equation

$$CLK_{27-1} \bmod T_B = D_B \qquad \text{for initialization 1}$$

$$(\overline{CLK_{27}}, CLK_{26-1}) \bmod T_B = D_B \qquad \text{for initialization 2}$$

After initialization, the clock value CLK(k+1) for the next beacon instant is found by adding the fixed interval T_B to the clock value of the current beacon instant:

$$CLK(k+1) = CLK(k) + T_B$$

The beacon channel serves four purposes:

1. transmission of master-to-slave packets which the parked slaves can use for re-synchronization
2. carrying messages to the parked slaves to change the beacon parameters
3. carrying general broadcast messages to the parked slaves
4. unparking of one or more parked slaves

Since a slave can synchronize to any packet which is preceded by the proper channel access code, the packets carried on the beacon slots do not have to contain specific broadcast packets for parked slaves to be able to synchronize; any packet can be used. The only requirement placed on the beacon slots is that there is master-to-slave transmission present. If there is no information to be sent, **NULL** packets can be transmitted by the master. If there is indeed

broadcast information to be sent to the parked slaves, the first packet of the broadcast message shall be repeated in every beacon slot of the beacon train. However, synchronous traffic like on the SCO link, may interrupt the beacon transmission.

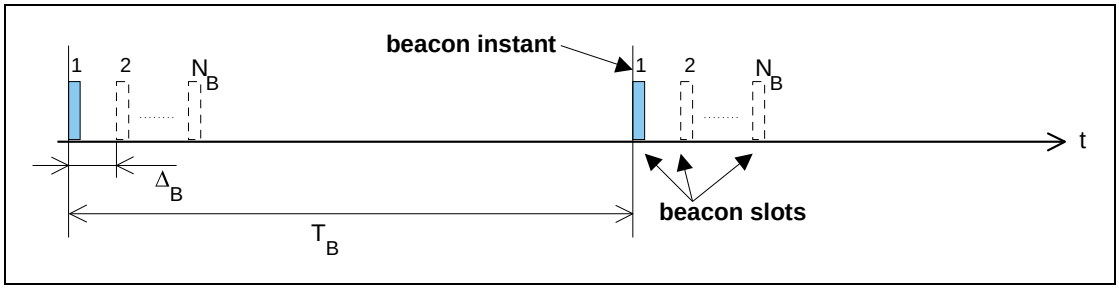


Figure 10.8: General beacon channel format

The master communicates with parked slaves using broadcast messages. Since these messages can be time - critical, an ongoing repetition train of broadcast message can be prematurely aborted by broadcast information destined to parked slaves in beacon slots and in access windows (see Section 10.8.4.2 on page 114).

10.8.4.2 Beacon access window

In addition to the beacon slots, an access window is defined where the parked slaves can send requests to be unparked. To increase reliability, the access window can be repeated M_{access} times ($M_{access} \geq 1$), see Figure 10.9 on page 114. The access window starts a fixed delay D_{access} after the beacon instant. The width of the access window is T_{access}.

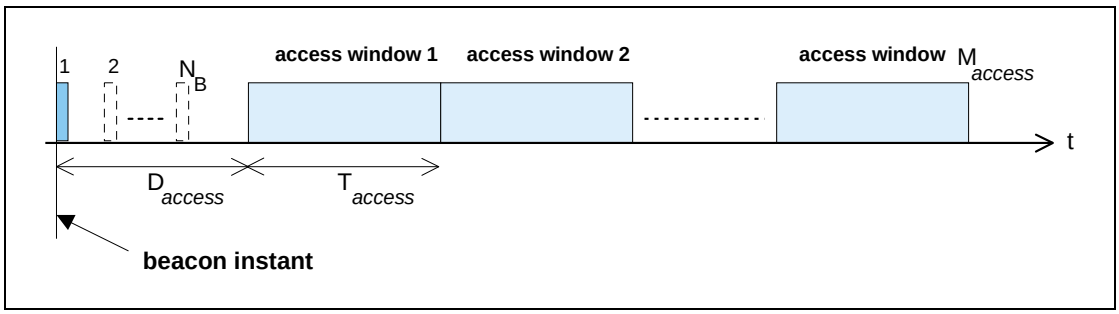


Figure 10.9: Definition of access window

The access window may support different slave access techniques, like polling, random access, or other forms of access. At this stage, only the polling technique has been defined. The format of the polling technique is shown in Figure 10.10 on page 115. The same TDD structure is used as on the piconet channel, i.e. master-to-slave transmission is alternated by slave-to-master transmission. The slave-to-master slot is divided into two half slots of 312.5 µs each. The half slot a parked slave is allowed to respond in corresponds to its access

request address (AR_ADDR), see also section 10.8.4.6 on page 118. For counting the half slots to determine the access request slot, the start of the access window is used, see Figure 10.10 on page 115. The slave is only allowed to send an access request in the proper slave-to-master half slot if in the preceding master-to-slave slot a broadcast packet has been received. In this way, the master polls the parked slaves.

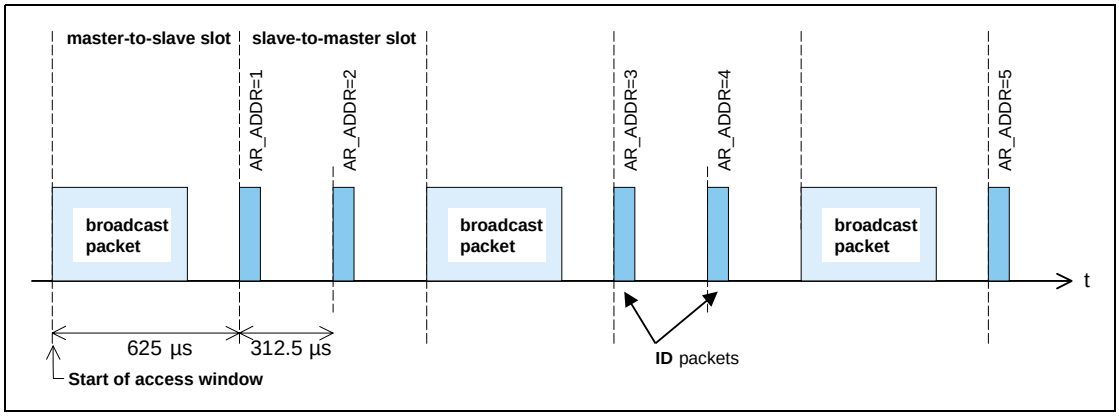


Figure 10.10: Access procedure applying the polling technique.

However, the slots of the access window can also be used for traffic on the piconet if required. For example, if an SCO connection has to be supported, the slots reserved for the SCO link may carry SCO information instead of being used for access requests, i.e. if the master-to-slave slot in the access window contains a packet different from a broadcast packet, the following slave-to-master slot cannot be used for slave access requests. Slots in the access window not affected by traffic can still be used according to the defined access structure; an example is shown in Figure 10.11 on page 115: the access procedure is continued as if no interruption had taken place.

When the slave is parked, it is indicated what type of access scheme will be used. For the polling scheme, the number of slave-to-master access slots N_{acc_slot} is indicated.

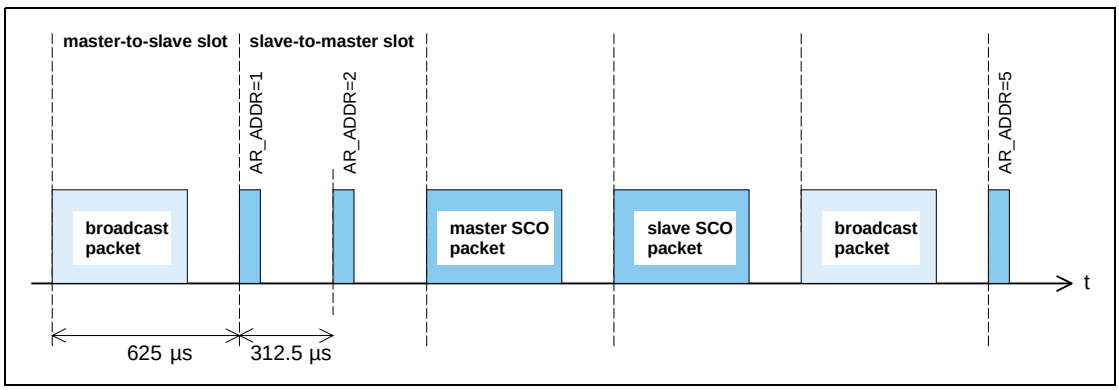


Figure 10.11: Disturbance of access window by SCO traffic

By default, the access window is always present. However, its activation depends on the master sending broadcast messages to the slave at the appropriate slots in the access window. A flag in a broadcast LMP message within the beacon slots may indicate that the access window(s) belonging to this instant will not be activated. This prevents unnecessary scanning of parked slaves that want to request access.

10.8.4.3 Parked slave synchronization

Parked slaves sleep most of the time. However, periodically they wake up to re-synchronize to the channel. Any packet exchanged on the channel can be used for synchronization. Since master transmission is mandatory on the beacon slots, parked slaves will exploit the beacon channel to re-synchronize. A parked slave will wake-up at the beacon instant to read the packet sent on the first beacon slot. If this fails, it will retry on the next beacon slot in the beacon train; in total, there are N_B opportunities per beacon instant to re-synchronize. During the search, the slave may increase its search window, see also Section 9.4 on page 88. The separation between the beacon slots in the beacon train Δ_B is chosen such that consecutive search windows will not overlap.

The parked slave does not have to wake up at every beacon instant. Instead, a sleep interval can be applied which is longer than the beacon interval T_B, see Figure 10.12 on page 117. The slave sleep window must be a multiple N_{B_sleep} of T_B. The precise beacon instant the slave shall wake up on is indicated by the master with D_{B_sleep} which indicates the offset (in multiples of T_B) with respect to the beacon instant ($0 < D_{B\_sleep} < N_{B\_sleep}-1$). To initialize the wake-up period, the following equations are used:

$$CLK_{27-1} \bmod (N_{B\_sleep} \bullet T_B) = D_B + D_{B\_sleep} \bullet T_B \quad \text{for initialization 1}$$

$$(\overline{CLK_{27}}, CLK_{26-1}) \bmod (N_{B\_sleep} \bullet T_B) = D_B + D_{B\_sleep} \bullet T_B \quad \text{for initialization 2}$$

where initialization 1 is chosen by the master if the MSB in the current master clock is 0 and initialization 2 is chosen if the MSB in the current master clock is 1.

When the master wants to send broadcast messages to the parked slaves, it may use the beacon slots for these broadcast messages. However, if $N_B < N_{BC}$, the slots following the last beacon slot in the beacon train shall be used for the remaining $N_{BC}-N_B$ broadcast packets. If $N_B > N_{BC}$, the broadcast message is repeated on all N_B beacon slots.

A parked slave shall at least read the broadcast messages sent in the beacon slot(s) it wakes up in; the minimum wake-up activity is to read the channel access code for re-synchronization and the packet header to check for broadcast messages.

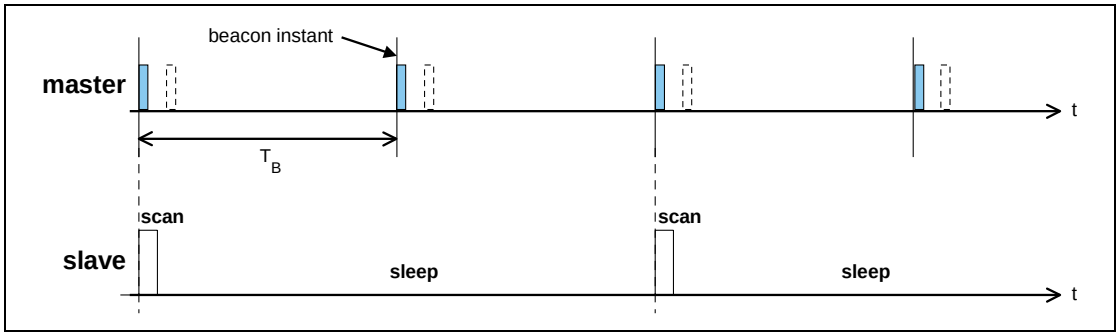


Figure 10.12: Extended sleep interval of parked slaves.

10.8.4.4 Parking

A master can park an active slave through the exchange of one or a few LMP commands. Before put into the park mode, the slave is assigned a PM_ADDR and an AR_ADDR. Every parked slave has a unique PM_ADDR; however, the AR_ADDR is not necessarily unique. Also, the beacon parameters are given by the master when the slave is parked. The slave then gives up its AM_ADDR and enters the park mode. A master can park only a single slave at a time. The park message is carried with a normal data packet and addresses the slave through its AM_ADDR.

10.8.4.5 Master-activated unparking

The master can unpark a parked slave by sending a dedicated LMP unpark command including the parked slave's address. This message is sent in a broadcast packet on the beacon slots. Either the slave's PM_ADDR is used, or its full BD_ADDR is used. The message also includes the active member address AM_ADDR the slave will use after it has re-entered the piconet. The unpark message can include a number of slave addresses so that multiple slaves can be unparked simultaneously. For each slave, a different AM_ADDR is assigned.

After having received the unpark message, the parked slave matching the PM_ADDR or BD_ADDR will leave the park mode and enter the active mode. It will keep listening to the master until it is addressed by the master through its AM_ADDR. The first packet sent by the master mustbe a POLL packet. The return packet in response to the POLL packet confirms that the slave has been unparked. If no response packets from the slave is received for *newconnectionTO* number of slots after the end of beacon repetition period, the master will unpark the slave again. If the slave does not receive the POLL packet for *newconnectionTO* number of slots after the end of beacon repetition period, it will return to park, with the same beacon parameters. After confirming that the slave is active, the master decides in which mode the slave will continue.

10.8.4.6 Slave-activated unparking

A slave can request access to the channel through the access window defined in section 10.8.4.2 on page 114. As shown in Figure 10.10 on page 115, the access window includes several slave-to-master half slots where the slave can send an access request message. The specific half slot the slave is allowed to respond in, corresponds to its access request address (AR_ADDR) which it has received when it was parked. The order of the half slots (in Figure 10.10 the AR_ADDR numbers linearly increase from 1 to 5) is not fixed: an LMP command sent in the beacon slots may reconfigure the access window. When a slave desires access to the channel, it sends an access request message in the proper slave-to-master half slot.The access request message of the slave is the **ID** packet containing the device access code (DAC) of the master (which is in this case the channel access code without the trailer). The parked slave is only allowed to transmit an access request message in the half slot when in the preceding master-to-slave slot, a broadcast packet has been received. This broadcast message can contain any kind of broadcast information not necessarily related to the parked slave(s). If no broadcast information is available, a broadcast **NULL** or broadcast **POLL** packet shall be sent.

After having sent an access request, the parked slave will listen for an unpark message from the master. As long as no unpark message is received, the slave will repeat the access requests in the subsequent access windows. After the last access window (there are M_{access} windows in total, see Section 10.8.4.2 on page 114), the parked slave shall listen for an additional N_{poll} time slots for an unpark message. If no unpark message is received within N_{poll} slots after the end of the last access window, the slave may return to sleep and retry an access attempt after the next beacon instant.

After having received the unpark message, the parked slave matching the PM_ADDR or BD_ADDR will leave the park mode and enter the active mode. It will keep listening to the master until it is addressed by the master through its AM_ADDR. The first packet sent by the master mustbe a POLL packet. The return packet in response to the POLL packet confirms that the slave has been unparked. If no response packet from the slave is received for *newconnectionTO* number of slots after N_{poll} slots after the end of the last access window, the master will send the unpark message to the slave again. If the slave does not receive the POLL packet for *newconnectionTO* number of slots after N_{poll} slots after the end of the last access window, it will return to park, with the same beacon parameters. After confirming that the slave is active, the master decides in which mode the slave will continue.

10.8.4.7 Broadcast scan window

In the beacon train, the master can support broadcast messages to the parked slaves. However, it may extend its broadcast capacity by indicating to the parked slaves that more broadcast information is following after the beacon train. This is achieved by a special LMP command ordering the parked slaves

(as well as the active slaves) to listen to the channel for broadcast messages during a limited time window. This time window starts at the beacon instant and continues for the period as indicated in the LMP command sent in the beacon train.

10.8.5 Polling schemes

10.8.5.1 Polling in active mode

The master always has full control over the piconet. Due to the stringent TDD scheme, slaves can only communicate with the master and not to other slaves. In order to avoid collisions on the ACL link, a slave is only allowed to transmit in the slave-to-master slot when addressed by the AM_ADDR in the packet header in the preceding master-to-slave slot. If the AM_ADDR in the preceding slot does not match, or an AM_ADDR cannot be derived from the preceding slot, the slave is not allowed to transmit.

On the SCO links, the polling rule is slightly modified. The slave is allowed to transmit in the slot reserved for his SCO link unless the (valid) AM_ADDR in the preceding slot indicates a different slave. If no valid AM_ADDR can be derived in the preceding slot, the slave is still allowed to transmit in the reserved SCO slot.

10.8.5.2 Polling in park mode

In the park mode, parked slaves are allowed to send access requests in the access window provided a broadcast packet is received in the preceding master-to-slave slot. Slaves in active mode will not send in the slave-to-master slots following the broadcast packet since they are only allowed to send if addressed specifically.

10.8.6 Slot reservation scheme

The SCO link is established by negotiations between the link managers which involves the exchange of important SCO timing parameters like T_{SCO} and D_{SCO} through LMP messages.

10.8.7 Broadcast scheme

The master of the piconet can broadcast messages which will reach all slaves. A broadcast packet is characterized by the all-zero AM_ADDR. Each new broadcast message (which may be carried by a number of packets) shall start with the flush indication (L_CH=10).

A broadcast packet is never acknowledged. In an error-prone environment, the master may carry out a number of retransmissions to increase the probability for error-free delivery, see also Section 5.3.5 on page 71.

In order to support the **park** mode (as described in Section 10.8.4 on page 112), a master transmission shall take place at fixed intervals. This master transmission will act as a beacon to which slaves can synchronize. If no traffic takes place at the beacon event, broadcast packets shall be sent. More information is given in Section 10.8.4 on page 112.

10.9 SCATTERNET

10.9.1 General

Multiple piconets may cover the same area. Since each piconet has a different master, the piconets hop independently, each with their own channel hopping sequence and phase as determined by the respective master. In addition, the packets carried on the channels are preceded by different channel access codes as determined by the master device addresses. As more piconets are added, the probability of collisions increases; a graceful degradation of performance results as is common in frequency-hopping spread spectrum systems.

If multiple piconets cover the same area, a unit can participate in two or more overlaying piconets by applying time multiplexing.To participate on the proper channel, it should use the associated master device address and proper clock offset to obtain the correct phase. A Bluetooth unit can act as a slave in several piconets, but only as a master in a single piconet: since two piconets with the same master are synchronized and use the same hopping sequence, they are one and the same piconet. A group of piconets in which connections consists between different piconets is called a **scatternet**.

A master or slave can become a slave in another piconet by being paged by the master of this other piconet. On the other hand, a unit participating in one piconet can page the master or slave of another piconet. Since the paging unit always starts out as master, a master-slave role exchange is required if a slave role is desired. This is described in the section 10.9.3 on page 121.

10.9.2 Inter-piconet communications

Time multiplexing must be used to switch between piconets. In case of ACL links only, a unit can request to enter the **hold** or **park** mode in the current piconet during which time it may join another piconet by just changing the channel parameters. Units in the **sniff** mode may have sufficient time to visit another piconet in between the sniff slots. If SCO links are established, other piconets can only be visited in the non-reserved slots in between. This is only possible if there is a single SCO link using **HV3** packets. In the four slots in between, one other piconet can be visited. Since the multiple piconets are not synchronized, guard time must be left to account for misalignment. This means that only 2 slots can effectively be used to visit another piconet in between the **HV3** packets.

Since the clocks of two masters of different piconets are not synchronized, a slave unit participating in two piconets has to take care of two offsets that, added to its own native clock, create one or the other master clock. Since the two master clocks drift independently, regular updates of the offsets are required in order for the slave unit to keep synchronization to both masters.

10.9.3 Master-slave switch

There are several occasions when a master-slave (MS) switch is desirable. Firstly, a MS switch is necessary when a unit paging the master of an existing piconet wants to join this piconet, since, by definition, the paging unit initially is master of a "small" piconet only involving the pager (master) and the paged (slave) unit. Secondly, when a slave in an existing piconet wants to set up a new piconet, involving itself as master and the current piconet master as slave. The latter case implies a double role of the original piconet master; it becomes a slave in the new piconet while still maintaining the original piconet as master. Thirdly, a much more complicated example is when a slave wants to fully take over an existing piconet, i.e., the switch also involves transfer of other slaves of the existing piconet to the new piconet. Clearly, this can be achieved by letting the new master setup a completely new piconet through the conventional paging scheme. However, that would require individual paging of the old slaves, and, thus, take unnecessarily long time. Instead, letting the new master utilize timing knowledge of the old master is more efficient. As a consequence of the MS switch, the slaves in the piconet have to be transferred to the new piconet, changing their timing and their hopping scheme.

The MS switch is described in step1 through step 3 below. Prior to the MS switch, encryption if present, must be stopped in the old piconet. For the third example involving the transfer, new piconet parameters have to be communicated to each slave. The process of this is described in step 4 below. Unfortunately, even though all the hooks are defined for an efficient transfer at baseband level, there are still many issues that lack sufficient support in the higher layers of the Bluetooth specification (such as how to handle security and transfer all kind of slave information from old to new master). Until all levels of the specification fully supports this kind of transfer, this functionality will have to be taken care of at application layer. These transfer procedures are outside the scope of the baseband specification.

The MS switch procedure will now be described in more detail. For the master and slave involved in the role switch, the MS switch results in a reversal of their TX and RX timing: a TDD switch. Moreover, since the piconet parameters are derived from the device address and clock of the master, an MS switch inherently involves a redefinition of the piconet as well: a piconet switch. The new piconet's parameters are derived from the former slave's device address and clock.

Assume unit A wants to become master; unit B was the former master. Then there are basically two alternative scenarios: either the slave takes the MS

switch initiative or the master takes the MS switch initiative. These scenarios are described in Link Manager Protocol, section 3.12 on page 208.

Both slave A and master B do the TDD switch but keep the former hopping scheme (still using the device address and clock of unit B), so there is no piconet switch yet. The slot offset information sent by slave A is not used yet but is used in step 3. Unit A now becomes the master, unit B the slave. The AM_ADDR formerly used by unit A in its slave role, is now used by slave B.

At the moment of the TDD switch, both units A and B will start a timer with a time out of newconnectionTO. The timer is stopped in slave B as soon as it receives an FHS packet from master A on the TDD-switched channel, the timer is stopped in master A as soon as it receives an ID packet from slave B. If the newconnectionTO expires, the master and slave will return to the old piconet timing and take their old role of master and slave. The FHS packet is sent by master A using the "old" piconet parameters. The AM_ADDR in the FHS packet header is the former AM_ADDR used by unit A. The AM_ADDR carried in the FHS payload is the new AM_ADDR intended for unit B when operating on the new piconet. After the FHS acknowledgment, which consists of the ID packet and is sent by the slave on the old hopping sequence, both master A and slave B turn to the new channel parameters of the new piconet as indicated by the FHS.

Since the old and new masters' clocks are synchronous, the clock information sent in the FHS payload should indicate the new master's clock at the begining of the FHS packet transmission. Furthermore, the 1.25 ms resolution of the clock information given in the FHS packet is not sufficient for aligning the slot boundaries of the two piconets. The slot-offset information in the LMP message previously sent by unit A is used to provide more accurate timing information. The slot offset indicates the delay between the start of the master-to-slave slots of the old and new piconet channels. This timing information ranges from 0 to 1249 µs with a resolution of 1µs. It is used together with the clock information in the FHS packet to accurately position the correlation window when switching to the new master's timing after acknowledgment of the FHS packet. After reception of the FHS packet acknowledgment, the new master A switches to its own timing and sends a POLL packet to verify the switch. Both the master and the slave will start a new timer with a time out of newconnectionTO on FHS packet.acknowledgment. The start of this timer shall be aligned with the beginning of the first master TX slot boundary of the new piconet, following the FHS packet acknowledgment. The slave stops the timer when the POLL packet is received; the master stops the timer when the POLL packet is acknowledged. The slave uses a NULL packet to acknowledge the POLL. If no response is received, the master re-sends the POLL packet until newconnectionTO is reached. Should this timer expire, both the slave and the master return to the old piconet timing with the old master and slave roles. The procedure may then start again beginning at step 1. Aligning the timer with TX boundaries of the new piconet ensures that no unit returns to the old piconet timing in the middle of a master RX slot.

If the new master wishes to take over slaves from the old piconet (which were slaves to the old master B), a piconet switch is enforced on each slave separately. Since the existing slaves already have the correct TDD timing, a TDD switch is not required. Master A sends a slot-offset LMP message to indicate the difference in slot timing of the old and new piconet channel. Thereafter, master A sends an FHS packets and waits for an acknowledgment in the form of an ID packet. When sending the FHS packet, the master starts a timer with a time out of newconnectionTO. The timer is stopped in master A as soon as it receives an ID packet from the slave. Transmission of the FHS packet and the acknowledgment is carried out with the "old" piconet parameters of unit B (compare this to the page hopping scheme used during connection establishment, see Section 10.6.4 on page 104). Should the timer in master A expire, it may restart the transfer operation of step 4. After FHS acknowledgment by the slave, the communication to this slave continues with the new device address and clock of unit A. The FHS packet sent to each slave has the old AM_ADDR in the FHS packet header and their new AM_ADDR in the FHS packet payload (the new AM_ADDR may be identical to the old AM_ADDR).

After reception of the FHS packet acknowledgment, the new master A switches to its own timing and hopping sequence and sends a POLL packet to verify the switch. Both the master and the slave will start a timer with a time out of newconnectionTO on FHS packet acknowledgment. The start of this timer shall be aligned with the beginning of the first master TX slot boundary of the new piconet, following the FHS packet acknowledgment. The slave stops the timer when the POLL packet is received; the master stops the timer when the POLL packet is acknowledged. The slave uses a NULL packet to acknowledge the POLL. If no response is received, the master re sends the POLL packet until newconnectionTO is reached. If the timer expires, both the slave and the master return to the old piconet parameters. The procedure of step 4 may then be repeated. If an existing slave is out of the range of the new master, master A cannot switch the slave to the new piconet. In that case, the slave loses the connection with the existing piconet after the TDD switch when master B is replaced with master A. As a result, the first message sent by master A being the slot-offset LMP message is never acknowledged by this slave. Due to the link supervision mechanism, the slave will eventually be detached from the old piconet.

Summarized, the MS-switch takes place in two steps: first a TDD switch of the considered master and slave, followed by a piconet switch of the both participants. Then, if so desired, other slaves of the old piconet can be transferred to the new piconet. When a unit have acknowledged the reception of the FHS packet, this unit uses the new piconet parameters defined by the new master and the piconet switch is completed. Note that the SEQN of the first data packet containing a CRC on the new piconet channel is set to 1, see section 5.3.2 on page 69

A parked slave must be unparked before it can participate in a MS switch. Parked slaves that are transferred to a new piconet shall be activated using the old park parameters, changed to the new piconet parameters, and then returned to the park mode using the new park parameters.

10.10 POWER MANAGEMENT

Features are included into Bluetooth to ensure a low-power operation. These features are both at the microscopic level when handling the packets, and at the macroscopic level using certain operation modes.

10.10.1 Packet handling

In order to minimize power consumption, packet handling is minimized both at TX and RX sides. At the TX side, power is minimized by only sending useful data. This means that if only link control information needs to be exchanged, **NULL** packets will be used. No transmission is carried out at all if there is no link control information or involves a NAK only (NAK is implicit on no reply). If there is data to be sent, the payload length is adapted in order to send only the valid data bytes. At the RX side, packet processing takes place in different steps. If no valid access code is found in the search window, the transceiver returns to sleep. If an access code is found, the receiver unit is woken up and starts to process the packet header. If the HEC fails, the unit will return to sleep after the packet header. A valid header will indicate if a payload will follow and how many slots are involved.

10.10.2 Slot occupancy

As was described in Section 4.4 on page 54, the packet type indicates how many slots a packet may occupy. A slave not addressed in the first slot can go to sleep for the remaining slots the packet may occupy. This can be read from the TYPE code.

10.10.3 Low-power modes

In Section 10.8 on page 110, three modes were described during the **CONNECTION** state which reduce power consumption. If we list the modes in increasing order of power efficiency then the **sniff** mode has the higher duty cycle, followed by the **hold** mode with a lower duty cycle, and finishing with the **park** mode with the lowest duty cycle.

10.11 LINK SUPERVISION

A connection may break down due to various reasons such as a device moving out of range or a power failure condition. Since this may happen without any prior warning, it is important to monitor the link on both the master and the slave side to avoid possible collisions when the AM_ADDR is reassigned to another slave.

To be able to supervise link loss, both the master and the slave use link supervision timers, $T_{supervision}$. Upon reception of a packet that passes the HEC check and has the correct AM_ADDR, the timer is reset. If at any time in con-

nection state, the timer reaches the *supervisionTO* value, the connection is reset. The same timeout value is used for both SCO and ACL connections.

The timeout period, *supervisionTO*, is negotiated at the LM level. Its value is chosen so that the supervision timeout will be longer than hold and sniff periods. Link supervision of a parked slave will be done by unparking and re-parking the slave.

11 HOP SELECTION

In total, 10 types of hopping sequences are defined – five for the 79-hop and five for the 23-hop system, respectively. Using the notation of parentheses () for figures related to the 23-hop system, these sequences are:

- A **page hopping sequence** with 32 (16) unique wake-up frequencies distributed equally over the 79 (23) MHz, with a period length of 32 (16);
- A **page response sequence** covering 32 (16) unique response frequencies that all are in an one-to-one correspondence to the current page hopping sequence. The master and slave use different rules to obtain the same sequence;
- An **inquiry sequence** with 32 (16) unique wake-up frequencies distributed equally over the 79 (23) MHz, with a period length of 32 (16);
- A **inquiry response sequence** covering 32 (16) unique response frequencies that all are in an one-to-one correspondence to the current inquiry hopping sequence.
- A **channel hopping sequence** which has a very long period length, which does not show repetitive patterns over a short time interval, but which distributes the hop frequencies equally over the 79 (23) MHz during a short time interval;

For the page hopping sequence, it is important that we can easily shift the phase forward or backward, so we need a 1-1 mapping from a counter to the hop frequencies. For each case, both a hop sequence from master to slave and from slave to master are required.

The inquiry and inquiry response sequences always utilizes the GIAC LAP as lower address part and the DCI (Section 5.4 on page 72) as upper address part in deriving the hopping sequence, even if it concerns a DIAC inquiry.

11.1 GENERAL SELECTION SCHEME

The selection scheme consists of two parts:

- selecting a sequence;
- mapping this sequence on the hop frequencies;

The general block diagram of the hop selection scheme is shown in Figure 11.1 on page 127. The mapping from the input to a particular hop frequency is performed in the selection box. Basically, the input is the native clock and the current address. In **CONNECTION** state, the native clock (CLKN) is modified by an offset to equal the master clock (CLK). Only the 27 MSBs of the clock are used. In the **page** and **inquiry** substates, all 28 bits of the clock are used. However, in **page** substate the native clock will be modified to the master's estimate of the paged unit.

The address input consists of 28 bits, i.e., the entire LAP and the 4 LSBs of the UAP. In **CONNECTION** state, the address of the master is used. In **page** substate the address of the paged unit is used. When in **inquiry** substate, the UAP/LAP corresponding to the GIAC is used. The output constitutes a pseudo-random sequence, either covering 79 hop or 23 hops, depending on the state.

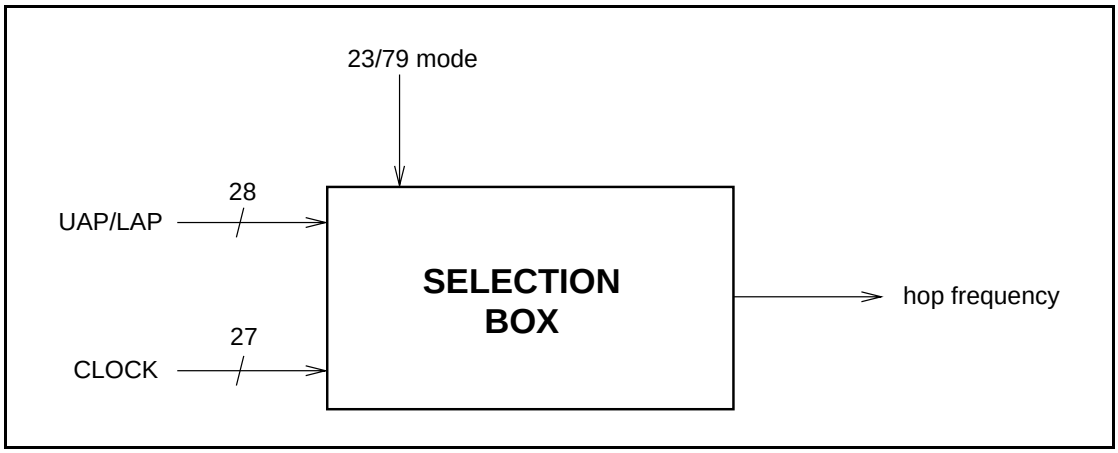


Figure 11.1: General block diagram of hop selection scheme.

For the 79-hop system, the selection scheme chooses a segment of 32 hop frequencies spanning about 64 MHz and visits these hops once in a random order. Next, a different 32-hop segment is chosen, etc. In case of the **page, page scan,** or **page response** substates, the same 32-hop segment is used all the time (the segment is selected by the address; different units will have different paging segments). In connection state, the output constitutes a pseudo-random sequence that slides through the 79 hops or 23 hops, depending on the selected hop system. For the 23-hop systems, the segment size is 16. The principle is depicted in Figure 11.2

.

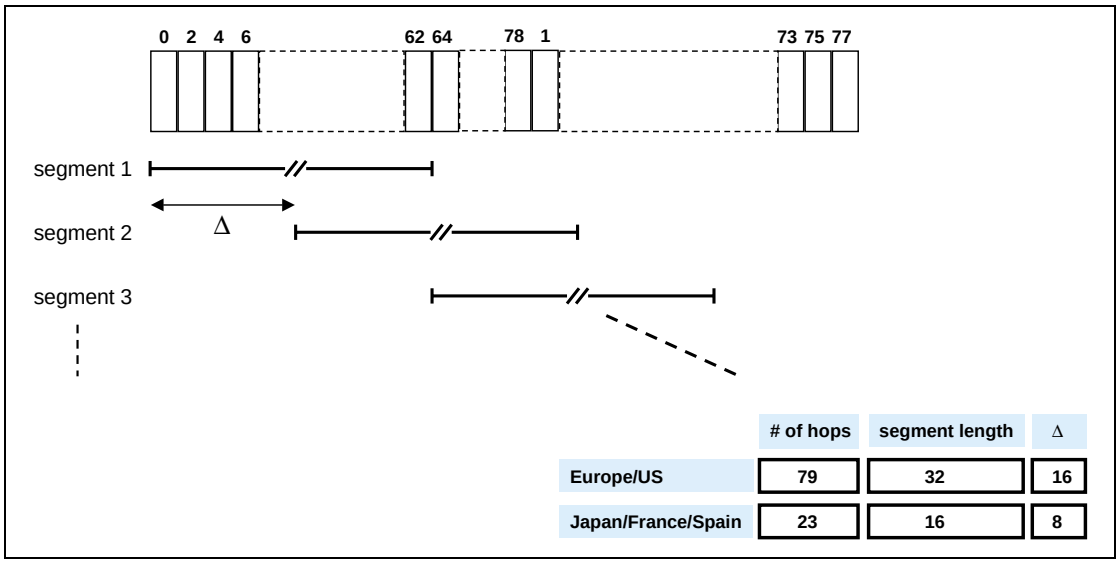


Figure 11.2: Hop selection scheme in CONNECTION state.

11.2 SELECTION KERNEL

The hop selection kernels for the 79 hop system and the 23 hop system are shown in Figure 11.3 on page 128 and Figure 11.4 on page 128, respectively. The X input determines the phase in the 32-hop segment, whereas Y1 and Y2 selects between master-to-slave and slave-to-master transmission. The inputs A to D determine the ordering within the segment, the inputs E and F determine the mapping onto the hop frequencies. The kernel addresses a register containing the hop frequencies. This list should be created such that first all even hop frequencies are listed and then all odd hop frequencies. In this way, a 32-hop segment spans about 64 MHz, whereas a 16-hop segment spans the entire 23-MHz.

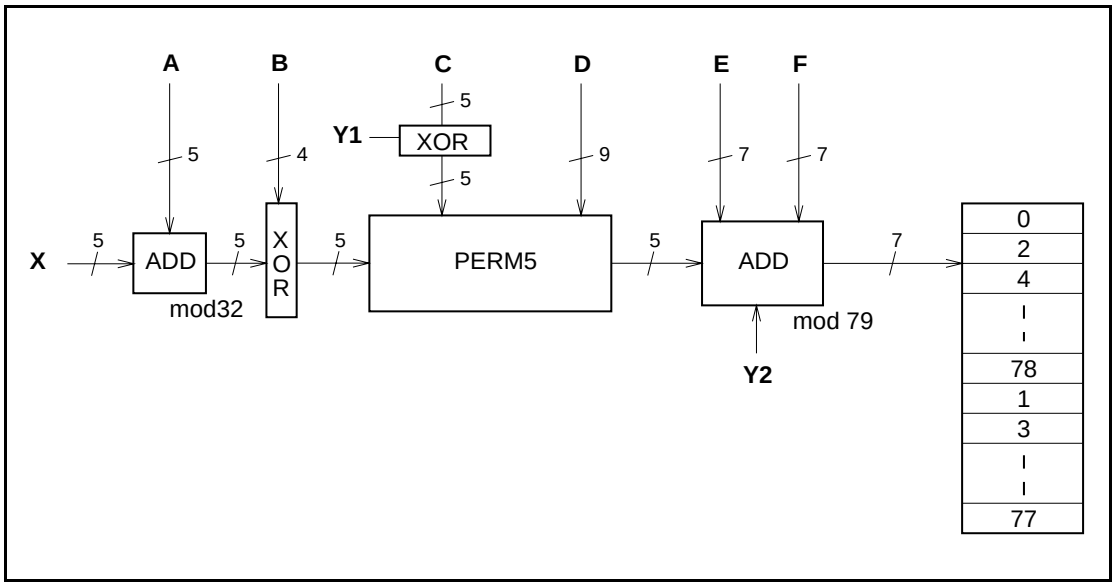


Figure 11.3: Block diagram of hop selection kernel for the 79-hop system.

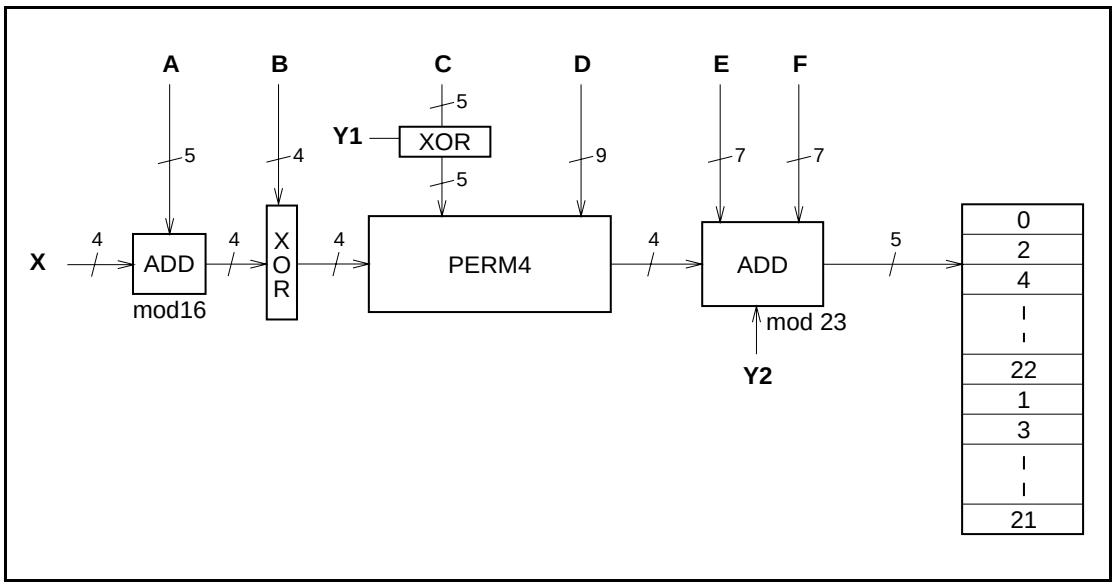


Figure 11.4: Block diagram of hop selection kernel for the 23-hop system.

The selection procedure consists of an addition, an XOR operation, a permutation operation, an addition, and finally a register selection. In the remainder of this chapter, the notation A_i is used for bit i of the BD_ADDR.

11.2.1 First addition operation

The first addition operation only adds a constant to the phase and applies a modulo 32 or a modulo 16 operation. For the page hopping sequence, the first addition is redundant since it only changes the phase within the segment. However, when different segments are concatenated (as in the channel hopping sequence), the first addition operation will have an impact on the resulting sequence.

11.2.2 XOR operation

Let Z' denote the output of the first addition. In the XOR operation, the four LSBs of Z' are modulo-2 added to the address bits $A_{22\text{-}19}$. The operation is illustrated in Figure 11.5 on page 129.

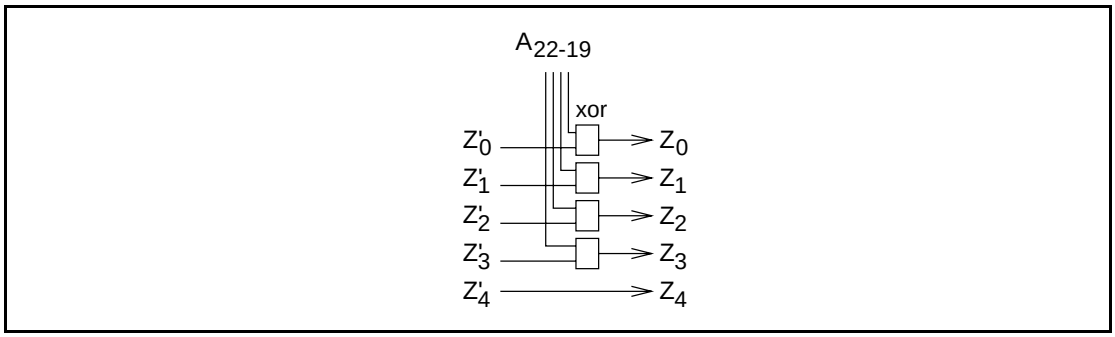


Figure 11.5: XOR operation for the 79-hop system. The 23-hop system is the same except for the Z'_4/Z_4 wire that does not exist.

11.2.3 Permutation operation

The permutation operation involves the switching from 5 inputs to 5 outputs for the 79 hop system and from 4 inputs to 4 outputs for 23 hop system, in a manner controlled by the control word. In Figure 11.6 on page 131 and Figure 11.7 on page 131 the permutation or switching box is shown. It consists of 7 stages of butterfly operations. Table 11.1 and Table 11.2 shows the control of the butterflies by the control signals P. Note that $P_{0\text{-}8}$ corresponds to $D_{0\text{-}8}$, and, P_{i+9} corresponds to $C_i \oplus Y1$ for $i = 0 \ldots 4$ in Figure 11.3 and Figure 11.4.

Control signal	Butterfly		Control signal	Butterfly
P_0	$\{Z_0,Z_1\}$		P_8	{Z1,Z4}
P_1	$\{Z_2,Z_3\}$		P_9	$\{Z_0,Z_3\}$
P_2	$\{Z_1,Z_2\}$		P_{10}	$\{Z_2,Z_4\}$
P_3	$\{Z_3,Z_4\}$		P_{11}	$\{Z_1,Z_3\}$
P_4	$\{Z_0,Z_4\}$		P_{12}	$\{Z_0,Z_3\}$
P_5	$\{Z_1,Z_3\}$		P_{13}	$\{Z_1,Z_2\}$
P_6	$\{Z_0,Z_2\}$			
P_7	$\{Z_3,Z_4\}$			

Table 11.1: Control of the butterflies for the 79 hop system

Control signal	Butterfly		Control signal	Butterfly
P_0	$\{Z_0,Z_1\}$		P_8	$\{Z_0,Z_2\}$
P_1	$\{Z_2,Z_3\}$		P_9	$\{Z_1,Z_3\}$
P_2	$\{Z_0,Z_3\}$		P_{10}	$\{Z_0,Z_3\}$
P_3	$\{Z_1,Z_2\}$		P_{11}	$\{Z_1,Z_2\}$
P_4	$\{Z_0,Z_2\}$		P_{12}	$\{Z_0,Z_1\}$
P_5	$\{Z_1,Z_3\}$		P_{13}	$\{Z_2,Z_3\}$
P_6	$\{Z_0,Z_1\}$			
P_7	$\{Z_2,Z_3\}$			

Table 11.2: Control of the butterflies for the 23 hop system

The Z input is the output of the XOR operation as described in the previous section. The butterfly operation can be implemented with multiplexers as depicted in Figure 11.8 on page 131.

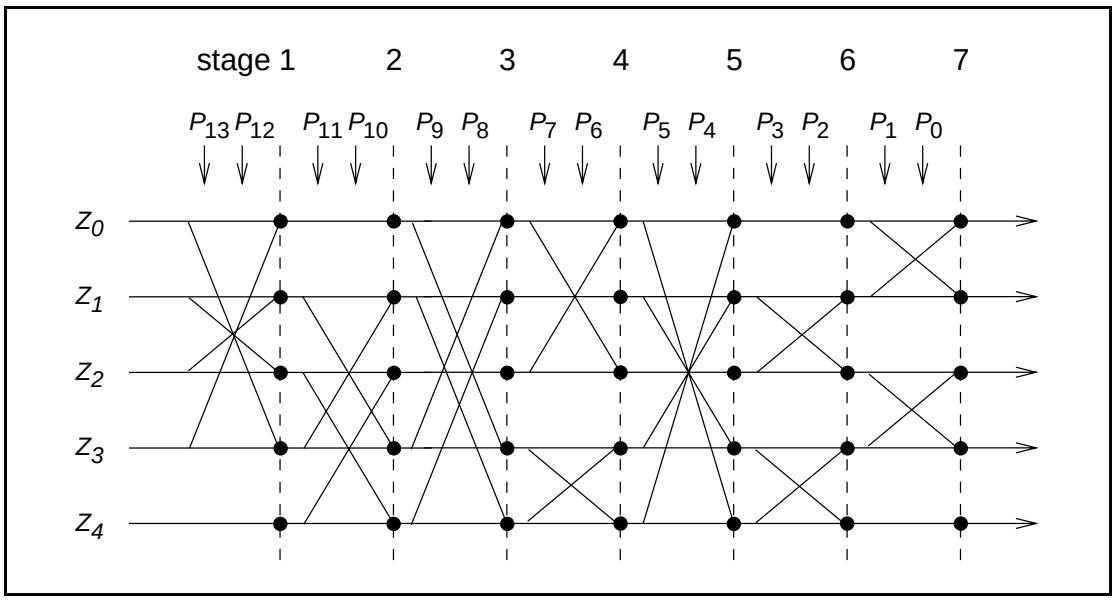


Figure 11.6: Permutation operation for the 79 hop system.

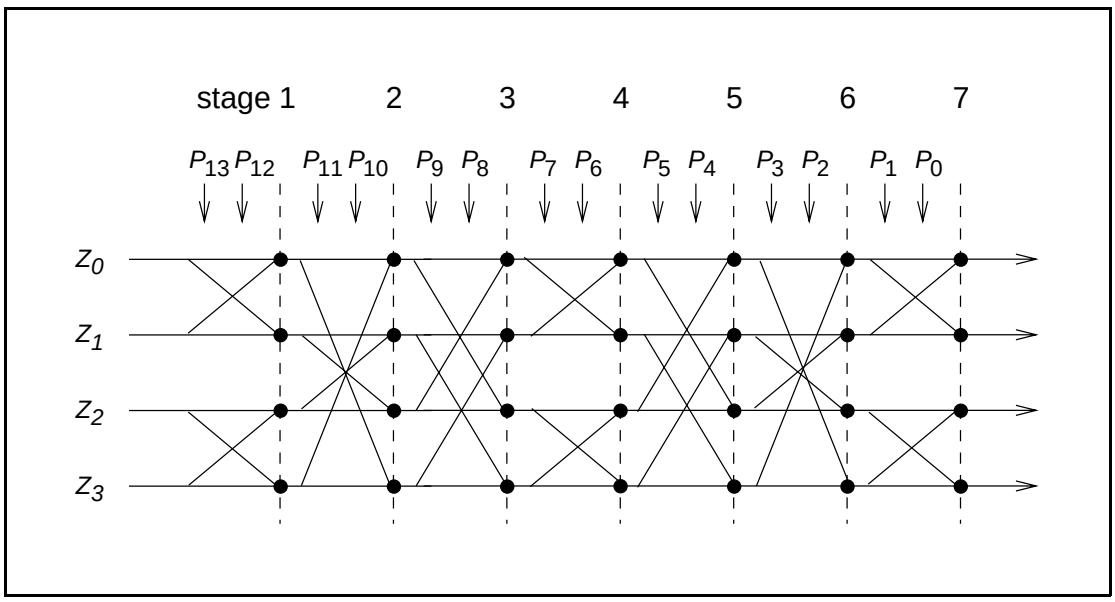


Figure 11.7: Permutation operation for the 23 hop system.

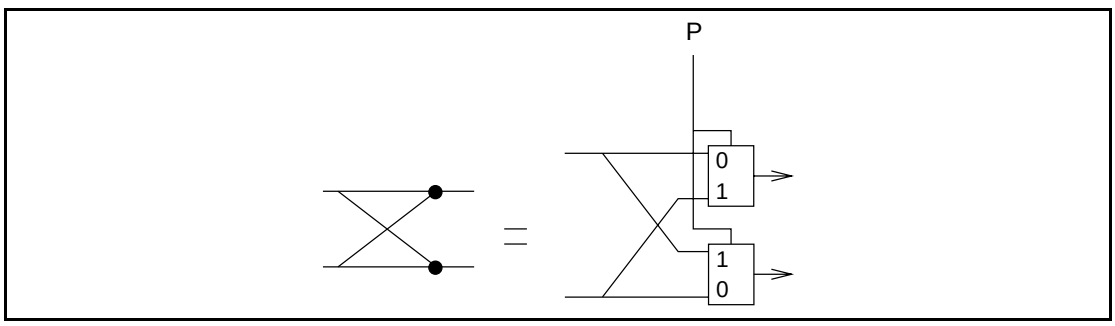


Figure 11.8: Butterfly implementation.

11.2.4 Second addition operation

The addition operation only adds a constant to the output of the permutation operation. As a result, the 16-hop or 32-hop segment is mapped differently on the hop frequencies. The addition is applied modulo 79 or modulo 23 depending on the system type (Europe/US vs. others).

11.2.5 Register bank

The output of the adder addresses a bank of 79 or 23 registers. The registers are loaded with the synthesizer code words corresponding to the hop frequencies 0 to 78 or 0 to 22. Note that the upper half of the bank contains the even hop frequencies, whereas the lower half of the bank contains the odd hop frequencies.

11.3 CONTROL WORD

In the following section X_{j-i}, $i<j$, will denote bits i, i+1,...,j of the bit vector X. By convention, X_0 is the least significant bit of the vector X.

The control word P of the kernel is controlled by the overall control signals X, Y1, Y2, and A to F as illustrated in Figure 11.3 on page 128 and Figure 11.4 on page 128. During paging and inquiry, the inputs A to E use the address values as given in the corresponding columns of Table 11.3 on page 133 and Table 11.4 on page 133. In addition, the inputs X, Y1 and Y2 are used. The F input is unused. In the 79-hop system, the clock bits CLK_{6-2} (i.e., input X) specifies the phase within the length 32 sequence, while for the 23-hop system, CLK_{5-2} specifies the phase within the length 16 sequence. For both systems, CLK_1 (i.e., inputs Y1 and Y2) is used to select between TX and RX. The address inputs determine the sequence order within segments. The final mapping onto the hop frequencies is determined by the register contents.

In the following we will distinguish between three types of clocks: the piconet's master clock, the Bluetooth unit's native clock, and the clock estimate of a paged Bluetooth unit. These types are marked in the following way:

1. CLK_{27-0}: Master clock of the current piconet.
2. $CLKN_{27-0}$: Native clock of the unit.
3. $CLKE_{27-0}$: The paging unit's estimate of the paged unit's native clock.

During the **CONNECTION** state, the inputs A, C and D result from the address bits being bit-wise XORed with the clock bits as shown in the "Connection state" column of Table 11.3 on page 133 and Table 11.4 on page 133 (the two MSBs are XORed together, the two second MSBs are XORed together, etc.). Consequently, after every 32 (16) time slots, a new length 32 (16) segment is selected in the 79-hop (23-hop) system. The sequence order within a specific

segment will not be repeated for a very long period. Thus, the overall hopping sequence consists of concatenated segments of 32-hops each. Since each 32-hop sequence spans more than 80% of the 79 MHz band, the desired frequency spreading over a short time interval is obtained.

	Page scan/ Inquiry scan	Page/Inquiry	Page response (master/slave) and Inquiry response	Connection state
X	$CLKN_{16-12}$ / $Xir^{(79)}_{4-0}$	$Xp^{(79)}_{4-0}/Xi^{(79)}_{4-0}$	$Xprm^{(79)}_{4-0}/Xprs^{(79)}_{4-0}/Xir^{(79)}_{4-0}$	CLK_{6-2}
Y1	0	$CLKE_1/CLKN_1$	$CLKE_1/CLKN_1/1$	CLK_1
Y2	0	$32 \times CLKE_1/$ $32 \times CLKN_1$	$32 \times CLKE_1$ / $32 \times CLKE_1$ 32×1	$32 \times CLK_1$
A	A_{27-23}	A_{27-23}	A_{27-23}	$A_{27-23} \oplus CLK_{25-21}$
B	A_{22-19}	A_{22-19}	A_{22-19}	A_{22-19}
C	$A_{8, 6, 4, 2, 0}$	$A_{8, 6, 4, 2, 0}$	$A_{8, 6, 4, 2, 0}$	$A_{8, 6, 4, 2, 0} \oplus CLK_{20-16}$
D	A_{18-10}	A_{18-10}	A_{18-10}	$A_{18-10} \oplus CLK_{15-7}$
E	$A_{13, 11, 9, 7, 5, 3, 1}$	$A_{13, 11, 9, 7, 5, 3, 1}$	$A_{13, 11, 9, 7, 5, 3, 1}$	$A_{13, 11, 9, 7, 5, 3, 1}$
F	0	0	0	$16 \times CLK_{27-7} \bmod 79$

Table 11.3: Control for 79-hop system.

	Page scan/ Inquiry scan	Page/Inquiry	Page response (master/slave) and Inquiry response	Connection state
X	$CLKN_{15-12}$ / $Xir^{(23)}_{3-0}$	$Xp^{(23)}_{3-0}/Xi^{(23)}_{3-0}$	$Xprm^{(23)}_{3-0}/Xprs^{(23)}_{3-0}/Xir^{(23)}_{3-0}$	CLK_{5-2}
Y1	0	$CLKE_1/CLKN_1$	$CLKE_1/CLKN_1/1$	CLK_1
Y2	0	$16 \times CLKE_1/$ $16 \times CLKN_1$	$16 \times CLKE_1$ / $16 \times CLKE_1$ 16×1	$16 \times CLK_1$
A	A_{27-23}	A_{27-23}	A_{27-23}	$A_{27-23} \oplus CLK_{25-21}$
B	A_{22-19}	A_{22-19}	A_{22-19}	A_{22-19}
C	$A_{8, 6, 4, 2, 0}$	$A_{8, 6, 4, 2, 0}$	$A_{8, 6, 4, 2, 0}$	$A_{8, 6, 4, 2, 0} \oplus CLK_{20-16}$

Table 11.4: Control for 23-hop system.

	Page scan/ Inquiry scan	Page/Inquiry	Page response (master/slave) and Inquiry response	Connection state
D	A_{18-10}	A_{18-10}	A_{18-10}	$A_{18-10} \oplus \mathrm{CLK}_{15-7}$
E	$A_{13, 11, 9, 7, 5, 3, 1}$	$A_{13, 11, 9, 7, 5, 3, 1}$	$A_{13, 11, 9, 7, 5, 3, 1}$	$A_{13, 11, 9, 7, 5, 3, 1}$
F	0	0	0	$6 \times \mathrm{CLK}_{27-6} \bmod 23$

Table 11.4: Control for 23-hop system.

11.3.1 Page scan and Inquiry scan substates

In **page scan**, the Bluetooth device address of the scanning unit is used as address input. In **inquiry scan**, the GIAC LAP and the four LSBs of the DCI (as A_{27-24}), are used as address input for the hopping sequence. Naturally, for the transmitted access code and in the receiver correlator, the appropriate GIAC or DIAC is used. The application decides which inquiry access code to use depending on the purpose of the inquiry.

The five X input bits vary depending on the current state of the unit. In the **page scan** and **inquiry scan** substates, the native clock (CLKN) is used. In **CONNECTION** state the master clock (CLK) is used as input. The situation is somewhat more complicated for the other states.

11.3.2 Page substate

In the **page** substate of the 79-hop system, the paging unit shall start using the **A**-train, i.e., $\{f(k-8), \ldots, f(k), \ldots, f(k+7)\}$, where $f(k)$ is the source's estimate of the current receiver frequency in the paged unit. Clearly, the index k is a function of all the inputs in Figure 11.3. There are 32 possible paging frequencies within each 1.28 second interval. Half of these frequencies belongs to the **A**-train, the rest (i.e., $\{f(k+8), \ldots, f(k+15), f(k-16), \ldots, f(k-9)\}$) belongs to the **B**-train. In order to achieve the -8 offset of the **A**-train, a constant of 24 can be added to the clock bits (which is equivalent to -8 due to the modulo 32 operation). Clearly, the **B**-train may be accomplished by setting the offset to 8. A cyclic shift of the order within the trains is also necessary in order to avoid a possible repetitive mismatch between the paging and scanning units. Thus,

$$Xp^{(79)} = [\mathrm{CLKE}_{16-12} + k_{offset} + (\mathrm{CLKE}_{4-2,0} - \mathrm{CLKE}_{16-12}) \bmod 16] \bmod 32, \quad \text{(EQ 2)}$$

where

$$k_{offset} = \begin{cases} 24 & \text{A-train,} \\ 8 & \text{B-train.} \end{cases} \quad \text{(EQ 3)}$$

Alternatively, each switch between the **A**- and **B**-trains may be accomplished by adding 16 to the current value of k_{offset} (originally initialized with 24).

In the **page** substate of the 23-hop system, the paging unit makes use of the **A**-train only. A constant offset of 8 is used in order to start with $f(k-8)$. Moreover, only four bits are needed since the additions are modulo 16. Consequently,

$$Xp^{(23)} = [\mathrm{CLKE}_{15-12} + 8 + \mathrm{CLKE}_{4-2,0}] \mod 16, \quad \text{(EQ 4)}$$

11.3.3 Page response

11.3.3.1 Slave response

A unit in the **page scan** substate recognizing its own access code enters the **slave response** substate. In order to eliminate the possibility of loosing the link due to discrepancies of the native clock CLKN and the master's clock estimate CLKE, the four bits CLKN_{16-12} must be frozen at their current value. The value is frozen to the content it has in the slot where the recipient's access code is detected. Note that the actual native clock is *not* stopped; it is merely the values of the bits used for creating the X-input that are kept fixed for a while. In the sequel, a frozen value is marked by an asterisk (*).

For each response slot the paged unit will use an X-input value one larger (modulo 32 or 16) than in the preceding response slot. However, the first response is made with the X-input kept at the same value as it was when the access code was recognized. Let N be a counter starting at zero. Then, the X-input in the $(N+1)$-th response slot (the first response slot being the one immediately following the page slot now responding to) of the **slave response** substate becomes

$$Xprs^{(79)} = [\mathrm{CLKN}^*_{16-12} + N] \mod 32, \quad \text{(EQ 5)}$$

and

$$Xprs^{(23)} = [\mathrm{CLKN}^*_{15-12} + N] \mod 16, \quad \text{(EQ 6)}$$

for the 79-hop and 23-hop systems, respectively. The counter N is set to zero in the slot where the slave acknowledges the page (see Figure 10.6 on page 102 and Figure 10.7 on page 102). Then, the value of N is increased by one each time CLKN_1 is set to zero, which corresponds to the start of a master TX slot. The X-input is constructed this way until the first accepted **FHS** packet is received *and* the immediately following response packet has been transmitted. After this the slave enters the **CONNECTION** state using the parameters received in the **FHS** packet.

11.3.3.2 Master response

The paging unit enters **master response** substate upon receiving a slave response. Clearly, also the master must freeze its estimated slave clock to the value that triggered a response from the paged unit. It is equivalent to using the values of the clock estimate when receiving the slave response (since only CLKE_1 will differ from the corresponding page transmission). Thus, the values are frozen when the slave **ID** packet is received. In addition to the used clock bits, also the current value of k_{offset} must be frozen. The master will adjust its X-input in the same way the paged unit does, i.e., by incrementing this value by one for each time CLKE_1 is set to zero. The first increment shall be done before sending the **FHS** packet to the paged unit. Let N be a counter starting at one. The rules for forming the X-inputs become

$$Xprm^{(79)} = [\mathrm{CLKE}^*_{16-12} + k_{offset}^* + (\mathrm{CLKE}^*_{4-2,0} - \mathrm{CLKE}^*_{16-12}) \bmod 16 + N] \bmod 32, \quad \text{(EQ 7)}$$

and

$$Xprm^{(23)} = [\mathrm{CLKE}^*_{15-12} + 8 + \mathrm{CLKE}^*_{4-2,0} + N] \bmod 16, \quad \text{(EQ 8)}$$

for the 79-hop and 23-hop systems, respectively. The value of N is increased each time CLKE_1 is set to zero, which corresponds to the start of a master TX slot.

11.3.4 Inquiry substate

The X-input of the **inquiry** substate is quite similar to what is used in the **page** substate. Since no particular unit is addressed, the native clock CLKN of the inquirer is used. Moreover, which of the two train offsets to start with is of no real concern in this state. Consequently,

$$Xi^{(79)} = [\mathrm{CLKN}_{16-12} + k_{offset} + (\mathrm{CLKN}_{4-2,0} - \mathrm{CLKN}_{16-12}) \bmod 16] \bmod 32, \quad \text{(EQ 9)}$$

where k_{offset} is defined by (EQ 3) on page 134. The initial choice of the offset is arbitrary. For the 23-hop system,

$$Xi^{(23)} = [\mathrm{CLKN}_{15-12} + 8 + \mathrm{CLKN}_{4-2,0}] \bmod 16, \quad \text{(EQ 10)}$$

The GIAC LAP and the four LSBs of the DCI (as A_{27-24}) are used as address input for the hopping sequence generator. (EQ 11)

11.3.5 Inquiry response

The **inquiry response** substate is similar to the **slave response** substate with respect to the X-input. However, there is no need to freeze the clock input, thus

$$Xir^{(79)} = [\mathrm{CLKN}_{16-12} + N] \mod 32, \quad \text{(EQ 12)}$$

and

$$Xir^{(23)} = [\mathrm{CLKN}_{15-12} + N] \mod 16, \quad \text{(EQ 13)}$$

for the 79-hop and 23-hop systems, respectively. Furthermore, the counter N is increased not on clock CLKN_1, but rather after each **FHS** packet has been transmitted in response to the inquiry. There is no restriction on the initial value of N as it is independent of the corresponding value in the inquiring unit.

The GIAC LAP and the four LSBs of the DCI (as A_{27-24}) are used as address input for the hopping sequence generator. The other input bits to the generator are the same as in the case of page response.

11.3.6 Connection state

In **CONNECTION** state, the clock bits to use in the channel hopping sequence generation are always according to the master clock, CLK. The address bits are taken from the Bluetooth device address of the master.

12 BLUETOOTH AUDIO

On the Bluetooth air-interface, either a 64 kb/s log PCM format (A-law or μ-law) is used, or a 64 kb/s CVSD (Continuous Variable Slope Delta Modulation) is used. The latter format applies an adaptive delta modulation algorithm with syllabic companding.

The voice coding on the line interface should have a quality equal to or better than the quality of 64 kb/s log PCM.

Table 12.1 on page 138 summarizes the voice coding schemes supported on the air interface. The appropriate voice coding scheme is selected after negotiations between the Link Managers.

Voice Codecs	
linear	CVSD
8-bit logarithmic	A-law
	μ-law

Table 12.1: Voice coding schemes supported on the air interface.

12.1 LOG PCM CODEC

Since the voice channels on the air-interface can support a 64 kb/s information stream, a 64 kb/s log PCM traffic can be used for transmission. Either A-law or μ-law compression can be applied. In the event that the line interface uses A-law and the air interface uses μ-law or vice versa, a conversion from A-law to μ-law is performed. The compression method follows ITU-T recommendations G. 711.

12.2 CVSD CODEC

A more robust format for voice over the air interface is a delta modulation. This modulation scheme follows the waveform where the output bits indicate whether the prediction value is smaller or larger then the input waveform.To reduce slope overload effects, syllabic companding is applied: the step size is adapted according to the average signal slope. The input to the CVSD encoder is 64 ksamples/s linear PCM. Block diagrams of the CVSD encoder and CVSD decoder are shown in Figure 12.1 on page 139, Figure 12.2 on page 139 and Figure 12.3 on page 139. The system is clocked at 64 kHz.

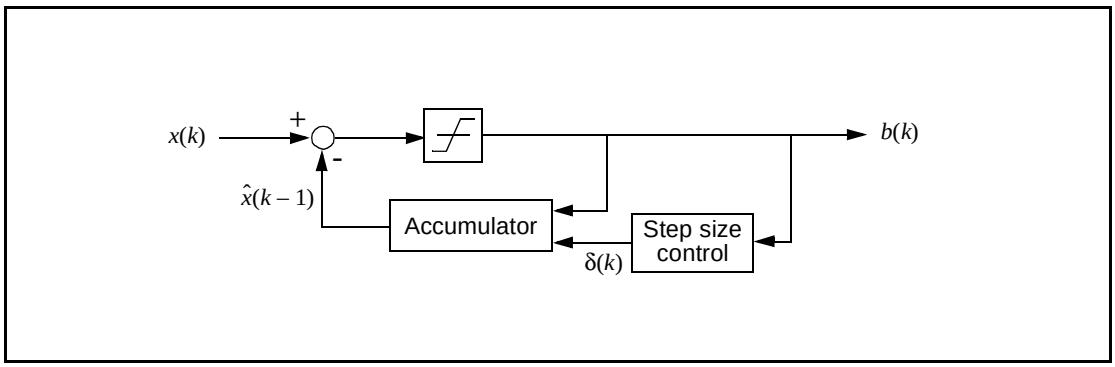


Figure 12.1: Block diagram of CVSD encoder with syllabic companding.

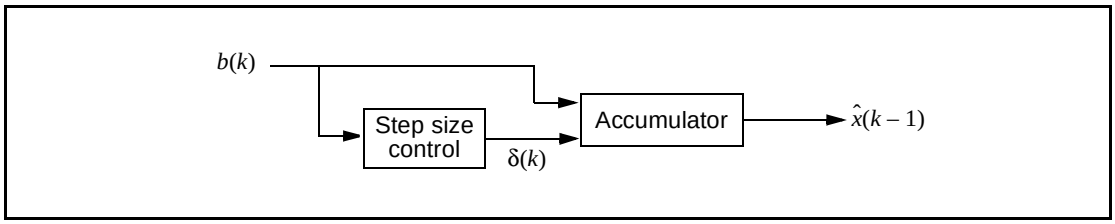


Figure 12.2: Block diagram of CVSD decoder with syllabic companding.

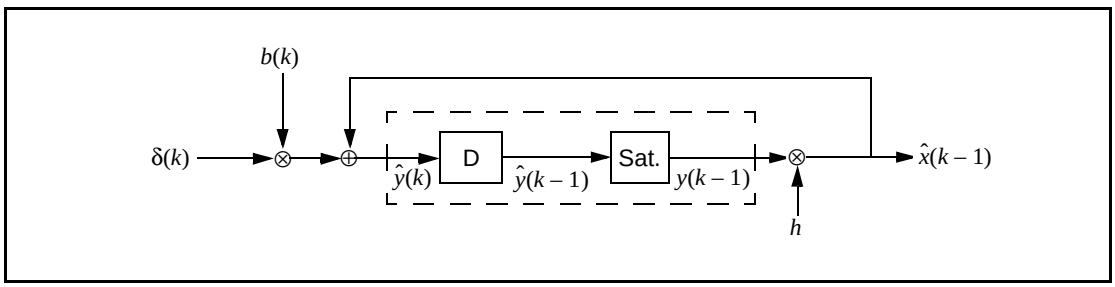


Figure 12.3: Accumulator procedure

Let $\mathrm{sgn}(x) = 1$ for $x \geq 0$, otherwise $\mathrm{sgn}(x) = -1$. On air these numbers are represented by the sign bit; i.e. negative numbers are mapped on “1” and positive numbers are mapped on “0”. Denote the CVSD encoder output bit $b(k)$, the accumulator contents $y(k)$, and the step size $\delta(k)$. Furthermore, let *h* be the decay factor for the accumulator, let β denote the decay factor for the step size, and, let α be the syllabic companding parameter. The latter parameter monitors the slope by considering the *K* most recent output bits

Let

$$\hat{x}(k) = hy(k). \qquad \text{(EQ 14)}$$

Then, the CVSD encoder internal state is updated according to the following set of equations:

$$b(k) = \mathrm{sgn}\{x(k) - \hat{x}(k-1)\}, \qquad \text{(EQ 15)}$$

$$\alpha = \begin{cases} 1, & \text{if } J \text{ bits in the last } K \text{ output bits are equal,} \\ 0, & \text{otherwise,} \end{cases} \quad \text{(EQ 16)}$$

$$\delta(k) = \begin{cases} \min\{\delta(k-1) + \delta_{min}, \delta_{max}\}, & \alpha = 1, \\ \max\{\beta\delta(k-1), \delta_{min}\}, & \alpha = 0, \end{cases} \quad \text{(EQ 17)}$$

$$y(k) = \begin{cases} \min\{\hat{y}(k), y_{max}\}, & \hat{y}(k) \geq 0. \\ \max\{\hat{y}(k), y_{min}\}, & \hat{y}(k) < 0. \end{cases} \quad \text{(EQ 18)}$$

where

$$\hat{y}(k) = \hat{x}(k-1) + b(k)\delta(k). \quad \text{(EQ 19)}$$

In these equations, δ_{min} and δ_{max} are the minimum and maximum step sizes, and, y_{min} and y_{max} are the accumulator's negative and positive saturation values, respectively. Over air, the bits are sent in the same order they are generated by the CVSD encoder.

For a 64 kb/s CVSD, the parameters as shown in Table 12.2 must be used. The numbers are based on a 16 bit signed number output from the accumulator. These values result in a time constant of 0.5 ms for the accumulator decay, and a time constant of 16 ms for the step size decay

Parameter	Value
h	$1 - \frac{1}{32}$
β	$1 - \frac{1}{1024}$
J	4
K	4
δ_{min}	10
δ_{max}	1280
y_{min}	-2^{15} or $-2^{15} + 1$
y_{max}	$2^{15} - 1$

Table 12.2: CVSD parameter values. The values are based on a 16 bit signed number output from the accumulator.

12.3 ERROR HANDLING

In the **DV** and **HV3** packet, the voice is not protected by FEC. The quality of the voice in an error-prone environment then depends on the robustness of the voice coding scheme. CVSD, in particular, is rather insensitive to random bit errors, which are experienced as white background noise. However, when a packet is rejected because either the channel access code or the HEC test was unsuccessful, measures have to be taken to “fill” in the lost speech segment.

The voice payload in the **HV2** packet is protected by a 2/3 rate FEC. For errors that are detected but cannot be corrected, the receiver should try to minimize the audible effects. For instance, from the 15-bit FEC segment with uncorrected errors, the 10-bit information part as found before the FEC decoder should be used. The **HV1** packet is protected by a 3 bit repetition FEC. For this code, the decoding scheme will always assume zero or one-bit errors. Thus, there exist no detectable but uncorrectable error events for **HV1** packets.

12.4 GENERAL AUDIO REQUIREMENTS

These specifications are tentative and will be fixed within 18 months after the release of the Bluetooth Specification version 1.0 Draft Foundation.

12.4.1 Signal levels

For A-law and μ-law log-PCM encoded signals the requirements on signal levels follows ITU-T G.711.

Full swing at the 16 bit linear PCM interface to the CVSD encoder is defined to be 3 dBm0. A digital CVSD encoded test signal is provided in a Test Signal file available on the website. This signal is generated by a software implementation of a reference CVSD encoder. The digital encoder input signal (1020 Hz, sine-wave) generating the test signal has a nominal power of -15 dBm0. When the CVSD encoded test signal is fed through the CVSD receiver chain, the nominal output power should be -15 ± 1.0 dBm0.

12.4.2 CVSD audio quality

For Bluetooth audio quality the requirements are put on the transmitter side. The 64 ksamples/s linear PCM input signal must have negligible spectral power density above 4 kHz. A set of reference input signals are encoded by the transmitter and sent through a reference decoder (available on the website). The power spectral density in the 4-32 kHz band of the decoded signal at the 64 ksample/s linear PCM output, should be more than 20 dB below the maximum in the 0-4 kHz range.

13 BLUETOOTH ADDRESSING

13.1 BLUETOOTH DEVICE ADDRESS (BD_ADDR)

Each Bluetooth transceiver is allocated a unique 48-bit Bluetooth device address (BD_ADDR). This address is derived from the IEEE802 standard. This 48-bit address is divided into three fields:

- LAP field: lower address part consisting of 24 bits
- UAP field: upper address part consisting of 8 bits
- NAP field: non-significant address part consisting of 16 bits

The LAP and UAP form the significant part of the BD_ADDR. The total address space obtained is 2^{32}.

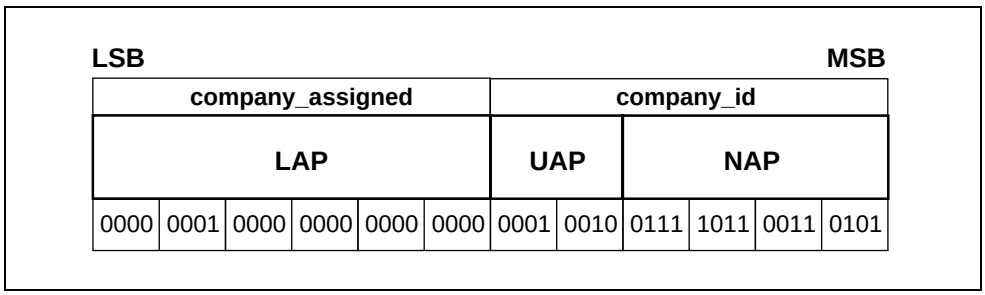


Figure 13.1: Format of BD_ADDR

13.2 ACCESS CODES

In the Bluetooth system, 72-bit and 68-bit access codes are used for signalling purposes. Three different access codes are defined, see also Section 4.2.1 on page 48:

- device access code (DAC)
- channel access code (CAC)
- inquiry access code (IAC)

There is one general IAC (GIAC) for general inquiry operations and there are 63 dedicated IACs (DIACs) for dedicated inquiry operations. All codes are derived from a LAP of the BD_ADDR. The device access code is used during page, page scan and page response substates. It is a code derived from the unit's BD_ADDR. The channel access code characterizes the channel of the piconet and forms the preamble of all packets exchanged on the channel. The channel access code is derived from the LAP of the master BD_ADDR. Finally, the inquiry access code is used in inquiry operations. A general inquiry access code is common to all Bluetooth units; a set of dedicated inquiry access codes is used to inquire for classes of devices.

The access code is also used to indicate to the receiver the arrival of a packet. It is used for timing synchronization and offset compensation. The receiver correlates against the entire sync word in the access code, providing a very robust signalling. During channel setup, the code itself is used as an ID packet to support the acquisition process. In addition, it is used during random access procedures in the PARK state.

The access code consists of preamble, sync word and a trailer, see Section 4.2 on page 48. The next two sections describe the generation of the sync word.

13.2.1 Synchronization word definition

The sync words are based on a (64,30) expurgated block code with an overlay (bit-wise XOR) of an 64 bit full length PN-sequence. The expurgated code guarantees large Hamming distance ($d_{min} = 14$) between sync words based on different addresses. The PN sequence improves the auto correlation properties of the access code. The following steps describe how to generate the sync word:

1. Generate information sequence;
2. XOR this with the “information covering” part of the PN overlay sequence;
3. Generate the codeword;
4. XOR the codeword with all 64 bits of the PN overlay sequence;

The information sequence is generated by appending 6 bits to the 24 bit LAP (step 1). The appended bits are 001101 if the MSB of the LAP equals 0. If the MSB of the LAP is 1 the appended bits are 110010. The LAP MSB together with the appended bits constitute a length-seven Barker sequence. The purpose of including a Barker sequence is to further improve the auto correlation properties. In step 2 the information is pre-scrambled by XORing it with the bits $p_{34} \ldots p_{63}$ of the *pseudo-random noise* (PN) sequence (defined in section 13.2.2 on page 145). After generating the codeword (step 3), the complete PN sequence is XORed to the codeword (step 4). This step de-scrambles the information part of the codeword. At the same time the parity bits of the codeword are scrambled. Consequently, the original LAP and Barker sequence are ensured a role as a part of the access code sync word, and the cyclic properties of the underlying code is removed. The principle is depicted in Figure 13.2 on page 144

In the sequel, binary sequences will be denoted by their corresponding D-transform (in which D^i represents a delay of *i* time units). Let $p'(D) = p'_0 + p'_1 D + \ldots + p'_{62} D^{62}$ be the 63 bit pseudo-random sequence, where p'_0 is the first bit (LSB) leaving the PRNG (see Figure 13.3 on page 146), and, p'_{62} is the last bit (MSB). To obtain 64 bits, an extra zero is appended at the *end* of this sequence (thus, $p'(D)$ is unchanged). For notational convenience, the

reciprocal of this extended polynomial, $p(D) = D^{63}p'(1/D)$, will be used in the sequel. This is the sequence $p'(D)$ in reverse order. We denote the 24 bit lower address part (LAP) of the Bluetooth address by $a(D) = a_0 + a_1D + \ldots + a_{23}D^{23}$ (a_0 is the LSB of the Bluetooth address).

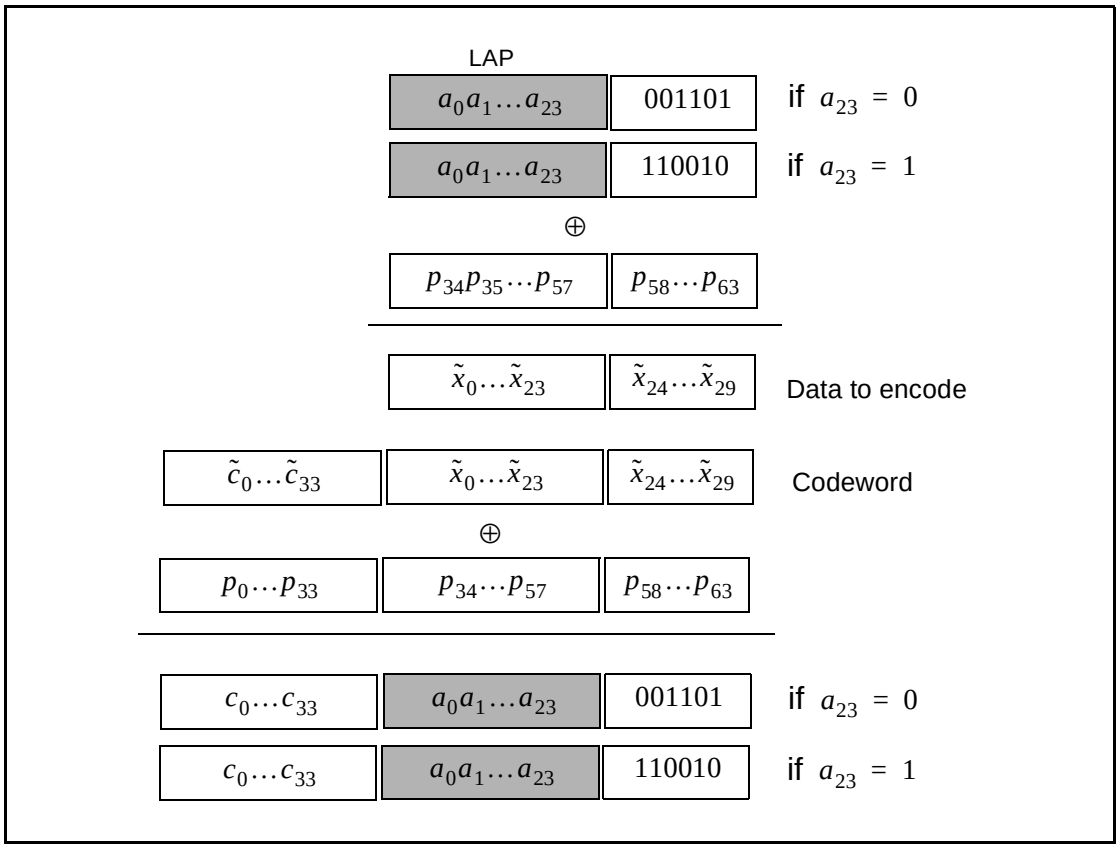


Figure 13.2: Construction of the sync word.

The (64,30) block code generator polynomial is denoted $g(D) = (1 + D)g'(D)$, where $g'(D)$ is the generator polynomial 157464165547 (octal notation) of a primitive binary (63,30) BCH code. Thus, in octal notation we have

$$g(D) = 260534236651, \quad \text{(EQ 20)}$$

the left-most bit corresponds to the high-order (g_{34}) coefficient.The DC-free four bit sequences 0101 and 1010 can be written

$$\begin{cases} F_0(D) = D + D^3, \\ F_1(D) = 1 + D^2, \end{cases} \quad \text{(EQ 21)}$$

respectively. Furthermore, we define

$$\begin{cases} B_0(D) = D^2 + D^3 + D^5, \\ B_1(D) = 1 + D + D^4, \end{cases} \quad \text{(EQ 22)}$$

which are used to create the length seven Barker sequences. Then, the access code is generated by the following procedure:

1. Format the 30 information bits to encode:

$$x(D) = a(D) + D^{24}B_{a_{23}}(D).$$

2. Add the information covering part of the PN overlay sequence:

$$\tilde{x}(D) = x(D) + p_{34} + p_{35}D + \ldots + p_{63}D^{29}.$$

3. Generate parity bits of the (64,30) expurgated block code:[1]

$$\tilde{c}(D) = D^{34}\tilde{x}(D) \mod g(D).$$

4. Create the codeword:

$$\tilde{s}(D) = D^{34}\tilde{x}(D) + \tilde{c}(D).$$

5. Add the PN sequence:

$$s(D) = \tilde{s}(D) + p(D).$$

6. Append the (DC-free) preamble and trailer:

$$y(D) = F_{c_0}(D) + D^4 s(D) + D^{68}F_{a_{23}}(D).$$

13.2.2 Pseudo-random noise sequence generation

To generate the pseudo-random noise sequence we use the primitive polynomial $h(D) = 1 + D + D^3 + D^4 + D^6$. The LFSR and its starting state are shown in Figure 13.3 on page 146. The PN sequence generated (including the extra terminating zero) becomes (hexadecimal notation) 83848D96BBCC54FC. The LFSR output starts with the left-most bit of this PN sequence. This corresponds to $p'(D)$ of the previous section. Thus, using the reciprocal $p(D)$ as overlay gives the 64 bit sequence

$$\boldsymbol{p} = 3F2A33DD69B121C1, \quad \text{(EQ 23)}$$

1. $x(D)$ mod $y(D)$ denotes the rest when $x(D)$ is divided by $y(D)$.

where the left-most bit is $p_0 = 0$ (there are two initial zeros in the binary representation of the hexadecimal digit 3), and $p_{63} = 1$ is the right-most bit.

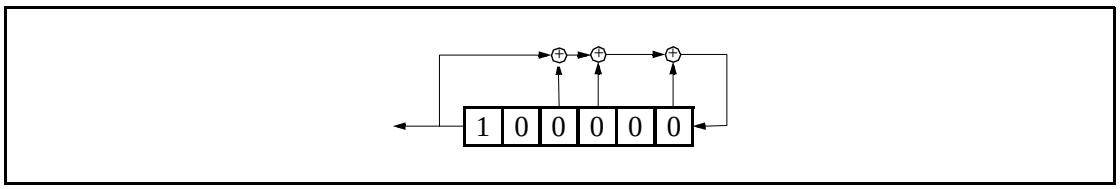


Figure 13.3: LFSR and the starting state to generate $p'(D)$

13.2.3 Reserved addresses for GIAC and DIAC

There is a block of 64 contiguous LAPs reserved for Bluetooth inquiry operations; one LAP common to all Bluetooth devices is reserved for general inquiry, the remaining 63 LAPs are reserved for dedicated inquiry of specific classes of Bluetooth devices. The same 64-block is used regardless of the contents of UAP and NAP. Consequently, none of these LAPs can be part of a user BD_ADDR.

When initializing HEC and CRC for the FHS packet of **inquiry response**, the UAP is replaced by DCI. Likewise, whenever one of the reserved BD_ADDRs is used for generating a frequency hop sequence, the UAP will be replaced by the DCI.

The reserved LAP addresses are tentatively chosen as `0x9E8B00-0x9E8B3F`. The general inquiry LAP is tentatively chosen to `0x9E8B33`. All addresses have the LSB at the rightmost position, hexadecimal notation.

13.3 ACTIVE MEMBER ADDRESS (AM_ADDR)

Each slave active in a piconet is assigned a 3-bit active member address (AM_ADDR). The all-zero AM_ADDR is reserved for broadcast messages. The master does not have an AM_ADDR. Its timing relative to the slaves distinguishes it from the slaves. A slave only accepts a packet with a matching AM_ADDR and broadcast packets. The AM_ADDR is carried in the packet header. The AM_ADDR is only valid as long as a slave is active on the channel. As soon as it is disconnected or parked, it loses the AM_ADDR.

The AM_ADDR is assigned by the master to the slave when the slave is activated. This is either at connection establishment or when the slave is unparked. At connection establishment, the AM_ADDR is carried in the **FHS** payload (the **FHS** header itself carries the all-zero AM_ADDR). When unparking, the AM_ADDR is carried in the unpark message.

13.4 PARKED MEMBER ADDRESS (PM_ADDR)

A slave in park mode can be identified by its BD_ADDR or by a dedicated parked member address (PM_ADDR). This latter address is a 8-bit member address that separates the parked slaves. The PM_ADDR is only valid as long as the slave is parked. When the slave is activated it is assigned an AM_ADDR but loses the PM_ADDR. The PM_ADDR is assigned to the slave the moment it is parked.

The all-zero PM_ADDR is reserved for parked slaves that only use their BD_ADDR to be unparked.

13.5 ACCESS REQUEST ADDRESS (AR_ADDR)

The access request address is used by the parked slave to determine the slave-to-master half slot in the access window it is allowed to send access request messages in, see also Section 10.8.4.6 on page 118. The AR_ADDR is assigned to the slave when it enters the park mode and is only valid as long as the slave is parked. The AR_ADDR is not necessarily unique; i.e. different parked slaves may have the same AR_ADDR.

14 BLUETOOTH SECURITY

The Bluetooth technology provides peer-to-peer communications over short distances. In order to provide usage protection and information confidentiality, the system has to provide security measures both at the application layer and the link layer. These measures shall be appropriate for a peer environment. This means that in each Bluetooth unit, the authentication and encryption routines are implemented in the same way. Four different entities are used for maintaining security at the link layer: a public address which is unique for each user[1], two secret keys, and a random number which is different for each new transaction. The four entities and their sizes as used in Bluetooth are summarized in Table 14.1.

Entity	Size
BD_ADDR	48 bits
Private user key, authentication	128 bits
Private user key, encryption configurable length (byte-wise)	8-128 bits
RAND	128 bits

Table 14.1: Entities used in authentication and encryption procedures.

The Bluetooth device address (BD_ADDR) is the 48-bit IEEE address which is unique for each Bluetooth unit. The Bluetooth addresses are publicly known, and can be obtained via MMI interactions, or, automatically, via an inquiry routine by a Bluetooth unit.

The secret keys are derived during initialization and are further never disclosed. Normally, the encryption key is derived from the authentication key during the authentication process. For the authentication algorithm, the size of the key used is always 128 bits. For the encryption algorithm, the key size may vary between 1 and 16 octets (8 - 128 bits). The size of the encryption key shall be configurable for two reasons. The first has to do with the many different requirements imposed on cryptographic algorithms in different countries – both w.r.t. export regulations and official attitudes towards privacy in general. The second reason is to facilitate a future upgrade path for the security without the need of a costly redesign of the algorithms and encryption hardware; increasing the effective key size is the simplest way to combat increased computing power at the opponent side. Currently (1999) it seems that an encryption key size of 64 bits gives satisfying protection for most applications.

The encryption key is entirely different from the authentication key (even though the latter is used when creating the former, as is described in Section 14.5.4 on page 176). Each time encryption is activated, a new encryption key

1. The BD_ADDR is not a secured identity.

shall be generated. Thus, the lifetime of the encryption key does not necessarily correspond to the lifetime of the authentication key.

It is anticipated that the authentication key will be more static in its nature than the encryption key – once established, the particular application running on the Bluetooth device decides when, or if, to change it. To underline the fundamental importance of the authentication key to a specific Bluetooth link, it will often be referred to as the link key.

The RAND is a random number which can be derived from a random or pseudo-random process in the Bluetooth unit. This is not a static parameter, it will change frequently.

In the remainder of this chapter, the terms user and application will be used interchangeably to designate the entity that is at the originating or receiving side.

14.1 RANDOM NUMBER GENERATION

Each Bluetooth unit has a random number generator. Random numbers are used for many purposes within the security functions – for instance, for the challenge-response scheme, for generating authentication and encryption keys, etc. Ideally, a true random generator based on some physical process with inherent randomness is used. Examples of such processes are thermal noise from a semiconductor or resistor and the frequency instability of a free running oscillator. For practical reasons, a software based solution with a pseudo-random generator is probably preferable. In general, it is quite difficult to classify the randomness of a pseudo-random sequence. Within Bluetooth, the requirements placed on the random numbers used are that they be non-repeating and randomly generated.

The expression ‘non-repeating’ means that it shall be highly unlikely that the value should repeat itself within the lifetime of the authentication key. For example, a non-repeating value could be the output of a counter that is unlikely to repeat during the lifetime of the authentication key, or a date/time stamp.

The expression ‘randomly generated’ means that it shall not be possible to predict its value with a chance that is significantly larger than 0 (e.g., greater than $1/2^L$ for a key length of L bits).

Clearly, the LM can use such a generator for various purposes; i.e. whenever a random number is needed (such as the RANDs, the unit keys, K_{init}, K_{master}, and random back-off or waiting intervals).

14.2 KEY MANAGEMENT

It is important that the encryption key size within a specific unit cannot be set by the user – this must be a factory preset entity. In order to prevent the user from over-riding the permitted key size, the Bluetooth baseband processing does not accept an encryption key given from higher software layers. Whenever a new encryption key is required, it must be created as defined in Section 14.5.4 on page 176.

Changing a link key should also be done through the defined baseband procedures. Depending on what kind of link key it is, different approaches are required. The details are found in Section 14.2.2.7 on page 156.

14.2.1 Key types

The link key is a 128-bit random number which is shared between two or more parties and is the base for all security transactions between these parties. The link key itself is used in the authentication routine. Moreover, the link key is used as one of the parameters when the encryption key is derived.

In the following, a session is defined as the time interval for which the unit is a member of a particular piconet. Thus, the session terminates when the unit disconnects from the piconet.

The link keys are either semi-permanent or temporary. A semi-permanent link key is stored in non-volatile memory and may be used after the current session is terminated. Consequently, once a semi-permanent link key is defined, it may be used in the authentication of several subsequent connections between the Bluetooth units sharing it. The designation semi-permanent is justified by the possibility to change it. How to do this is described in Section 14.2.2.7 on page 156.

The lifetime of a temporary link key is limited by the lifetime of the current session – it cannot be reused in a later session. Typically, in a point-to-multipoint configuration where the same information is to be distributed securely to several recipients, a common encryption key is useful. To achieve this, a special link key (denoted master key) can temporarily replace the current link keys. The details of this procedure are found in Section 14.2.2.6 on page 156.

In the sequel we sometimes refer to what is denoted as the current link key. This is simply the link key in use at the current moment. It can be semi-permanent or temporary. Thus, the current link key is used for all authentications and all generation of encryption keys in the on-going connection (session).

In order to accommodate for different types of applications, four types of link keys have been defined:

- the combination key K_{AB}
- the unit key K_A
- the temporary key K_{master}
- the initialization key K_{init}

In addition to these keys there is an encryption key, denoted K_c. This key is derived from the current link key. Whenever the encryption is activated by a LM command, the encryption key shall be changed automatically. The purpose of separating the authentication key and encryption key is to facilitate the use of a shorter encryption key without weakening the strength of the authentication procedure. There are no governmental restrictions on the strength of authentication algorithms. However, in some countries, such restrictions exist on the strength of encryption algorithms.

For a Bluetooth unit, the combination key K_{AB} and the unit key K_A are functionally indistinguishable; the difference is in the way they are generated. The unit key K_A is generated in, and therefore dependent on, a single unit A. The unit key is generated once at installation of the Bluetooth unit; thereafter, it is very rarely changed. The combination key is derived from information in both units A and B, and is therefore always dependent on two units. The combination key is derived for each new combination of two Bluetooth units.

It depends on the application or the device whether a unit key or a combination key is used. Bluetooth units which have little memory to store keys, or, when installed in equipment that must be accessible to a large group of users, will preferably use their own unit key. In that case, they only have to store a single key. Applications that require a higher security level preferably use the combination keys. These applications will require more memory since a combination key for each link to a different Bluetooth unit has to be stored.

The master key, K_{master}, is a link key only used during the current session. It will replace the original link key only temporarily. For example, this may be utilized when a master wants to reach more than two Bluetooth units simultaneously using the same encryption key, see Section 14.2.2.6 on page 156.

The initialization key, K_{init}, is used as link key during the initialization process when no combination or unit keys have been defined and exchanged yet or when a link key has been lost. The initialization key protects the transfer of initialization parameters. The key is derived from a random number, an L-octet PIN code, and a BD_ADDR. This key is only to be used during initialization.

The PIN can be a fixed number provided with the Bluetooth unit (for example when there is no MMI as in a PSTN plug). Alternatively, the PIN can be selected arbitrarily by the user, and then entered in both units that have to be

matched. The latter procedure is used when both units have an MMI, for example a phone and a laptop. Entering a PIN in both units is more secure than using a fixed PIN in one of the units, and should be used whenever possible. Even if a fixed PIN is used, it shall be possible to change the PIN; this in order to prevent re-initialization by users who once got hold of the PIN. If no PIN is available, a default value of zero is to be used.

For many applications the PIN code will be a relatively short string of numbers. Typically, it may consist of only four decimal digits. Even though this gives sufficient security in many cases, there exist countless other, more sensitive, situations where this is not reliable enough. Therefore, the PIN code can be chosen to be any length from 1 to 16 octets. For the longer lengths, we envision the units exchanging PIN codes not through mechanical (i.e. human) interaction, but rather through means supported by software at the application layer. For example, this can be a Diffie-Hellman key agreement, where the exchanged key is passed on to the K_{init} generation process in both units, just as in the case of a shorter PIN code.

14.2.2 Key generation and initialization

The link keys have to be generated and distributed among the Bluetooth units in order to be used in the authentication procedure. Since the link keys must be secret, they cannot be obtained through an inquiry routine in the same way as the Bluetooth addresses. The exchange of the keys takes place during an initialization phase which has to be carried out separately for each two units that want to implement authentication and encryption. All initialization procedures consist of the following five parts:

- generation of an initialization key
- generation of link key
- link key exchange
- authentication
- generating of encryption key in each unit (optional)

After the initialization procedure, the units can proceed to communicate, or the link can be disconnected. If encryption is implemented, the E_0 algorithm is used with the proper encryption key derived from the current link key. For any new connection established between units A and B, they will use the common link key for authentication, instead of once more deriving K_{init} from the PIN. A new encryption key derived from that particular link key will be created next time encryption is activated.

If no link key is available, the LM shall automatically start aninitialization procedure.

14.2.2.1 Generation of initialization key, K_{init}

A link key used temporarily during initialization is derived – the initialization key K_{init}. This key is derived by the E_{22} algorithm from a BD_ADDR, a PIN code, the length of the PIN (in octets), and a random number IN_RAND. The principle is depicted in Figure 14.15 on page 176. The 128-bit output from E_{22} will be used for key exchange during the generation of a linkkey. When the units have performed the link key exchange, the initialization key shall be discarded.

When the initialization key is generated, the PIN is augmented with the BD_ADDR. If one unit has a fixed PIN the BD_ADDR of the other unit is used. If both units have a variable PIN the BD_ADDR of the device that received IN_RAND is used. If both units have a fixed PIN they cannot be paired. Since the maximum length of the PIN used in the algorithm cannot exceed 16 octets, it is possible that not all octets of BD_ADDR will be used. This procedure ensures that K_{init} depends on the identity of the unit with a variable PIN. A fraudulent Bluetooth unit may try to test a large number of PINs by each time claiming another BD_ADDR. It is the application's responsibility to take countermeasures against this threat. If the device address is kept fixed, the waiting interval until next try is permitted is increased exponentially (see Section 14.4.1 on page 169).

The details of the E_{22} algorithm can be found in Section 14.5.3 on page 174.

14.2.2.2 Authentication

The authentication procedure is carried out as described in Section 14.4 on page 168. Note that during each authentication, a new AU_RAND_A is issued.

Mutual authentication is achieved by first performing the authentication procedure in one direction andimmediately followed by performing the authentication procedure in the opposite direction.

As a side effect of a successful authentication procedure an auxiliary parameter, the Authenticated Ciphering Offset (ACO), will be computed. The ACO is used for ciphering key generation as described in Section 14.2.2.5 on page 155.

The claimant/verifier status is determined by the LM.

14.2.2.3 Generation of a unit key

A unit key K_A is generated when the Bluetooth unit is for the first time in operation; i.e. not during each initialization! The unit key is generated by the E_{21} algorithm as described in Section 14.5.3 on page 174. Once created, the unit key is stored in non-volatile memory and (almost) never changed. If after initial-

ization the unit key is changed, the previously initialized units will possess a wrong link key. At initialization, the application has to determine which of the two parties will provide the unit key as link key. Typically, this will be the unit with restricted memory capabilities, since this unit only has to remember its own unit key. The unit key is transferred to the other party and then stored as link key for that particular party. So, for example in Figure 14.1 on page 154, the unit key of unit A, K_A , is being used as link key for the connection A-B; unit A sends the unit key K_A to unit B; unit B will store K_A as the link key K_{BA}. For another initialization, for example with unit C, unit A will reuse its unit key K_A, whereas unit C stores it as K_{CA}

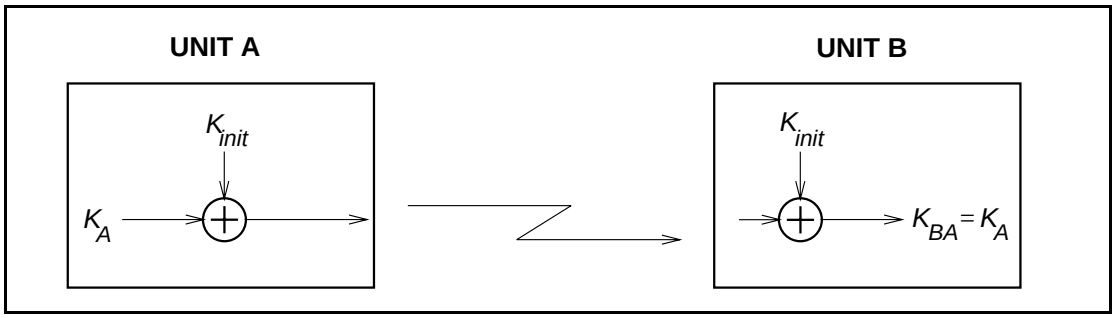


Figure 14.1: Generation of unit key. When the unit key has been exchanged, the initialization key shall be discarded in both units.

14.2.2.4 Generation of a combination key

If it is desired to use a combination key, this key is first generated during the initialization procedure. The combination key is the combination of two numbers generated in unit A and B, respectively. First, each unit generates a random number, say LK_RAND_A and LK_RAND_B. Then, utilizing E_{21} with the random number and the own BD_ADDR, the two random numbers

$$LK\_K_A = E_{21}(LK\_RAND_A, BD\_ADDR_A), \quad \text{(EQ 24)}$$

and

$$LK\_K_B = E_{21}(LK\_RAND_B, BD\_ADDR_B), \quad \text{(EQ 25)}$$

are created in unit A and unit B, respectively. These numbers constitute the units' contribution to the combination key that is to be created. Then, the two random numbers LK_RAND_A and LK_RAND_B are exchanged securely by XORing with the current link key, say K. Thus, unit A sends $K \oplus LK\_RAND_A$ to unit B, while unit B sends $K \oplus LK\_RAND_B$ to unit A. Clearly, if this is done during the initialization phase the link key $K = K_{init}$

When the random numbers LK_RAND_A and LK_RAND_B have been mutually exchanged, each unit recalculates the other units contribution to the combina-

tion key. This is possible since each unit knows the Bluetooth device address of the other unit. Thus, unit A calculates (EQ 25) on page 154 and unit B calculates (EQ 24) on page 154. After this, both units combine the two numbers to generate the 128-bit link key. The combining operation is a simple bitwise modulo-2 addition (i.e. XOR). The result is stored in unit A as the link key K_{AB} and in unit B as the link key K_{BA}. When both units have derived the new combination key, a mutual authentication procedure shall be initiated to confirm the success of the transaction. The old link key shall be discarded after a successful exchange of a new combination key. The message flow between master and slave and the principle for creating the combination key is depicted in Figure 14.2 on page 155.

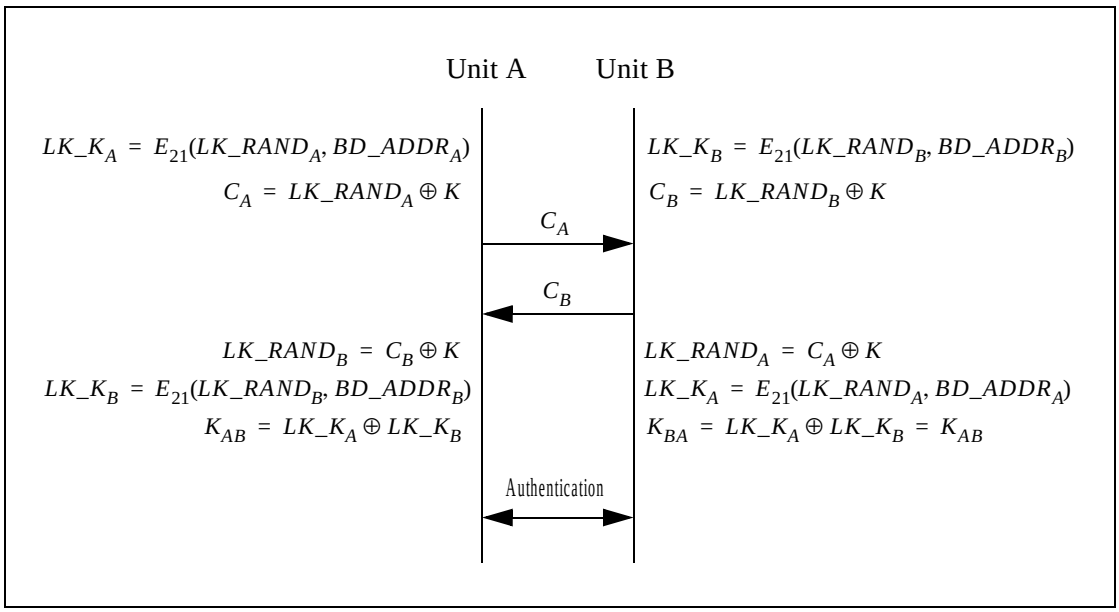


Figure 14.2: Generating a combination key. The old link key (K) shall be discarded after the exchange of a new combination key has succeeded

14.2.2.5 Generating the encryption key

The encryption key, K_C, is derived by algorithm E_3 from the current link key, a 96-bit Ciphering OFfset number (COF), and a 128-bit random number. The COF is determined in one of two ways. If the current link key is a master key, then COF is derived from the master BD_ADDR. Otherwise the value of COF is set to the value of ACO as computed during the authentication procedure. More precisely, we have[1]

$$\text{COF} = \begin{cases} \text{BD\_ADDR} \cup \text{BD\_ADDR}, & \text{if link key is a master key} \\ \text{ACO}, & \text{otherwise.} \end{cases} \quad \text{(EQ 26)}$$

1. $x \cup y$ denotes the concatenation of the octet strings x and y.

There is an explicit call of E_3 when the LM activates encryption. Consequently, the encryption key is automatically changed each time the unit enters the encryption mode. The details of the key generating function E_3 can be found in Section 14.5.4 on page 176.

14.2.2.6 Point-to-multipoint configuration

It is quite possible for the master to use separate encryption keys for each slave in a point-to-multipoint configuration with ciphering activated. Then, if the application requires more than one slave to listen to the same payload, each slave must be addressed individually. This may cause unwanted capacity loss for the piconet. Moreover, a Bluetooth unit (slave) is not capable of switching between two or more encryption keys in real time (e.g., after looking at the AM_ADDR in the header). Thus, the master cannot use different encryption keys for broadcast messages and individually addressed traffic. Alternatively, the master may tell several slave units to use a common link key (and, hence, indirectly also to use a common encryption key) and broadcast the information encrypted. For many applications, this key is only of temporary interest. In the sequel, this key is denoted K_{master}.

The transfer of necessary parameters is protected by the routine described in Section 14.2.2.8 on page 157. After the confirmation of successful reception in each slave, the master shall issue a command to the slaves to replace their respective current link key by the new (temporary) master key. Before encryption can be activated, the master also has to generate and distribute a common EN_RAND to all participating slaves. Using this random number and the newly derived master key, each slave generates a new encryption key.

Note that the master must negotiate what encryption key length to use individually with each slave who wants to use the master key. In case the master already has negotiated with some of these slaves, it has knowledge of what sizes can be accepted. Clearly, there might be situations where the permitted key lengths of some units are incompatible. In that case, the master must have the limiting unit excluded from the group.

When all slaves have received the necessary data, the master can communicate information on the piconet securely using the encryption key derived from the new temporary link key. Clearly, each slave in possession of the master key can eavesdrop on all encrypted traffic, not only the traffic intended for itself. If so desired, the master can tell all participants to fall back to their old link keys simultaneously.

14.2.2.7 Modifying the link keys

In certain circumstances, it is desirable to be able to modify the link keys. A link key based on a unit key can be changed, but not very easily. The unit key is created once during the first use. Changing the unit key is a less desirable

alternative, as several units may share the same unit key as link key (think of a printer whose unit key is distributed among all users using the printer's unit key as link key). Changing the unit key will require re-initialization of all units trying to connect. In certain cases, this might be desirable; for example to deny access to previously allowed units.

If the key change concerns combination keys, then the procedure is rather straightforward. The change procedure isidentical to the procedure illustrated in Figure 14.2 on page 155, using the current value of the combination key as link key. This procedure can be carried out at any time after the authentication and encryption start. In fact, since the combination key corresponds to a single link, it can be modified each time this link is established. This will improve the security of the system since then old keys lose their validity after each session.

Of course, starting up an entirely new initialization procedure is also a possibility. In that case, user interaction is necessary since a PIN is required in the authentication and encryption procedures.

14.2.2.8 Generating a master key

The key-change routines described so far are semi-permanent. To create the master link key, which can replace the current link key during an initiated session (see Section 14.2.2.6 on page 156), other means are needed. First, the master creates a new link key from two 128-bit random numbers, RAND1 and RAND2. This is done by

$$K_{master} = E_{22}(\text{RAND1}, \text{RAND2}, 16). \qquad \text{(EQ 27)}$$

Clearly, this key is a 128-bit random number. The reason to use the output of E_{22} and not directly chose a random number as the key, is to avoid possible problems with degraded randomness due to a poor implementation of the random number generator within the Bluetooth unit.

Then, a third random number, say RAND, is transmitted to the slave. Using E_{22} with the current link key and RAND as inputs, both the master and slave computes a 128-bit overlay. The master sends the bitwise XOR of the overlay and the new link key to the slave. The slave, who knows the overlay, recalculates K_{master}. To confirm the success of this transaction, the units shall perform a mutual authentication procedure using the new link key . This procedure is then repeated for each slave who shall receive the new link key. The ACO values from the involved authentications should not replace the current existing ACO as this ACO is needed to (re)compute a ciphering key when the master wants to fall back to the previous link (non-temporary) key.

When so required – and potentially long after the actual distribution of the master key – the master activates encryption by an LM command. Before doing that, the master must ensure that all slaves receive the same random number,

say EN_RAND, since the encryption key is derived through the means of E_3 individually in all participating units. Then, each slave computes a new encryption key,

$$K_C = E_3(K_{master}, \text{EN\_RAND}, \text{COF}), \quad \text{(EQ 28)}$$

where the value of COF is derived from the master's BD_ADDR as specified by equation (EQ 26) on page 155. The details on the encryption key generating function can be found in Section 14.5.4 on page 176. The principle of the message flow between the master and slave when generating the master key is depicted in Figure 14.3. Note that in this case the ACO produced during the authentication is not used when computing the ciphering key.

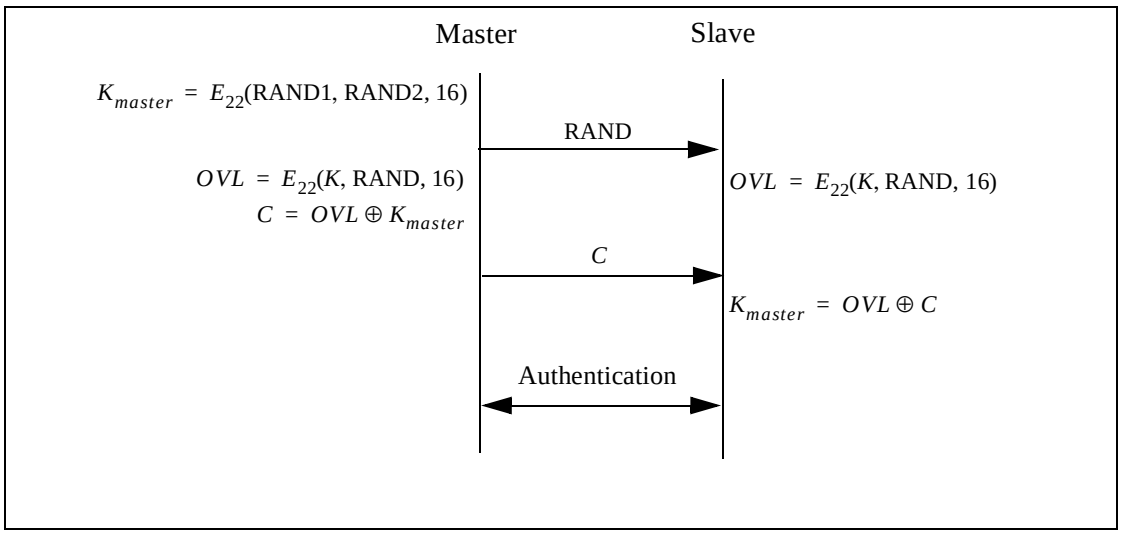


Figure 14.3: Master link key distribution and computation of the corresponding encryption key.

14.3 ENCRYPTION

User information can be protected by encryption of the packet payload; the access code and the packet header are never encrypted. The encryption of the payloads is carried out with a stream cipher called E_0 that is re-synchronized for every payload. The overall principle is shown in Figure 14.4 on page 159.

The stream cipher system E_0 consists of three parts. One part performs the initialization (generation of the payload key), the second part generates the key stream bits, and the third part performs the encryption and decryption. The payload key generator is very simple – it merely combines the input bits in an appropriate order and shift them into the four LFSRs used in the key stream generator. The main part of the cipher system is the second, as it also will be used for the initialization. The key stream bits are generated by a method derived from the summation stream cipher generator attributable to Massey and Rueppel. The method has been thoroughly investigated, and there exist good estimates of its strength with respect to presently known methods for cryptanalysis. Although the summation generator has weaknesses that can be

used in so-called correlation attacks, the high re-synchronization frequency will disrupt such attacks.

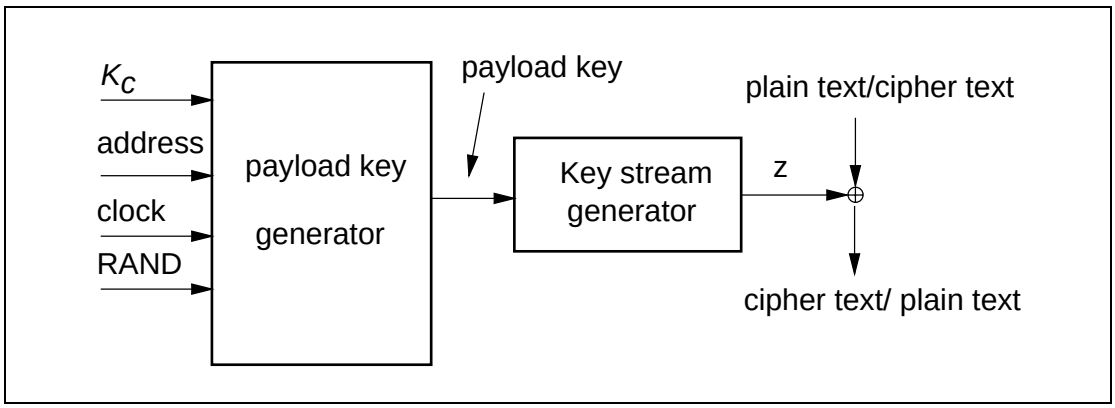


Figure 14.4: Stream ciphering for Bluetooth with E_0.

14.3.1 Encryption key size negotiation

Each Bluetooth device implementing the baseband specification needs a parameter defining the maximal allowed key length, L_{max}, $1 \le L_{max} \le 16$ (number of octets in the key). For each application, a number L_{min} is defined indicating the smallest acceptable key size for that particular application. Before generating the encryption key, the involved units must negotiate to decide what key size to actually use.

The master sends a suggested value, $L_{sug}^{(M)}$, to the slave. Initially, the suggested value is set to $L_{max}^{(M)}$. If $L_{min}^{(S)} \le L_{sug}^{(M)}$, and, the slave supports the suggested length, the slave acknowledges and this value will be the length of the encryption key for this link. However, if both conditions are not fulfilled, the slave sends a new proposal, $L_{sug}^{(S)}$ to the master. This value should be the largest among all supported lengths less than the previous master suggestion. Then, the master performs the corresponding test on the slave suggestion. This procedure is repeated until a key length agreement is reached, or, one unit aborts the negotiation. An abortion may be caused by lack of support for L_{sug} and all smaller key lengths, or if $L_{sug} < L_{min}$ in one of the units. In case of abortion Bluetooth link encryption can not be employed.

The possibility of a failure in setting up a secure link is an unavoidable consequence of letting the application decide whether to accept or reject a suggested key size. However, this is a necessary precaution. Otherwise a fraudulent unit could enforce a weak protection on a link by claiming a small maximum key size.

14.3.2 Encryption modes

If a slave has a semi-permanent link key (i.e. a combination key or a unit key), it can only accept encryption on slots individually addressed to itself (and, of course, in the reverse direction to the master). In particular, it will assume that broadcast messages are not encrypted. The possible traffic modes are described in Table 14.2. When an entry in the table refers to a link key, it means that the encryption/decryption engine uses the encryption key derived from that link key.

Broadcast traffic	Individually addressed traffic
No encryption	No encryption
No encryption	Encryption, Semi-permanent link key

Table 14.2: Possible traffic modes for a slave using a semi-permanent link key.

If the slave has received a master key, there are three possible combinations as defined in Table 14.3 on page 160. In this case, all units in the piconet uses a common link key, K_{master}. Since the master uses encryption keys derived from this link key for all secure traffic on the piconet, it is possible to avoid ambiguity in the participating slaves on which encryption key to use. Also in this case the default mode is that broadcast messages are not encrypted. A specific LM-command is required to activate encryption – both for broadcast and for individually addressed traffic.

Broadcast traffic	Individually addressed traffic
No encryption	No encryption
No encryption	Encryption, K_{master}
Encryption, K_{master}	Encryption, K_{master}

Table 14.3: Possible encryption modes for a slave in possession of a master key.

The master can issue an LM-command to the slaves telling them to fall back to their previous semi-permanent link key. Then, regardless of the previous mode they were in, they will end up in the first row of Table 14.2 on page 160; i.e. no encryption.

14.3.3 Encryption concept

For the encryption routine, a stream cipher algorithm will be used in which ciphering bits are bit-wise modulo-2 added to the data stream to be sent over the air interface. The payload is ciphered after the CRC bits are appended, but, prior to the FEC encoding.

Each packet payload is ciphered separately. The cipher algorithm E_0 uses the master Bluetooth address, 26 bits of the master real-time clock (CLK $_{26\text{-}1}$) and the encryption key K_C as input, see Figure 14.5 on page 161 (where it is assumed that unit A is the master).

The encryption key K_C is derived from the current link key, COF, and a random number, EN_RAND_A (see Section 14.5.4 on page 176). The random number is issued by the master before entering encryption mode. Note that EN_RAND_A is publicly known since it is transmitted as plain text over the air.

Within the E_0 algorithm, the encryption key K_C is modified into another key denoted K'_C. The maximum effective size of this key is factory preset and may be set to any multiple of eight between one and sixteen (8-128 bits). The procedure for deriving the key is described in Section 14.3.5 on page 164.

The real-time clock is incremented for each slot. The E_0 algorithm is re-initialized at the start of each new packet (i.e. for Master-to-Slave as well as for Slave-to-Master transmission). By using $CLK_{26\text{-}1}$ at least one bit is changed between two transmissions. Thus, a new keystream is generated after each re-initialization. For packets covering more than a single slot, the Bluetooth clock as found in the first slot is being used for the entire packet.

The encryption algorithm E_0 generates a binary keystream, K_{cipher}, which is modulo-2 added to the data to be encrypted. The cipher is symmetric; decryption is performed in exactly the same way using the same key as used for encryption.

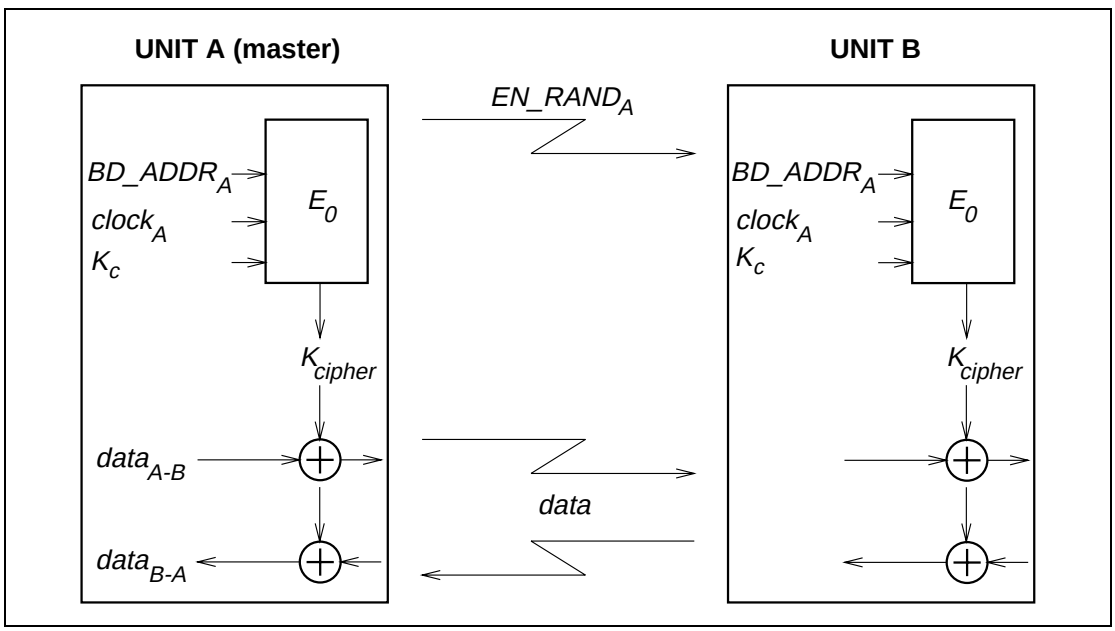


Figure 14.5: Functional description of the encryption procedure

14.3.4 Encryption algorithm

The system uses linear feedback shift registers (LFSRs) whose output is combined by a simple finite state machine (called the summation combiner) with 16 states. The output of this state machine is the key stream sequence, or, during initialization phase, the randomized initial start value. The algorithm is presented with an encryption key K_C, an 48-bit Bluetooth address, the master clock bits $CLK_{26\text{-}1}$, and a 128-bit RAND value. Figure 14.6 on page 162 shows the setup.

There are four LFSRs ($LFSR_1$,...,$LFSR_4$) of lengths $L_1 = 25$, $L_2 = 31$, $L_3 = 33$, and, $L_4 = 39$, with feedback polynomials as specified in Table 14.4 on page 163. The total length of the registers is 128. These polynomials are all primitive. The Hamming weight of all the feedback polynomials is chosen to be five – a reasonable trade-off between reducing the number of required XOR gates in the hardware realization and obtaining good statistical properties of the generated sequences.

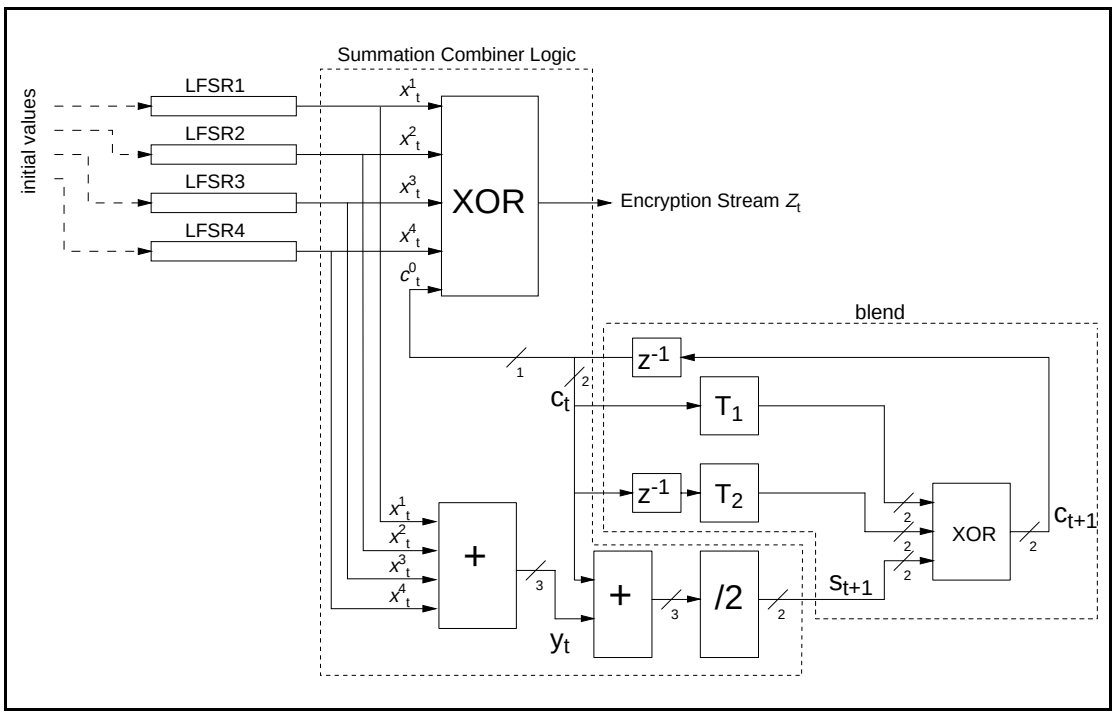


Figure 14.6: Concept of the encryption engine.

i	L_i	**feedback** $f_i(t)$	**weight**
1	25	$t^{25} + t^{20} + t^{12} + t^8 + 1$	5
2	31	$t^{31} + t^{24} + t^{16} + t^{12} + 1$	5
3	33	$t^{33} + t^{28} + t^{24} + t^4 + 1$	5
4	39	$t^{39} + t^{36} + t^{28} + t^4 + 1$	5

Table 14.4: The four primitive feedback polynomials.

Let x_t^i denote the tth symbol of LSFR$_i$. From the four-tuple $x_t^1, \ldots, x_t^4$ we derive the value y_t as

$$y_t = \sum_{i=1}^{4} x_t^i, \qquad \text{(EQ 29)}$$

where the sum is over the integers. Thus y_t can take the values 0,1,2,3, or 4. The output of the summation generator is now given by the following equations

$$z_t = x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus c_t^0 \in \{0, 1\}, \qquad \text{(EQ 30)}$$

$$s_{t+1} = (s_{t+1}^1, s_{t+1}^0) = \left\lfloor \frac{y_t + c_t}{2} \right\rfloor \in \{0, 1, 2, 3\}, \qquad \text{(EQ 31)}$$

$$c_{t+1} = (c_{t+1}^1, c_{t+1}^0) = s_{t+1} \oplus T_1[c_t] \oplus T_2[c_{t-1}], \qquad \text{(EQ 32)}$$

where $T_1[.]$ and $T_2[.]$ are two different linear bijections over GF(4). Suppose GF(4) is generated by the irreducible polynomial $x^2 + x + 1$, and let α be a zero of this polynomial in GF(4). The mappings T_1 and T_2 are now defined as

$$T_1 : \text{GF}(4) \rightarrow \text{GF}(4)$$

$$x \mapsto x$$

$$T_2 : \text{GF}(4) \rightarrow \text{GF}(4)$$

$$x \mapsto (\alpha + 1)x.$$

We can write the elements of GF(4) as binary vectors. This is summarized in Table 14.5.

Since the mappings are linear, we can realize them using XOR gates; i.e.

x	$T_1[x]$	$T_2[x]$
00	00	00
01	01	11
10	10	01
11	11	10

Table 14.5: The mappings T_1 and T_2.

$$T_1: \quad (x_1, x_0) \mapsto (x_1, x_0),$$
$$T_2: \quad (x_1, x_0) \mapsto (x_0, x_1 \oplus x_0).$$

14.3.4.1 The operation of the cipher

Figure 14.7 on page 164 gives an overview of the operation in time. The encryption algorithm shall run through the initialization phase before the start of transmitting or receiving a new packet. Thus, for multislot packets the cipher is initialized using the clock value of the first slot in the multislot sequence.

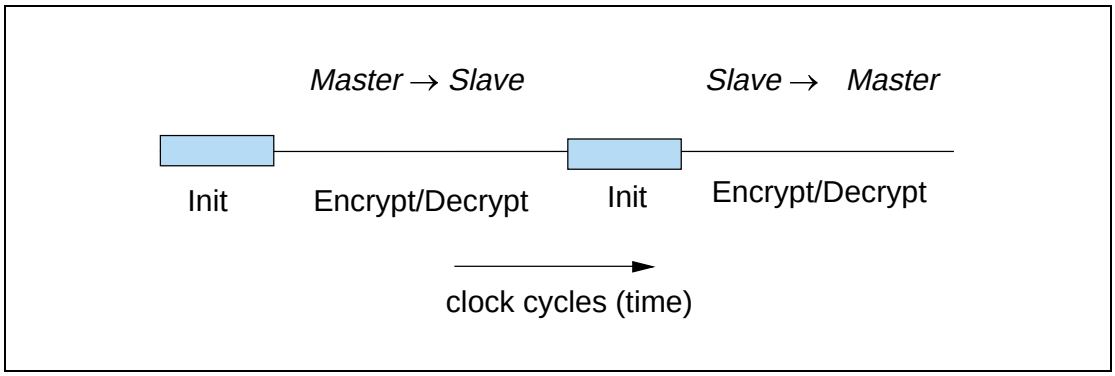


Figure 14.7: Overview of the operation of the encryption engine. Between each start of a packet (TX or RX), the LFSRs are re-initialized.

14.3.5 LFSR initialization

The key stream generator needs to be loaded with an initial value for the four LFSRs (in total 128 bits) and the 4 bits that specify the values of c_0 and c_{-1}. The 132 bit initial value is derived from four inputs by using the key stream generator itself. The input parameters are the key K_C, a 128-bit random number RAND, a 48-bit Bluetooth address, and the 26 master clock bits $CLK_{26\text{-}1}$.

The effective length of the encryption key can vary between 8 and 128 bits. Note that the actual key length as obtained from E_3 is 128 bits. Then, within E_0, the key length is reduced by a modulo operation between K_C and a polynomial of desired degree. After reduction, the result is encoded with a block code in

order to distribute the starting states more uniformly. The operation is defined in (EQ 33) on page 165.

When the encryption key has been created the LFSRs are loaded with their initial values. Then, 200 stream cipher bits are created by operating the generator. Of these bits, the last 128 are fed back into the key stream generator as an initial value of the four LFSRs. The values of c_t and c_{t-1} are kept. From this point on, when clocked the generator produces the encryption (decryption) sequence which is bitwise XORed to the transmitted (received) payload data.

In the following, we will denote octet i of a binary sequence X by the notation $X[i]$. We define bit 0 of X to be the LSB. Then, the LSB of $X[i]$ corresponds to bit $8i$ of the sequence X, the MSB of $X[i]$ is bit $8i + 7$ of X. For instance, bit 24 of the Bluetooth address is the LSB of ADR[3].

The details of the initialization are as follows:

1. Create the encryption key to use from the 128-bit secret key K_C and the 128-bit publicly known EN_RAND. Let $L, 1 \le L \le 16$, be the effective key length in number of octets. The resulting encryption key will be denoted K'_C:

$$K'_C(x) = g_2^{(L)}(x)(K_C(x) \bmod g_1^{(L)}(x)), \qquad \text{(EQ 33)}$$

   where $\deg(g_1^{(L)}(x)) = 8L$ and $\deg(g_2^{(L)}(x)) \le 128 - 8L$. The polynomials are defined in Table 14.6.

2. Shift in the 3 inputs K'_C, the Bluetooth address, the clock, and the six-bit constant 111001 into the LFSRs. In total 208 bits are shifted in.

   a) Open all switches shown in Figure 14.8 on page 167;

   b) Arrange inputs bits as shown in Figure 14.8; Set the content of all shift register elements to zero. Set $t = 0$.

   c) Start shifting bits into the LFSRs. The rightmost bit at each level of Figure 14.8 is the first bit to enter the corresponding LFSR.

   d) When the first input bit at level i reaches the rightmost position of $LFSR_i$, close the switch of this LFSR.

   e) At $t = 39$ (when the switch of $LFSR_4$ is closed), reset both blend registers $c_{39} = c_{39-1} = 0$; Up to this point, the content of c_t and c_{t-1} has been of no concern. However, from this moment forward their content will be used in computing the output sequence.

   f) From now on output symbols are generated. The remaining input bits are continuously shifted into their corresponding shift register. When the last bit has been shifted in, the shift register is clocked with input = 0;

      Note: When finished, $LFSR_1$ has effectively clocked 30 times with feedback closed, $LFSR_2$ has clocked 24 times, $LFSR_3$ has

clocked 22 times, and $LFSR_4$ has effectively clocked 16 times with feedback closed.

3. To mix initial data, continue to clock until 200 symbols have been produced with all switches closed $t = 239$ ();
4. Keep blend registers c_t and c_{t+1} make a parallel load of the last 128 generated bits into the LFSRs according to Figure 14.9 at $t = 240$;

After the parallel load in item 4, the blend register contents will be updated for each subsequent clock.

L	deg	$g_1^{(L)}$	deg	$g_2^{(L)}$
1	[8]	00000000 00000000 00000000 0000011d	[119]	00e275a0 abd218d4 cf928b9b bf6cb08f
2	[16]	00000000 00000000 00000000 0001003f	[112]	0001e3f6 3d7659b3 7f18c258 cff6efef
3	[24]	00000000 00000000 00000000 010000db	[104]	000001be f66c6c3a b1030a5a 1919808b
4	[32]	00000000 00000000 00000001 000000af	[96]	00000001 6ab89969 de17467f d3736ad9
5	[40]	00000000 00000000 00000100 00000039	[88]	00000000 01630632 91da50ec 55715247
6	[48]	00000000 00000000 00010000 00000291	[77]	00000000 00002c93 52aa6cc0 54468311
7	[56]	00000000 00000000 01000000 00000095	[71]	00000000 000000b3 f7fffce2 79f3a073
8	[64]	00000000 00000001 00000000 0000001b	[63]	00000000 00000000 a1ab815b c7ec8025
9	[72]	00000000 00000100 00000000 00000609	[49]	00000000 00000000 0002c980 11d8b04d
10	[80]	00000000 00010000 00000000 00000215	[42]	00000000 00000000 0000058e 24f9a4bb
11	[88]	00000000 01000000 00000000 0000013b	[35]	00000000 00000000 0000000c a76024d7
12	[96]	00000001 00000000 00000000 000000dd	[28]	00000000 00000000 00000000 1c9c26b9
13	[104]	00000100 00000000 00000000 0000049d	[21]	00000000 00000000 00000000 0026d9e3
14	[112]	00010000 00000000 00000000 0000014f	[14]	00000000 00000000 00000000 00004377
15	[120]	01000000 00000000 00000000 000000e7	[7]	00000000 00000000 00000000 00000089
16	[128]	1 00000000 00000000 00000000 00000000	[0]	00000000 00000000 00000000 00000001

Table 14.6: Polynomials used when creating K'_C.

All polynomials are in hexadecimal notation. The LSB is in the rightmost position.

In Figure 14.8, all bits are shifted into the LFSRs, starting with the least significant bit (LSB). For instance, from the third octet of the address, ADR[2], first ADR_{16} is entered, followed by ADR_{17}, etc. Furthermore, CL_0 corresponds to CLK_1,..., CL_{25} corresponds to CLK_{26}.

Note that the output symbols x_t^i, $i = 1, \ldots, 4$ are taken from the positions 24, 24, 32, and 32 for $LFSR_1$, $LFSR_2$,$LFSR_3$, and $LFSR_4$, respectively (counting the leftmost position as number 1).

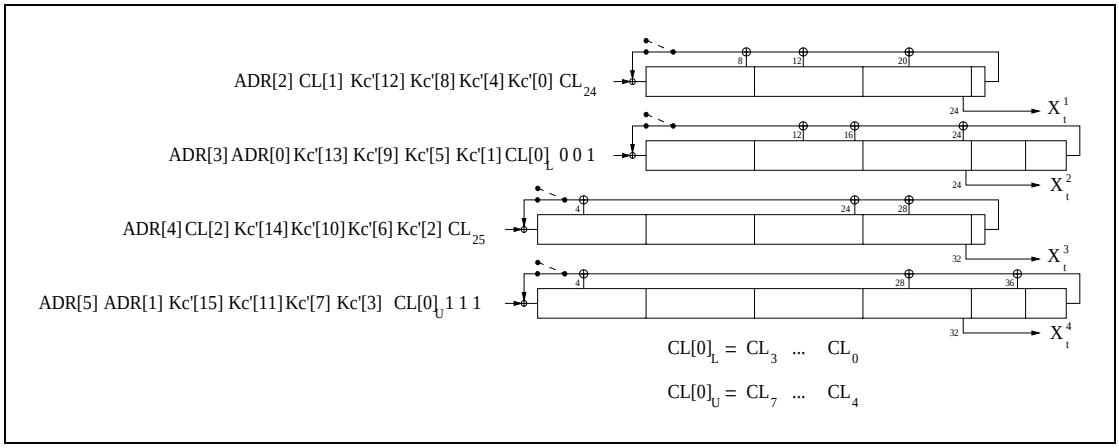


Figure 14.8: Arranging the input to the LFSRs.

In Figure 14.9, the 128 binary output symbols Z_0,..., Z_{127} are arranged in octets denoted Z[0],..., Z[15]. The LSB of Z[0] corresponds to the first of these symbols, the MSB of Z[15] is the latest output from the generator. These bits shall be loaded into the LFSRs according to the figure. It is a parallel load and no update of the blend registers is done. The first output symbol is generated at the same time. The octets are written into the registers with the LSB in the leftmost position (i.e. the opposite of before). For example, Z_{24} is loaded into position 1 of $LFSR_4$.

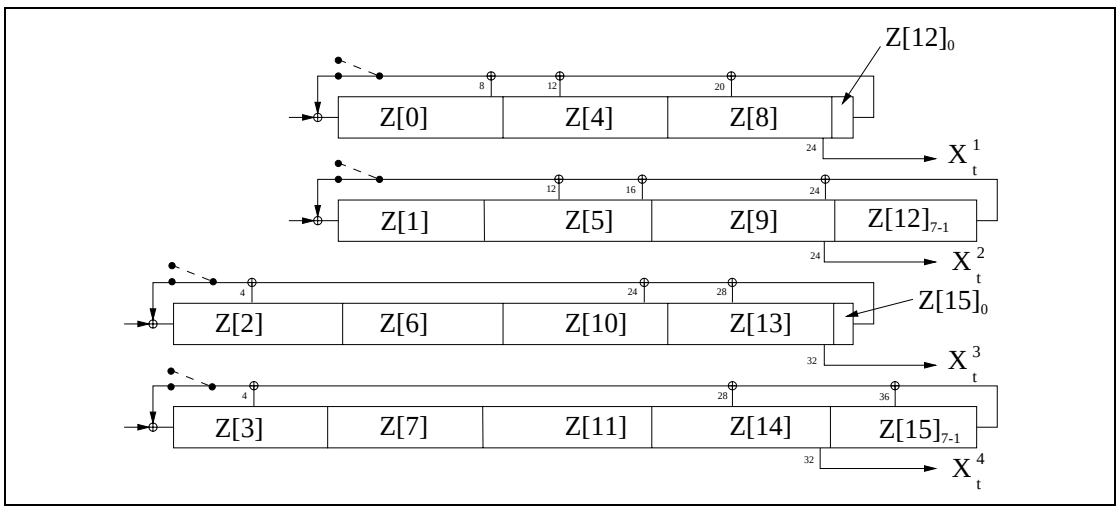


Figure 14.9: Distribution of the 128 last generated output symbols within the LFSRs.

14.3.6 Key stream sequence

When the initialization is finished, the output from the summation combiner is used for encryption/decryption. The first bit to use is the one produced at the parallel load, i.e. at $t = 240$. The circuit is run for the entire length of the current payload. Then, before the reverse direction is started, the entire initialization process is repeated with updated values on the input parameters.

Sample data of the encryption output sequence can be found in "Appendix IV" on page 923, Encryption Sample Data. A necessary, but not sufficient, condition for all Bluetooth-compliant implementations is to produce these encryption streams for identical initialization values.

14.4 AUTHENTICATION

The entity authentication used in Bluetooth uses a challenge-response scheme in which a claimant's knowledge of a secret key is checked through a 2-move protocol using symmetric secret keys. The latter implies that a correct claimant/verifier pair share the same secret key, for example K. In the challenge-response scheme the verifier challenges the claimant to authenticate a random input (the challenge), denoted by AU_RAND_A, with an authentication code, denoted by E_1, and return the result SRES to the verifier, see Figure 14.10 on page 168. This figure shows also that in Bluetooth the input to E_1 consists of the tuple AU_RAND_A and the Bluetooth device address (BD_ADDR) of the claimant. The use of this address prevents a simple reflection attack[1]. The secret K shared by units A and B is the current link key.

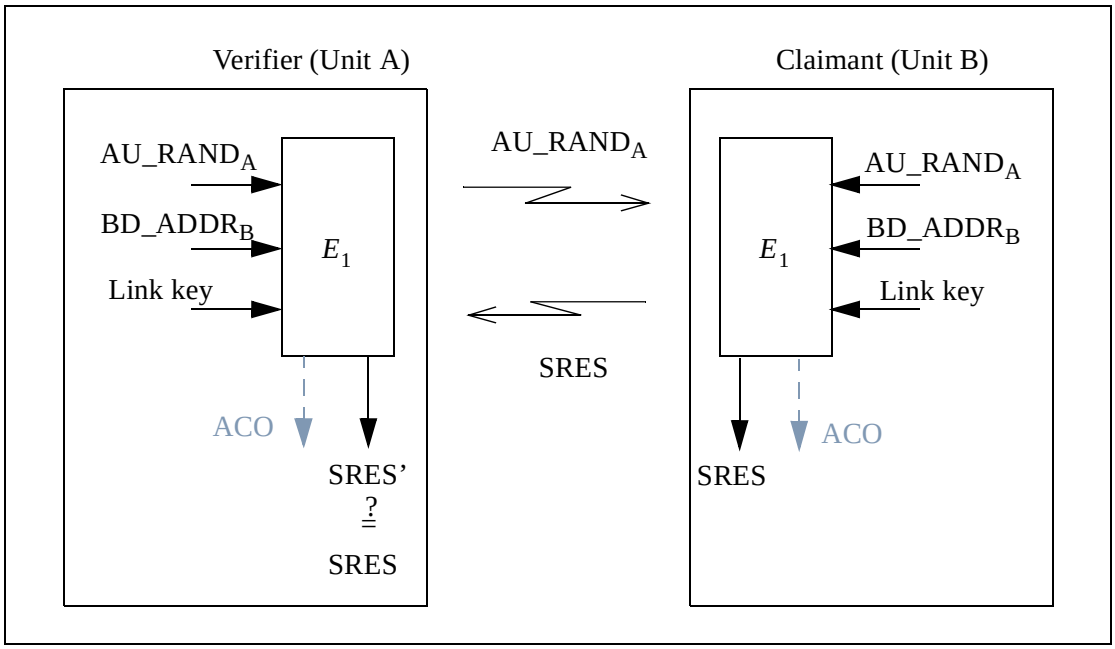


Figure 14.10: Challenge-response for the Bluetooth.

The challenge-response scheme for symmetric keys used in the Bluetooth is depicted in Figure 14.11 on page 169.

1. The reflection attack actually forms no threat in Bluetooth because all service requests are dealt with on a FIFO bases. When præmption is introduced, this attack is potentially dangerous.

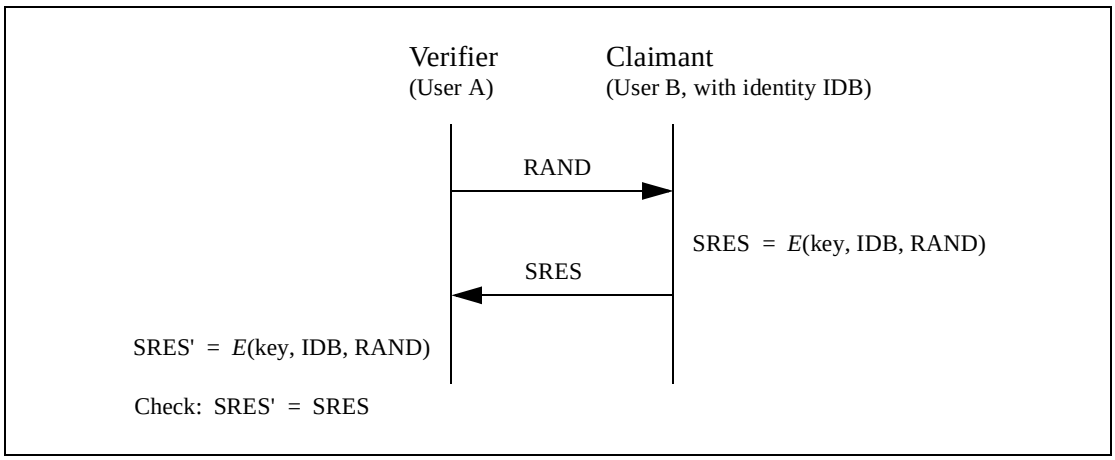


Figure 14.11: Challenge-response for symmetric key systems.

In the Bluetooth, the verifier is not necessarily the master. The application indicates who has to be authenticated by whom. Certain applications only require a one-way authentication. However, in some peer-to-peer communications, one might prefer a mutual authentication in which each unit is subsequently the challenger (verifier) in two authentication procedures. The LM coordinates the indicated authentication preferences by the application to determine in which direction(s) the authentication(s) has to take place. For mutual authentication with the units of Figure 14.10 on page 168, after unit A has successfully authenticated unit B, unit B could authenticate unit A by sending a AU_RAND_B (different from the AU_RAND_A that unit A issued) to unit A, and deriving the SRES and SRES' from the new AU_RAND_B, the address of unit A, and the link key K_{AB}.

If an authentication is successful the value of ACO as produced by E_1 should be retained.

14.4.1 Repeated attempts

When the authentication attempt fails, a certain waiting interval must pass before the verifier will initiate a new authentication attempt to the same claimant, or before it will respond to an authentication attempt initiated by a unit claiming the same identity as the suspicious unit. For each subsequent authentication failure with the same Bluetooth address, the waiting interval shall be increased exponentially. That is, after each failure, the waiting interval before a new attempt can be made, for example, twice as long as the waiting interval prior to the previous attempt[1]. The waiting interval shall be limited to a maximum. The maximum waiting interval depends on the implementation. The waiting time shall exponentially decrease to a minimum when no new failed attempts are being made during a certain time period. This procedure prevents

1. An other appropriate value larger than 1 may be used.

an intruder to repeat the authentication procedure with a large number of different keys.

To make the system somewhat less vulnerable to denial-of-service attacks, the Bluetooth units should keep a list of individual waiting intervals for each unit it has established contact with. Clearly, the size of this list must be restricted only to contain the N units with which the most recent contact has been made. The number N can vary for different units depending on available memory size and user environment.

14.5 THE AUTHENTICATION AND KEY-GENERATING FUNCTIONS

This section describes the algorithmic means for supporting the Bluetooth security requirements on authentication and key generation.

14.5.1 The authentication function E_1

The authentication function E_1 proposed for the Bluetooth is a computationally secure authentication code, or often called a MAC. E_1 uses the encryption function called SAFER+. The algorithm is an enhanced version[1] of an existing 64-bit block cipher SAFER-SK128, and it is freely available. In the sequel the block cipher will be denoted as the function A_r which maps under a 128-bit key, a 128-bit input to a 128-bit output, i.e.

$$\begin{aligned} A_r\colon\ \{0,1\}^{128}\times\{0,1\}^{128} &\rightarrow \{0,1\}^{128} \\ (k\times x) &\mapsto t. \end{aligned} \qquad \text{(EQ 34)}$$

The details of A_r are given in the next section. The function E_1 is constructed using A_r as follows

$$\begin{aligned} E_1\colon\ \{0,1\}^{128}\times\{0,1\}^{128}\times\{0,1\}^{48} &\rightarrow \{0,1\}^{32}\times\{0,1\}^{96} \\ (K,\ \mathrm{RAND},\ \mathrm{address}) &\mapsto (\mathrm{SRES},\ \mathrm{ACO}), \end{aligned} \qquad \text{(EQ 35)}$$

where $\mathrm{SRES} = Hash(K, \mathrm{RAND}, \mathrm{address}, 6)[0, \ldots, 3]$, where $Hash$ is a keyed hash function defined as[2],

$$\begin{aligned} &Hash\colon\{0,1\}^{128}\times\{0,1\}^{128}\times\{0,1\}^{8\times L}\times\{6,12\}\rightarrow\{0,1\}^{128} \\ &(K, I_1, I_2, L)\mapsto A'_r([\tilde{K}], [E(I_2, L)+_{16}(A_r(K, I_1)\oplus_{16} I_1)]), \end{aligned} \qquad \text{(EQ 36)}$$

1. It is presently one of the contenders for the Advanced Encryption Standard (AES) submitted by Cylink, Corp, Sunnyvale, USA
2. The operator $+_{16}$ denotes bytewise addition mod 256 of the 16 octets, and the operator $\oplus_{16}$ denotes bytewise XORing of the 16 octets.

and where

$$E:\{0,1\}^{8\times L}\times\{6,12\}\rightarrow\{0,1\}^{8\times 16}$$
$$(X[0,\ldots,L-1],L)\mapsto(X[i(\text{mod L})]\ \text{ for } i=0...15), \quad \text{(EQ 37)}$$

is an expansion of the L octet word X into a 128-bit word. Thus we see that we have to evaluate the function A_r twice for each evaluation of E_1. The key $\tilde{K}$ for the second use of A_r (actually A'_r) is offseted from K as follows[1]

$$\begin{aligned}
\tilde{K}[0] &= (K[0]+233) \mod 256, & \tilde{K}[1] &= K[1]\oplus 229,\\
\tilde{K}[2] &= (K[2]+223) \mod 256, & \tilde{K}[3] &= K[3]\oplus 193,\\
\tilde{K}[4] &= (K[4]+179) \mod 256, & \tilde{K}[5] &= K[5]\oplus 167,\\
\tilde{K}[6] &= (K[6]+149) \mod 256, & \tilde{K}[7] &= K[7]\oplus 131,\\
\tilde{K}\{8\} &= K[8]\oplus 233, & \tilde{K}[9] &= (K[9]+229) \mod 256,\\
\tilde{K}[10] &= K[10]\oplus 223, & \tilde{K}[11] &= (K[11]+193) \mod 256,\\
\tilde{K}[12] &= K[12]\oplus 179, & \tilde{K}[13] &= (K[13]+167) \mod 256,\\
\tilde{K}[14] &= K[14]\oplus 149, & \tilde{K}[15] &= (K[15]+131) \mod 256.
\end{aligned} \quad \text{(EQ 38)}$$

A data flowchart of the computation of E_1 is depicted in Figure 14.12 on page 172. E_1 is also used to deliver the parameter ACO (Authenticated Ciphering Offset) that is used in the generation of the ciphering key by E_3, see equations (EQ 26) on page 155 and (EQ 46) on page 176. The value of ACO is formed by the octets 4 through 15 of the output of the hash function defined in (EQ 36) on page 170, i.e.

$$\text{ACO} = Hash(K, \text{RAND}, \text{address}, 6)[4,\ldots,15]. \quad \text{(EQ 39)}$$

1. The constants are the first largest primes below 257 for which 10 is a primitive root.

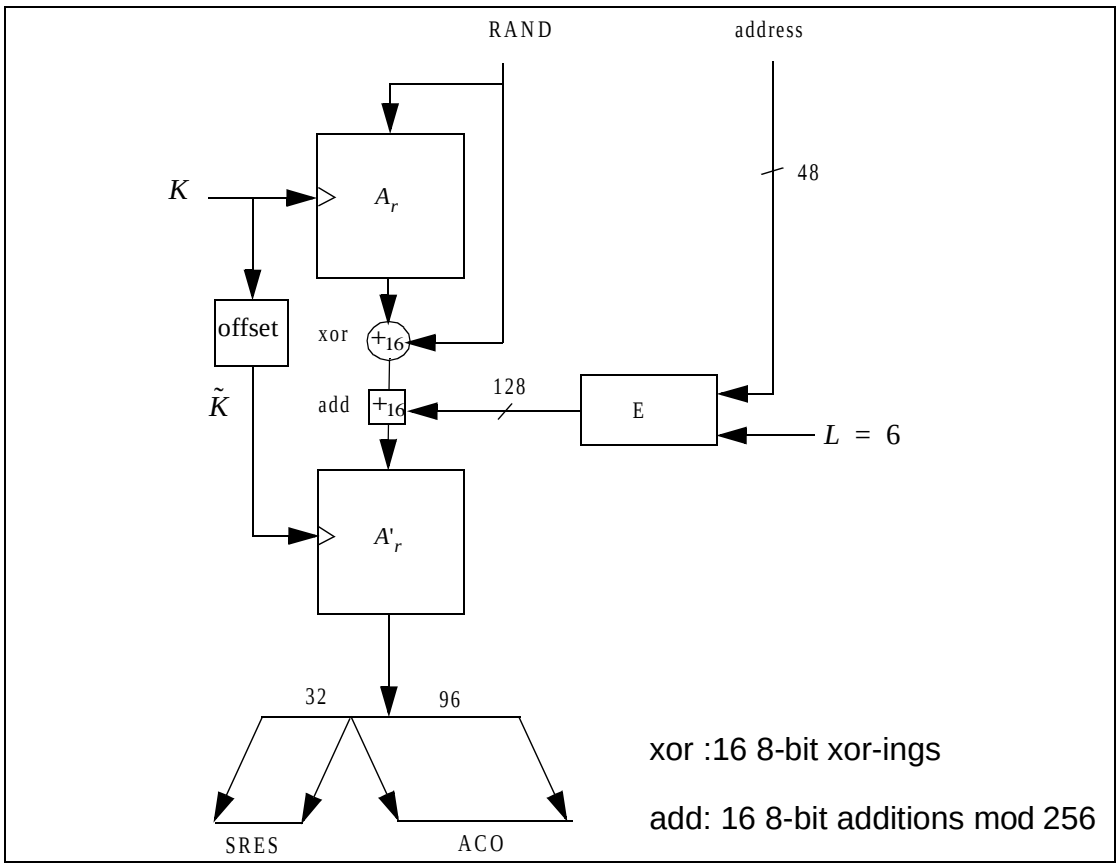


Figure 14.12: Flow of data for the computation of E_1.

14.5.2 The functions A_r and A'_r

The function A_r is identical to SAFER+. It consists of a set of 8 layers, (each layer is called a round) and a parallel mechanism for generating the sub keys $K_p[j]$, $p = 1, 2, \ldots, 17$, the so-called round keys to be used in each round. The function will produce a 128-bit result from a 128-bit "random" input string and a 128-bit "key". Besides the function A_r, a slightly modified version referred to as A'_r is used in which the input of round 1 is added to the input of the 3rd round. This is done to make the modified version non-invertible and prevents the use of A'_r (especially in E_{2x}) as an encryption function. See Figure 14.13 on page 173 for details.

14.5.2.1 The round computations

The computations in each round are a composition of encryption with a round key, substitution, encryption with the next round key, and, finally, a Pseudo Hadamard Transform (PHT). The computations in a round are shown in Figure 14.13 on page 173. The sub keys for round $r, r = 1, 2, \ldots, 8$ are denoted

$K_{2r-1}[j]$, $K_{2r}[j]$, $j = 0, 1, \ldots, 15$. After the last round $k_{17}[j]$ is applied in a similar fashion as all previous odd numbered keys.

14.5.2.2 The substitution boxes "e" and "l"

In Figure 14.13 on page 173 two boxes occur, marked "e" and "l". These boxes implement the same substitutions as used in SAFER+; i.e. they implement

$$\begin{aligned} e, l &: \{0, \ldots, 255\} \rightarrow \{0, \ldots, 255\}, \\ e &: i \mapsto (45^i \pmod{257}) \pmod{256}, \\ l &: i \mapsto j \text{ s.t. } i = e(j). \end{aligned}$$

Their role, as in the SAFER+ algorithm, is to introduce non-linearity.

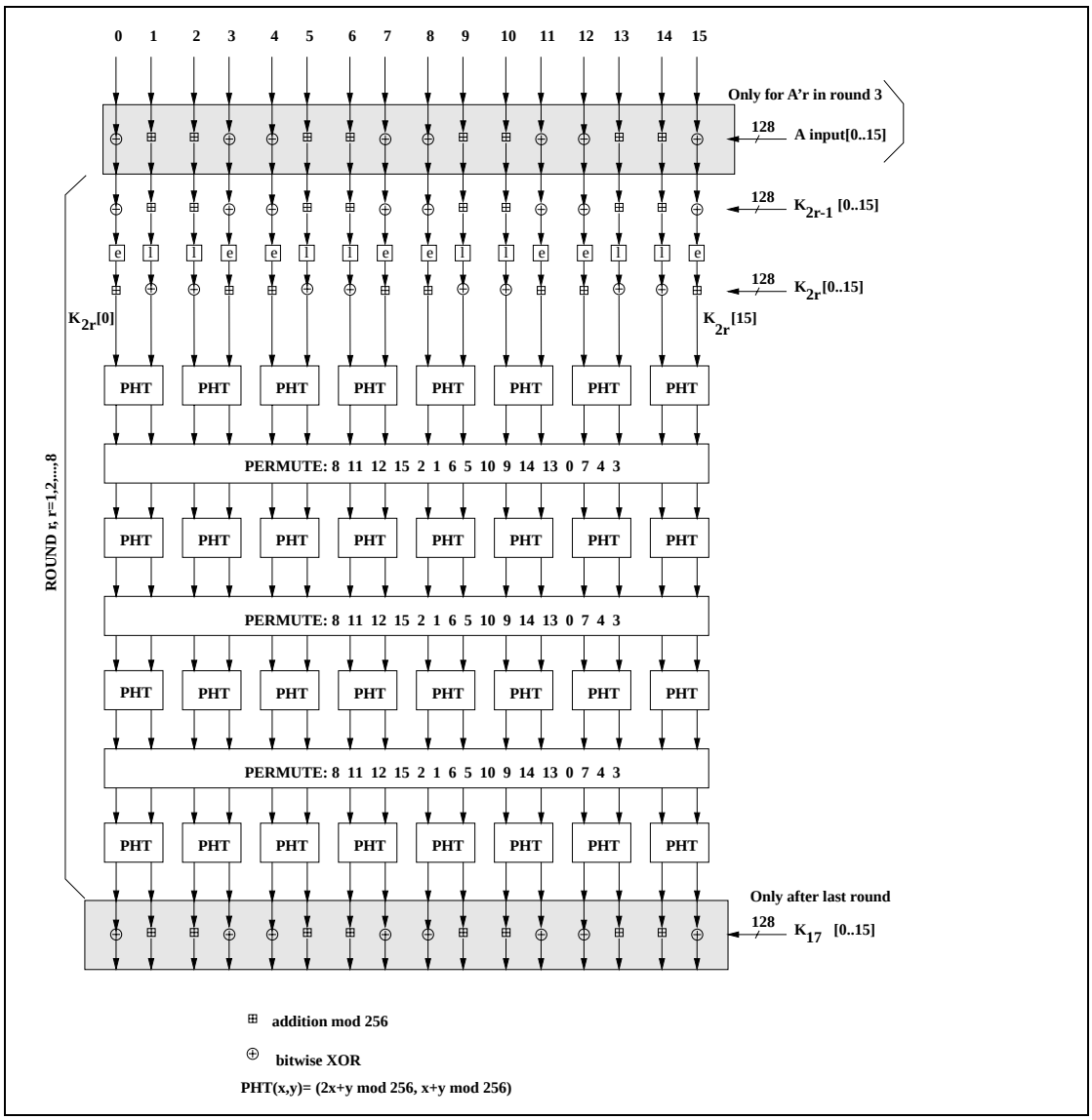


Figure 14.13: One round in A_r and A'_r. The permutation boxes show how input byte indices are mapped onto output byte indices. Thus, position 0 (leftmost) is mapped on position 8, position 1 is mapped on position 11, et cetera.

14.5.2.3 Key scheduling

In each round, 2 batches of 16 octet-wide keys are needed. These so-called round keys are derived as specified by the key scheduling in SAFER+. Figure 14.14 on page 174 gives an overview of how the round keys $K_p[j]$ are determined. The bias vectors B_2, B_3, ..., B_{17} are computed according to following equation:

$$B_p[i] = \left(\left(45^{(45^{17p+i+1} \bmod 257)} \bmod 257\right) \bmod 256\right), \text{ for } i = 0, \ldots, 15. \qquad \text{(EQ 40)}$$

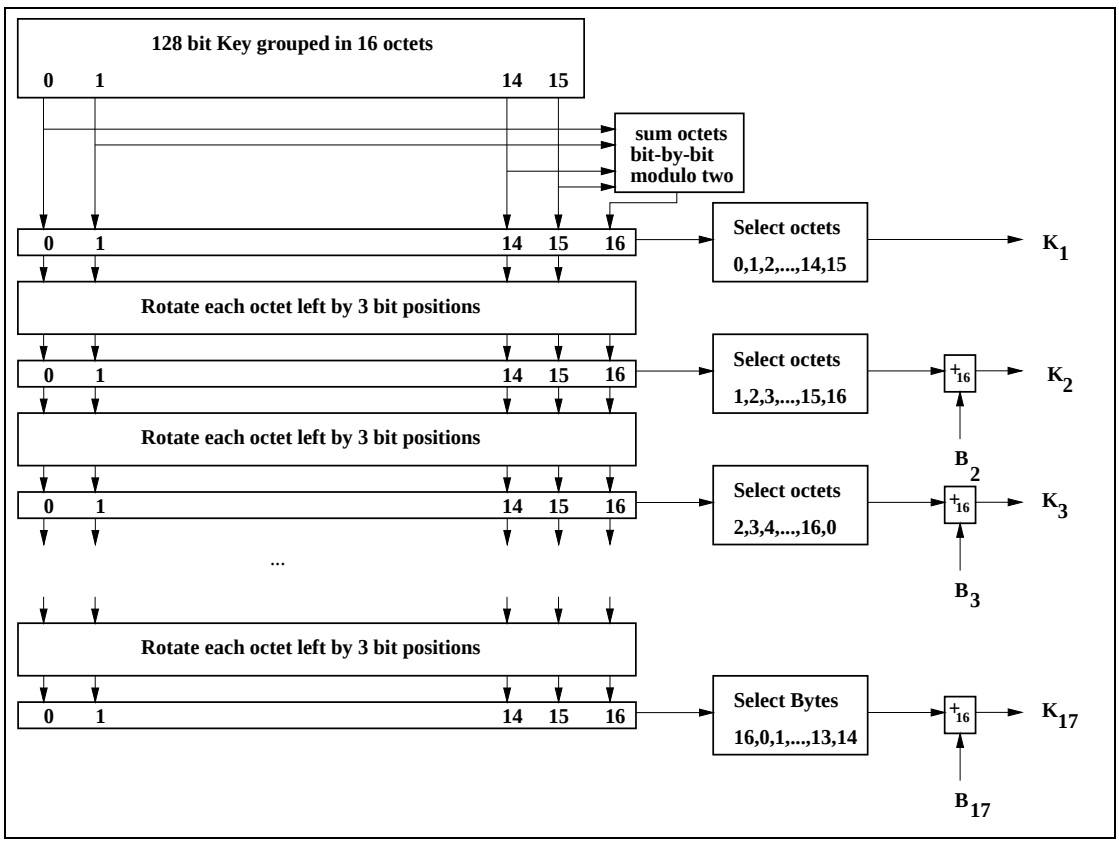


Figure 14.14: Key scheduling in A_r .

14.5.3 E_2-Key generation function for authentication

The key used for authentication is derived through a procedure that is shown in Figure 14.15 on page 176. The figure shows two different modes of operation for the algorithm. In the first mode, the function E_2 should produce on input of a 128-bit RAND value and a 48-bit address, a 128-bit link key K. This mode is utilized when creating unit keys and combination keys. In the second mode the function E_2 should produce, on input of a 128-bit RAND value and an L-octet user PIN, a 128-bit link key K. The second mode is used to create the initialization key, and also whenever a master key is to be generated.

When the initialization key is generated, the PIN is augmented with the BD_ADDR, see Section 14.2.2.1 on page 153 for which address to use. The augmentation always starts with the least significant octet of the address immediately following the most significant octet of the PIN. Since the maximum length of the PIN used in the algorithm cannot exceed 16 octets, it is possible that not all octets of BD_ADDR will be used.

This key generating algorithm again exploits the cryptographic function. Formally E_2 can be expressed for mode 1 (denoted E_{21}) as

$$E_{21}: \{0,1\}^{128} \times \{0,1\}^{48} \rightarrow \{0,1\}^{128}$$
$$(\text{RAND}, \text{address}) \mapsto A'_r(X, Y) \qquad \text{(EQ 41)}$$

where (for mode 1)

$$\begin{cases} X = \text{RAND}[0 \ldots 14] \cup (\text{RAND}[15] \oplus 6) \\ Y = \bigcup_{i=0}^{15} \text{address}[i \pmod 6] \end{cases} \qquad \text{(EQ 42)}$$

Let L be the number of octets in the user PIN. The augmenting is defined by

$$\text{PIN}' = \begin{cases} \text{PIN}[0 \ldots L-1] \cup \text{BD\_ADDR}[0 \ldots \min\{5, 15-L\}], & L < 16, \\ \text{PIN}[0 \ldots L-1], & L = 16, \end{cases} \qquad \text{(EQ 43)}$$

Then, in mode 2, E_2 (denoted E_{22}) can be expressed as

$$E_{22}: \{0,1\}^{8L'} \times \{0,1\}^{128} \times \{1, 2, \ldots, 16\} \rightarrow \{0,1\}^{128}$$
$$(\text{PIN}', \text{RAND}, \text{L}') \mapsto A'_r(X, Y) \qquad \text{(EQ 44)}$$

where

$$\begin{cases} X = \bigcup_{i=0}^{15} \text{PIN}'[i \pmod{L'}], \\ Y = \text{RAND}[0 \ldots 14] \cup (\text{RAND}[15] \oplus L'), \end{cases} \qquad \text{(EQ 45)}$$

and $L' = \min\{16, L+6\}$ is the number of octets in PIN'.

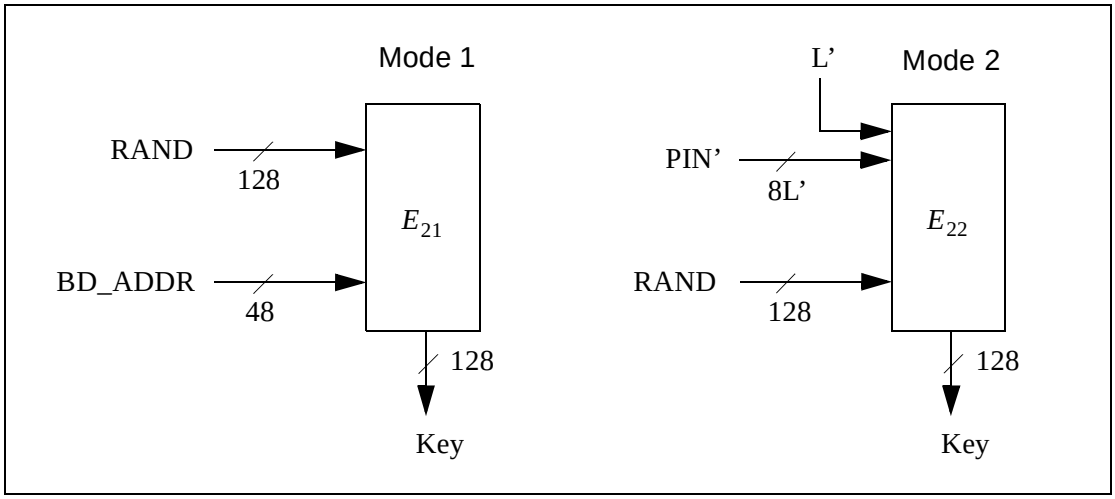


Figure 14.15: Key generating algorithm E_2 and its two modes. Mode 1 is used for unit and combination keys, while mode 2 is used for K_{init} and K_{master}.

14.5.4 E_3-Key generation function for encryption

The ciphering key K_C used by E_0 is generated by E_3. The function E_3 is constructed using A'_r as follows

$$E_3: \{0,1\}^{128} \times \{0,1\}^{128} \times \{0,1\}^{96} \rightarrow \{0,1\}^{128}$$
$$(K, \text{RAND}, \text{COF}) \mapsto Hash(K, \text{RAND}, \text{COF}, 12) \qquad \text{(EQ 46)}$$

where $Hash$ is the hash function as defined by (EQ 36) on page 170. Note that the produced key length is 128 bits. However, before use within E_0, the encryption key K_C will be shortened to the correct encryption key length, as described in Section 14.3.5 on page 164. A block scheme of E_3 is depicted in Figure 14.16.

The value of COF is determined as specified by equation (EQ 26) on page 155.

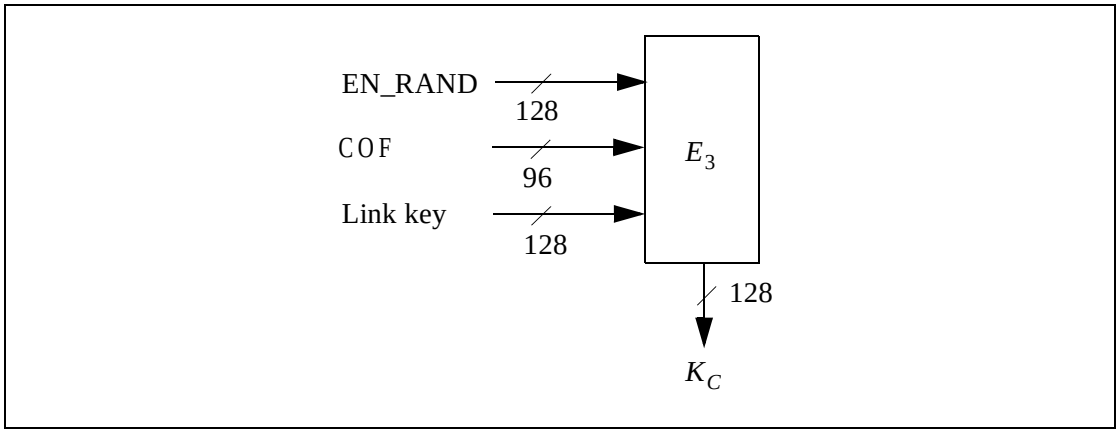


Figure 14.16: Generation of the encryption key.

15 LIST OF FIGURES

Figure 1.1: Different functional blocks in the Bluetooth system 41
Figure 1.2: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c). 42

Figure 2.1: TDD and timing 44
Figure 2.2: Multi-slot packets 44

Figure 4.1: Standard packet format. 47
Figure 4.2: Access code format 48
Figure 4.3: Preamble 49
Figure 4.4: Trailer in CAC when MSB of sync word is 0 (a), and when MSB of sync word is 1 (b). 50
Figure 4.5: Header format. 51
Figure 4.6: Format of the FHS payload 56
Figure 4.7: DV packet format 59
Figure 4.8: Payload header format for single-slot packets. 62
Figure 4.9: Payload header format for multi-slot packets. 62

Figure 5.1: Bit-repetition encoding scheme. 67
Figure 5.2: LFSR generating the (15,10) shortened Hamming code. 68
Figure 5.3: Receive protocol for determining the ARQN bit. 69
Figure 5.4: Retransmit filtering for packets with CRC. 71
Figure 5.5: Broadcast repetition scheme 72
Figure 5.6: The LFSR circuit generating the HEC. 73
Figure 5.7: Initial state of the HEC generating circuit. 74
Figure 5.8: HEC generation and checking. 74
Figure 5.9: The LFSR circuit generating the CRC. 74
Figure 5.10: Initial state of the CRC generating circuit. 75
Figure 5.11: CRC generation and checking 75

Figure 7.1: Data whitening LFSR. 78

Figure 8.1: Functional diagram of TX buffering. 79
Figure 8.2: Functional diagram of RX buffering 82
Figure 8.3: Header bit processes. 84
Figure 8.4: Payload bit processes. 84

Figure 9.1: RX/TX cycle of Bluetooth master transceiver in normal mode for single-slot packets. 86
Figure 9.2: RX/TX cycle of Bluetooth slave transceiver in normal mode for single-slot packets. 87

Figure 9.3: RX timing of slave returning from hold state. 88
Figure 9.4: RX/TX cycle of Bluetooth transceiver in PAGE mode. 89
Figure 9.5: Timing of FHS packet on successful page in first half slot. 90
Figure 9.6: Timing of FHS packet on successful page in second half slot. .91
Figure 9.7: RX/TX timing in multi-slave configuration 92

Figure 10.1: Bluetooth clock. 94
Figure 10.2: Derivation of CLKE 95
Figure 10.3: Derivation of CLK in master (a) and in slave (b). 95
Figure 10.4: State diagram of Bluetooth link controller. 96
Figure 10.5: Conventional page (a), page while one SCO link present (b), page while two SCO links present (c). 101
Figure 10.6: Messaging at initial connection when slave responds to first page message. 103
Figure 10.7: Messaging at initial connection when slave responds to second page message. 103
Figure 10.8: General beacon channel format 115
Figure 10.9: Definition of access window 115
Figure 10.10:Access procedure applying the polling technique. 116
Figure 10.11:Disturbance of access window by SCO traffic 116
Figure 10.12:Extended sleep interval of parked slaves. 118

Figure 11.1: General block diagram of hop selection scheme. 128
Figure 11.2: Hop selection scheme in CONNECTION state. 128
Figure 11.3: Block diagram of hop selection kernel for the 79-hop system. 129
Figure 11.4: Block diagram of hop selection kernel for the 23-hop system. 129
Figure 11.5: XOR operation for the 79-hop system. The 23-hop system is the same except for the Z’4/Z4 wire that does not exist. 130
Figure 11.6: Permutation operation for the 79 hop system. 132
Figure 11.7: Permutation operation for the 23 hop system. 132
Figure 11.8: Butterfly implementation. 132

Figure 12.1: Block diagram of CVSD encoder with syllabic companding. ...140
Figure 12.2: Block diagram of CVSD decoder with syllabic companding. ...140
Figure 12.3: Accumulator procedure 140

Figure 13.1: Format of BD_ADDR 143
Figure 13.2: Construction of the sync word. 145
Figure 13.3: LFSR and the starting state to generate 147

Figure 14.1: Generation of unit key. When the unit key has been exchanged, the initialization key shall be discarded in both units. 155

Figure 14.2: Generating a combination key. The old link key (K) shall be discarded after the exchange of a new combination key has succeeded 156
Figure 14.3: Master link key distribution and computation of the corresponding encryption key. 159
Figure 14.4: Stream ciphering for Bluetooth with E0. 160
Figure 14.5: Functional description of the encryption procedure 162
Figure 14.6: Concept of the encryption engine. 163
Figure 14.7: Overview of the operation of the encryption engine. Between each start of a packet (TX or RX), the LFSRs are re-initialized. 165
Figure 14.8: Arranging the input to the LFSRs. 168
Figure 14.9: Distribution of the 128 last generated output symbols within the LFSRs. 168
Figure 14.10: Challenge-response for the Bluetooth. 169
Figure 14.11: Challenge-response for symmetric key systems. 170
Figure 14.12: Flow of data for the computation of 173
Figure 14.13: One round in A_r and A'_r. 174
Figure 14.14: Key scheduling in A_r. 175
Figure 14.15: Key generating algorithm E_2 and its two modes. 177
Figure 14.16: Generation of the encryption key. 178

16 LIST OF TABLES

Table 4.1: Summary of access code types. 49
Table 4.2: Packets defined for SCO and ACL link types....54
Table 4.3: Description of the FHS payload 56
Table 4.4: Contents of SR field 57
Table 4.5: Contents of SP field 57
Table 4.6: Contents of page scan mode field....58
Table 4.7: Logical channel L_CH field contents....63
Table 4.8: Use of payload header flow bit on the logical channels. 64
Table 4.9: Link control packets 65
Table 4.10: ACL packets....65
Table 4.11: SCO packets....65

Table 10.1: Relationship between scan interval, train repetition, and paging modes R0, R1 and R2....98
Table 10.2: Relationship between train repetition, and paging modes R0, R1 and R2 when SCO links are present.....101
Table 10.3: Initial messaging during start-up. 102
Table 10.4: Increase of train repetition when SCO links are present. 109
Table 10.6: Mandatory scan periods for P0, P1, P2 scan period modes. ... 110
Table 10.5: Messaging during inquiry routines. 110

Table 11.1: Control of the butterflies for the 79 hop system 131
Table 11.2: Control of the butterflies for the 23 hop system 131
Table 11.3: Control for 79-hop system.....134
Table 11.4: Control for 23-hop system.....134

Table 12.1: Voice coding schemes supported on the air interface.....139
Table 12.2: CVSD parameter values. The values are based on a 16 bit signed number output from the accumulator.....141

Table 14.1: Entities used in authentication and encryption procedures.....149
Table 14.2: Possible traffic modes for a slave using a semi-permanent link key. 161
Table 14.3: Possible encryption modes for a slave in possession of a master key.....161
Table 14.4: The four primitive feedback polynomials.....164
Table 14.5: The mappings T_1 and T_2. 165
Table 14.6: Polynomials used when creating K'_C. 167

Part C

LINK MANAGER PROTOCOL

This specification describes the Link Manager Protocol (LMP) which is used for link set-up and control. The signals are interpreted and filtered out by the Link Manager on the receiving side and are not propagated to higher layers.

CONTENTS

1 General 191

2 Format of LMP 193

3 The Procedure Rules and PDUs 195

3.1 General Response Messages 195

3.2 Authentication 196

3.2.1 Claimant has link key 196

3.2.2 Claimant has no link key 197

3.2.3 Repeated attempts 197

3.3 Pairing 197

3.3.1 Responder accepts pairing 198

3.3.2 Responder has a fixed PIN 198

3.3.3 Responder rejects pairing 199

3.3.4 Creation of the link key 199

3.3.5 Repeated attempts 199

3.4 Change Link Key 200

3.5 Change the Current Link Key 201

3.5.1 Change to a temporary link key 201

3.5.2 Make the semi-permanent link key the current link key 202

3.6 Encryption 203

3.6.1 Encryption mode 203

3.6.2 Encryption key size 204

3.6.3 Start encryption 205

3.6.4 Stop encryption 205

3.6.5 Change encryption mode, key or random number 206

3.7 Clock Offset Request 206

3.8 Slot Offset Information 207

3.9 Timing Accuracy Information Request 207

3.10 LMP Version 209

3.11 Supported Features 209

3.12 Switch of Master-Slave Role 210

3.13 Name Request 212

3.14 Detach 213

3.15 Hold Mode 213

3.15.1 Master forces hold mode 214

3.15.2 Slave forces hold mode 214

3.15.3 Master or slave requests hold mode 215

3.16 Sniff Mode 216
3.16.1 Master or slave requests sniff mode 216
3.16.2 Moving a slave from sniff mode to active mode 217
3.17 Park Mode 218
3.17.1 Master requests slave to enter park mode 220
3.17.2 Slave requests to enter park mode 220
3.17.3 Slave requests to be placed in park mode 222
3.17.4 Master sets up broadcast scan window 222
3.17.5 Master modifies beacon parameters 223
3.17.6 Unparking slaves 223
3.18 Power Control 224
3.19 Channel Quality-driven Change Between DM and DH 225
3.20 Quality of Service (QoS) 227
3.20.1 Master notifies slave of the quality of service 227
3.20.2 Device requests new quality of service 228
3.21 SCO Links 228
3.21.1 Master initiates an SCO link 229
3.21.2 Slave initiates an SCO link 229
3.21.3 Master requests change of SCO parameters 230
3.21.4 Slave requests change of SCO parameters 230
3.21.5 Remove an SCO link 230
3.22 Control of Multi-slot Packets 231
3.23 Paging Scheme 232
3.23.1 Page mode 232
3.23.2 Page scan mode 232
3.24 Link Supervision 233

4 Connection Establishment 234

5 Summary of PDUs 236

5.1 Description of Parameters 241
5.1.1 Coding of features 244
5.1.2 List of error reasons 246
5.2 Default Values 247

6 Test Modes 248
6.1 Activation and Deactivation of Test Mode 248
6.2 Control of Test Mode 248
6.3 Summary of Test Mode PDUs 249
7 Error Handling 250
8 List of Figures 251
9 List of Tables 253

1 GENERAL

LMP messages are used for link set-up, security and control. They are transferred in the payload instead of L2CAP and are distinguished by a reserved value in the L_CH field of the payload header. The messages are filtered out and interpreted by LM on the receiving side and are not propagated to higher layers.

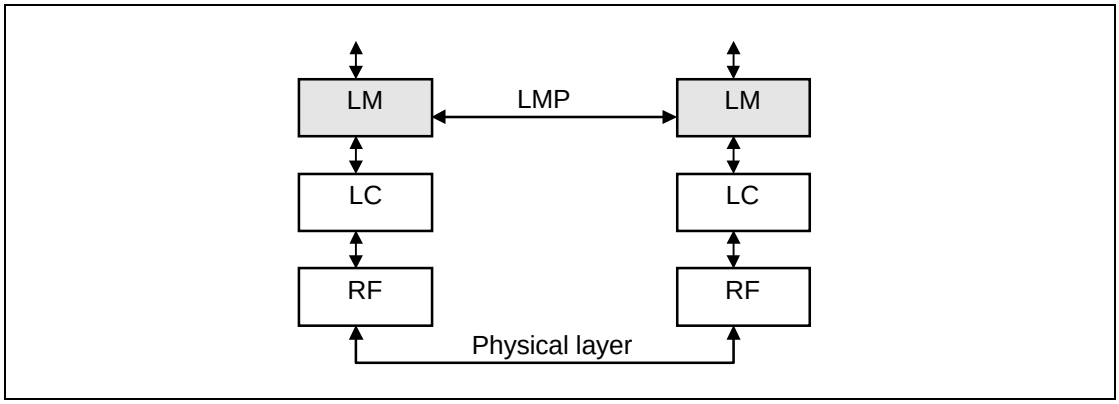


Figure 1.1: Link Manager's place on the global scene.

Link Manager messages have higher priority than user data. This means that if the Link Manager needs to send a message, it shall not be delayed by the L2CAP traffic, although it can be delayed by many retransmissions of individual baseband packets.

We do not need to explicitly acknowledge the messages in LMP since LC (see Baseband Specification Section 5, on page 66) provides us with a reliable link.

LC does not guarantee either the time taken to deliver a message to the remote device or the delay between the delivery of the message to the remote device and the reception of the corresponding ACK by the sender. This means that we must be aware of the underlying LC mechanism's limitations to synchronize state changes between master and slave. The criteria for determining when the master can reuse an AM_ADDR following the detach or park of a slave is based on the reception of the Baseband-level acknowledgement. Synchronization of a master-slave switch or the starting of hold mode utilizes the Bluetooth master clock, which the LM reads from the LC.

LC only guarantees that it will attempt to communicate with each slave once per T_{poll} slots.

T_{poll} is the poll interval as described in section 3.20 on page 225.

The time between receiving a baseband packet carrying an LMP PDU and sending a baseband packet carrying a valid response PDU, according to the

procedure rules in Section 3 on page 193, must be less than the LMP Response Timeout. The value of this timeout is 30 seconds. Note that the LMP Response Timeout is applied not only to sequences described in Section 3 on page 193, but also to the series of the sequences defined as the transactions in Section 3 on page 193. It is also applied to the series of the transactions, as long as no L2CAP PDUs are allowed, for example, any transactions until the PDUs LMP_setup_complete are exchanged.

2 FORMAT OF LMP

LM PDUs are always sent as single-slot packets and the payload header is therefore one byte. The two least significant bits in the payload header determine the logical channel. For LM PDUs these bits are set.

L_CH code	Logical Channel	Information
00	NA	undefined
01	UA/I	Continuing L2CAP message
10	UA/I	Start L2CAP message
11	LM	LMP message

Table 2.1: Logical channel L_CH field contents.

The FLOW bit in the payload header is always one and is ignored on the receiving side. Each PDU is assigned a 7-bit opcode used to uniquely identify different types of PDUs, see Table 5.1 on page 234. The opcode and a one-bit transaction ID are positioned in the first byte of the payload body. The transaction ID is positioned in the LSB. It is 0 if the PDU belongs to a transaction initiated by the master and 1 if the PDU belongs to a transaction initiated by the slave. If the PDU contains one or more parameters these are placed in the payload starting at the second byte of the payload body. The number of bytes used depends on the length of the parameters. If an SCO link is present using HV1 packets and length of *content* is less than 9 bytes the PDUs can be transmitted in DV packets. Otherwise DM1 packets must be used. All parameters have little endian format, i.e. the least significant byte is transmitted first.

The source/destination of the PDUs is determined by the AM_ADDR in the packet header.

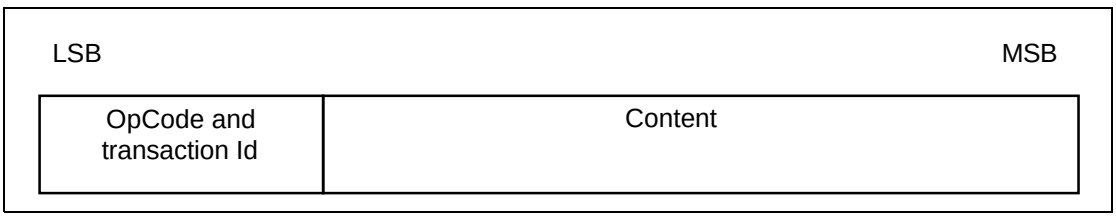


Figure 2.1: Payload body when LM PDUs are sent.

Each PDU is either mandatory or optional. The M/O field in the tables of Heading 1 indicates this. The LM does not need to be able to transmit a PDU that is optional. The LM must recognize all optional PDUs that it receives and, if a response is required, send a valid response according to the procedure rules in Heading 1. The reason that should be used in this case is *unsupported LMP feature*. If the optional PDU that is received does not require a response, no response is sent. Which of the optional PDUs a device supports can be requested, see Section 3.11 on page 207.

Each sequence described in section 3 is normally defined as a transaction. For pairing, see section 3.3, or encryption, see section 3.6, all sequences belonging to each section are counted as one transaction and shall use the same transaction ID. For connection establishment, see section 4, LMP_host_connection_req and the response with LMP_accepted or LMP_not_accepted form one transaction and have the transaction ID of 0. LMP_setup_complete is a stand-alone PDU, which forms a transaction by itself. For error handling, see section 7, the PDU to be rejected and LMP_not_accepted form a single transaction. Therefore the LMP_not_accepted shall have the same transaction ID as the PDU which is being rejected.

3 THE PROCEDURE RULES AND PDUs

Each procedure is described and depicted with a sequence diagram. The following symbols are used in the sequence diagrams:

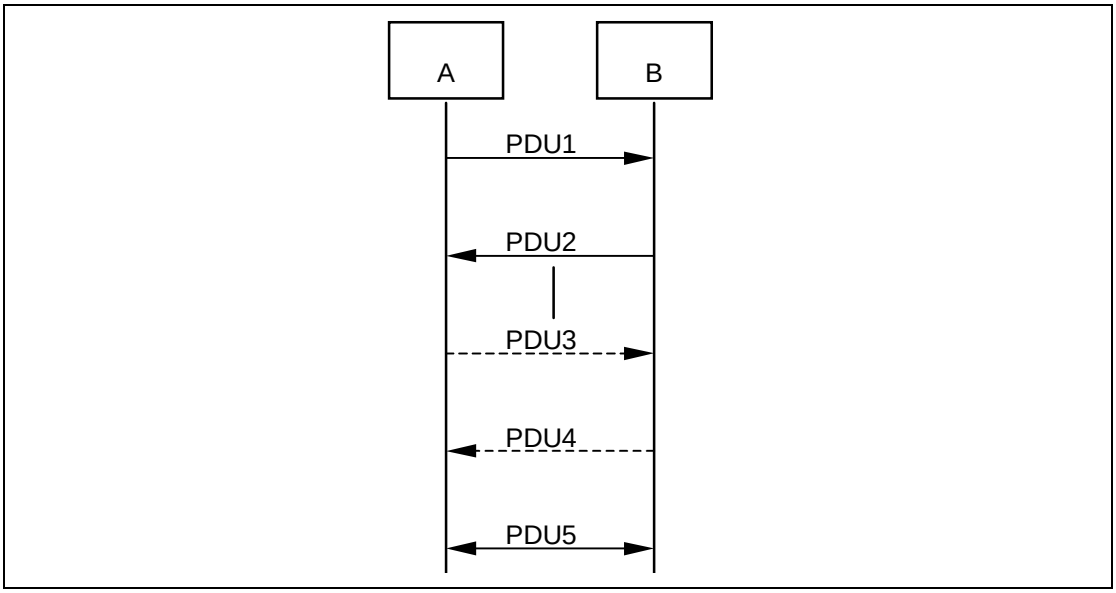


Figure 3.1: Symbols used in sequence diagrams.

PDU1 is a PDU sent from A to B. PDU2 is a PDU sent from B to A. PDU3 is a PDU that is optionally sent from A to B. PDU4 is a PDU that is optionally sent from B to A. PDU5 is a PDU sent from either A or B. A vertical line indicates that more PDUs can optionally be sent.

3.1 GENERAL RESPONSE MESSAGES

The PDUs LMP_accepted and LMP_not_accepted are used as response messages to other PDUs in a number of different procedures. The PDU LMP_accepted includes the opcode of the message that is accepted. The PDU LMP_not_accepted includes the opcode of the message that is not accepted and the reason why it is not accepted.

M/O	PDU	Contents
M	LMP_accepted	op code
M	LMP_not_accepted	op code reason

Table 3.1: General response messages.

3.2 AUTHENTICATION

The authentication procedure is based on a challenge-response scheme as described in Baseband Specification Section 14.4, on page 168. The verifier sends an LMP_au_rand PDU which contains a random number (the challenge) to the claimant. The claimant calculates a response, which is a function of the challenge, the claimant's BD_ADDR and a secret key. The response is sent back to the verifier, which checks if the response was correct or not. How the response should be calculated is described in Baseband Specification Section 14.5.1, on page 170. A successful calculation of the authentication response requires that two devices share a secret key. How this key is created is described in Section 3.3 on page 195. Both the master and the slave can be verifiers. The following PDUs are used in the authentication procedure:

M/O	PDU	Contents
M	LMP_au_rand	random number
M	LMP_sres	authentication response

Table 3.2: PDUs used for authentication.

3.2.1 Claimant has link key

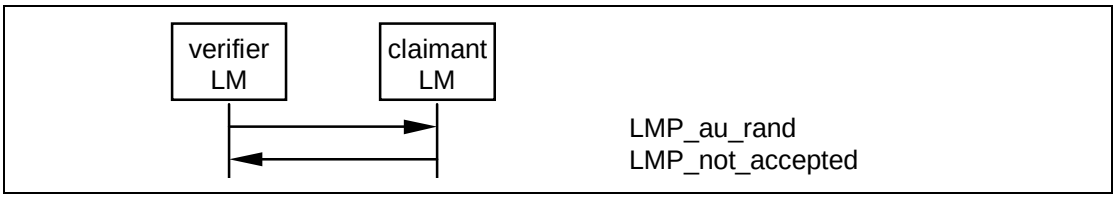
If the claimant has a link key associated with the verifier, it calculates the response and sends it to the verifier with LMP_sres. The verifier checks the response. If the response is not correct, the verifier can end the connection by sending LMP_detach with the reason code *authentication failure,* see Section 3.14 on page 211.

Sequence 1: Authentication. Claimant has link key.

If an LM receives LMP_au_rand and also wants to initiate an authentication it must first reply with LMP_sres before starting its own challenge. There can however be concurrent requests caused by master and slave simultaneously initiating an authentication. To avoid that this results in different ACOs in the units, this situation is resolved by the rule outlined in Section 7: If the master sends LMP_au_rand and receives another LMP_au_rand before receiving LMP_sres it shall respond with LMP_not_accepted with the reason code *LMP Error Transaction Collision;* in that case the slave LM shall complete the master's challenge by sending LMP_sres and may then initiate its authentication again.

3.2.2 Claimant has no link key

If the claimant does not have a link key associated with the verifier it sends LMP_not_accepted with the reason code *key missing* after receiving LMP_au_rand.



Sequence 2: Authentication fails. Claimant has no link key.

3.2.3 Repeated attempts

The scheme described in Baseband Specification Section 14.4.1, on page 169 shall be applied when an authentication fails. This will prevent an intruder from trying a large number of keys in a relatively short time.

3.3 PAIRING

When two devices do not have a common link key an initialization key (K_{init}) is created based on a PIN and a random number and a BD address. How the K_{init} is calculated is described in Baseband Specification Section 14.5.3, on page 174. When both devices have calculated K_{init} the link key is created, and finally a mutual authentication is made. The pairing procedure starts with a device sending LMP_in_rand; this device is referred to as "initiating LM" or "initiator" in Section 3.3.1 on page 196 - Section 3.3.5 on page 197. The other device is referred to as "responding LM" or "responder". The PDUs used in the pairing procedure are:

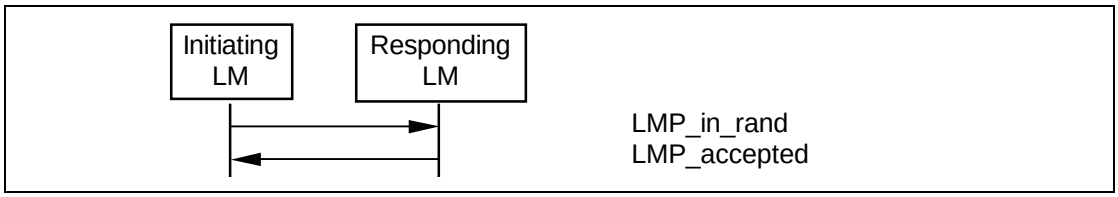
M/O	PDU	Contents
M	LMP_in_rand	random number
M	LMP_au_rand	random number
M	LMP_sres	authentication response
M	LMP_comb_key	random number
M	LMP_unit_key	key

Table 3.3: PDUs used for pairing

Note: all sequences described in section 3.3, including the mutual authentication after the link key has been created, form a single transaction. The transaction ID from the first LMP_in_rand will be used for all subsequent sequences.

3.3.1 Responder accepts pairing

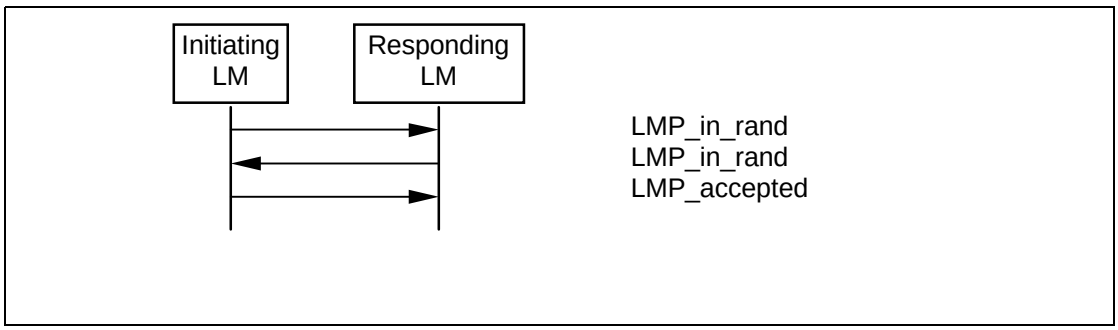
The initiator sends LMP_in_rand and the responder replies with LMP_accepted. Both devices calculate K_{init} based on the BD address of the responder and the procedure continues with creation of the link key; see Section 3.3.4 on page 197.



Sequence 3: Pairing accepted. Responder has a variable PIN.Initiator has a variable or fixed PIN.

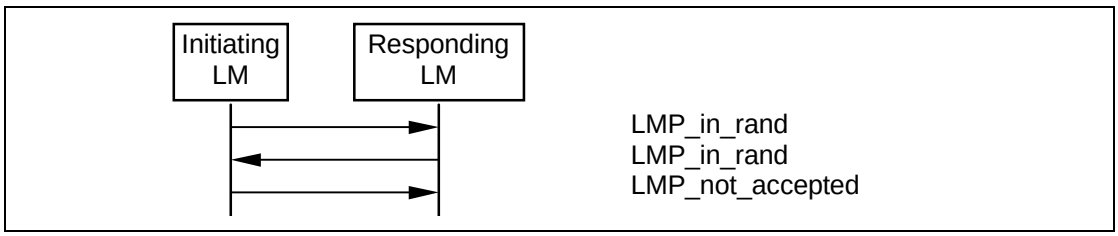
3.3.2 Responder has a fixed PIN

If the responder has a fixed PIN it shall generate a new random number and send it back in LMP_in_rand. If the initiator has a variable PIN it must accept this and respond with LMP_accepted. Both sides then calculate K_{init} based on the last IN_RAND and the BD address of the initiator. Thereafter the procedure continues with creation of the link key; see Section 3.3.4 on page 197.



Sequence 4: Responder has a fixed PIN and initiator has a variable PIN.

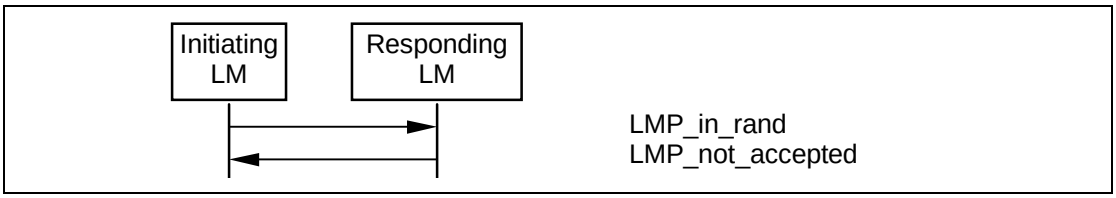
If the responder has a fixed PIN and the initiator also has a fixed PIN, the second LMP_in_rand is rejected by the initiator sending LMP_not_accepted with the reason code *pairing not allowed*; the pairing procedure is then ended.



Sequence 5: Both devices have a fixed PIN.

3.3.3 Responder rejects pairing

If the responder rejects pairing it sends LMP_not_accepted with the reason code *pairing not allowed* after receiving LMP_in_rand.



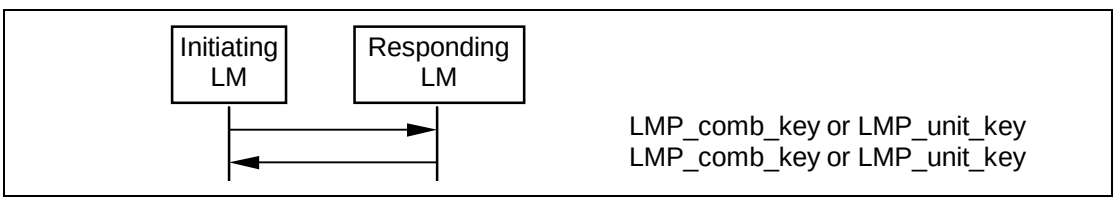
Sequence 6: Responder rejects pairing.

3.3.4 Creation of the link key

When K_{init} is calculated in both units the link key must be created. This link key will be used in the authentication between the two units for all subsequent connections until it is changed; see section 3.4 on page 198 and section 3.5 on page 199. The link key created in the pairing procedure will either be a combination key or one of the unit's unit keys. The following rules apply to the selection of the link key:

- if one unit sends LMP_unit_key and the other unit sends LMP_comb_key,the unit key will be the link key
- if both units send LMP_unit_key, the master's unit key will be the link key
- if both units send LMP_comb_key, the link key is calculated as described in Baseband Specification Section 14.2.2, on page 152.

The content of LMP_unit_key is the unit key bitwise XORed with K_{init}. The content of LMP_comb_key is LK_RAND bitwise XORed with K_{init}. Any device configured to use a combination key will store the link key.



Sequence 7: Creation of the link key.

3.3.5 Repeated attempts

When the authentication after creation of the link key fails because of a wrong authentication response, the same scheme as in Section 3.2.3 on page 195 is applied. This prevents an intruder from trying a large number of different PINs in a relatively short time.

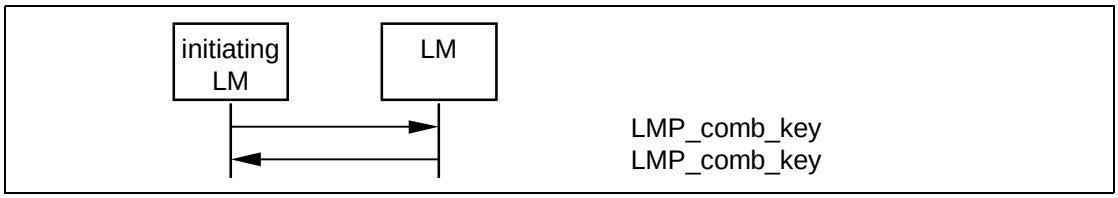
3.4 CHANGE LINK KEY

If the link key is derived from combination keys and the current link is the semi-permanent link key, the link key can be changed. If the link key is a unit key, the units must go through the pairing procedure in order to change the link key. The contents of LMP_comb_key is protected by a bitwise XOR with the current link key.

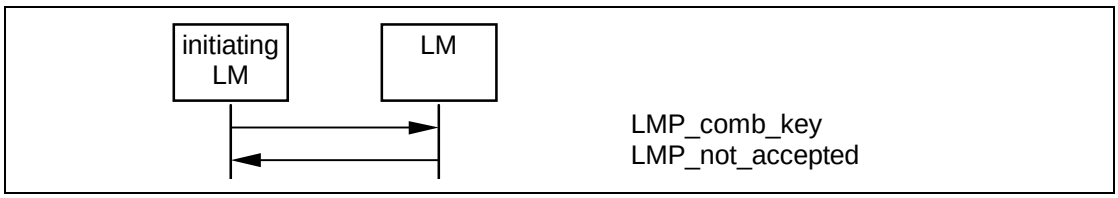
M/O	PDU	Contents
M	LMP_comb_key	random number

Table 3.4: PDUs used for change of link key.

Note: all sequences described in Section 3.4 on page 198, including the mutual authentication after the link key has been changed, form a single transaction. The transaction ID from the first LMP_comb_key will be used for all subsequent sequences.



Sequence 8: Successful change of the link key.



Sequence 9: Change of the link key not possible since the other unit uses a unit key.

If the change of link key is successful the new link key is stored and the old link key is discarded. The new link key will be used as link key for all the following connections between the two devices until the link key is changed again. The new link key also becomes the current link key. It will remain the current link key until the link key is changed again, or until a temporary link key is created, see Section 3.5 on page 199.

When the new link key has been created mutual authentication must be made to confirm that the same link key has been created in both units. The first authentication in the mutual authentication is made with the unit that initiated change link key as verifier. When finalized an authentication in the reversed direction is made.

3.5 CHANGE THE CURRENT LINK KEY

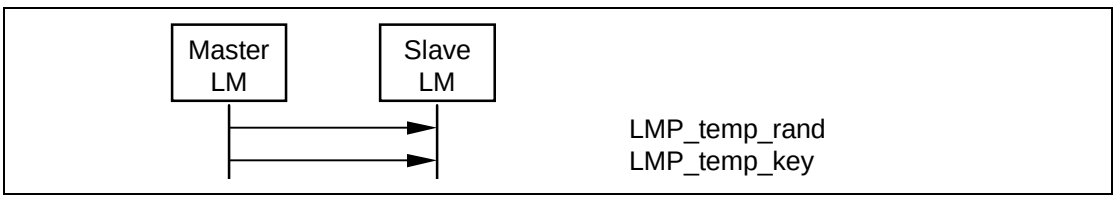
The current link key can be a semi-permanent link key or a temporary link key key. It can be changed temporarily, but the change is only valid for the session, see Baseband Specification Section 14.2.1, on page 150. Changing to a temporary link key is necessary if the piconet is to support encrypted broadcast.

M/O	PDU	Contents
M	LMP_temp_rand	random number
M	LMP_temp_key	key
M	LMP_use_semi_permanent_key	-

Table 3.5: PDUs used to change the current link key.

3.5.1 Change to a temporary link key

In the following, we use the same terms as in Baseband Specification Section 14.2.2.8, on page 157. The master starts by creating the master key K_{master} as described in Baseband Specification (EQ 27), on page 157. Then the master issues a random number RAND and sends it to the slave in LMP_temp_rand. Both sides can then calculate an overlay denoted OVL as OVL= E_{22}(current link key, RAND, 16). Then the master sends K_{master} protected by a modulo-2 addition with OVL to the slave in LMP_temp_key. The slave, who knows OVL, calculates K_{master}. After this, K_{master} becomes the current link key. It will be the current link key until the units fall back to the semi-permanent link key, see section 3.5.2 on page 200.



Sequence 10: Change to a temporary link key.

Note: all sequences described in section 3.4.1, including the mutual authentication after K_{master} has been created, form a single transaction. The transaction ID is set to 0.

When the units have changed to the temporary key a mutual authentication must be made to confirm that the same link key has been created in both units. The first authentication in the mutual authentication is made with the master as verifier. When finalized an authentication in the reversed direction is made.

3.5.2 Make the semi-permanent link key the current link key

After the current link key has been changed to K_{master}, this change can be undone and the semi-permanent link key becomes the current link key again. If encryption is used on the link, the procedure of going back to the semi-permanent link key must be immediately followed by a stop of the encryption by the master invoking the procedure described in Section 3.6.4 on page 203. Encryption can then be started again by the master according to the procedures in section 3.6.1 on page 201 subsection 3. This is to assure that encryption with encryption parameters known by other devices in the piconet is not used when the semi-permanent link key is the current link key.

Sequence 11: Link key changed to the semi-permanent link key.

3.6 ENCRYPTION

If at least one authentication has been performed encryption may be used. If the master wants all slaves in the piconet to use the same encryption parameters it must issue a temporary key (K_{master}) and make this key the current link key for all slaves in the piconet before encryption is started, see Section 3.5 on page 199. This is necessary if broadcast packets should be encrypted.

M/O	PDU	Contents
O	LMP_encryption_mode_req	encryption mode
O	LMP_encryption_key_size_req	key size
O	LMP_start_encryption_req	random number
O	LMP_stop_encryption_req	-

Table 3.6: PDUs used for handling encryption.

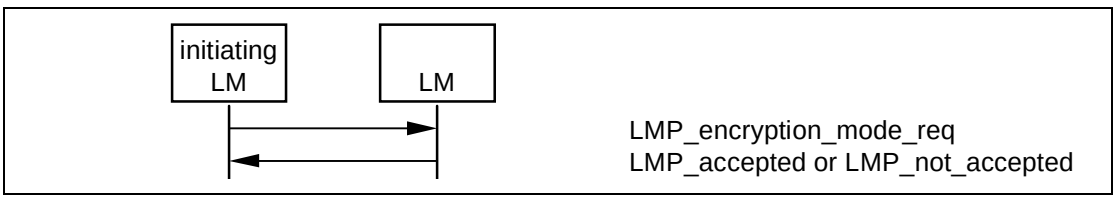
Note: all sequences described in Section 3.6 on page 201 form a single transaction. The transaction ID from the LMP_encryption_mode_req will be used for all subsequent sequences.

3.6.1 Encryption mode

First of all the master and the slave must agree upon whether to use encryption or not and if encryption shall only apply to point-to-point packets or if encryption shall apply to both point-to-point packets and broadcast packets. If master and slave agree on the encryption mode, the master continues to give more detailed information about the encryption.

The initiating LM finalizes the transmission of the current ACL packet with L2CAP information, stops L2CAP transmission and sends LMP_encryption_mode_req. If the change in encryption mode is accepted then the other device finalizes the transmission of the current ACL packet with L2CAP information, stops L2CAP transmission and responds with LMP_accepted.

L2CAP transmission is re-enabled when the attempt to encrypt or decrypt the link is completed i.e. at the end of Sequence 14, 15 or 16.

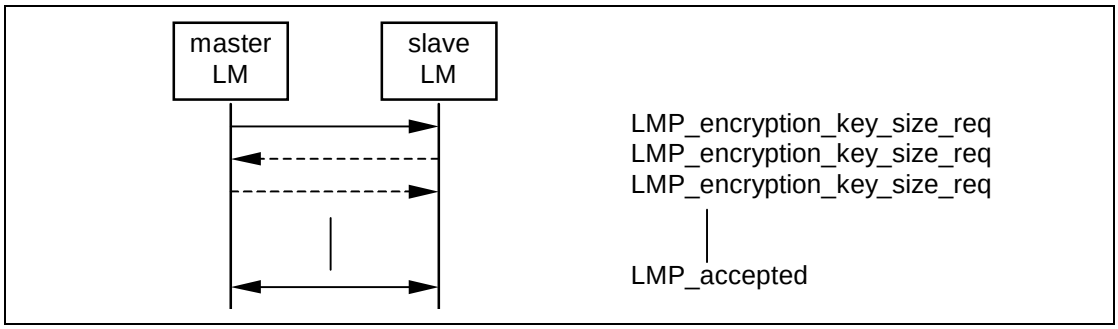


Sequence 12: Negotiation for encryption mode.

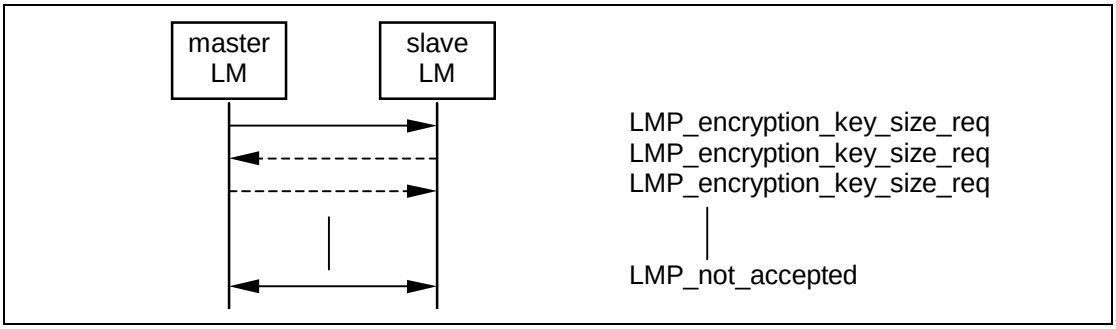
After a unit has sent LMP_encryption_mode_req it is not allowed to send LMP_au_rand before encryption is actually switched on. After a unit has received LMP_encryption_mode_req and sent LMP_accepted it is not allowed to send LMP_au_rand before encryption is actually switched on. If an LMP_au_rand is still sent violating these rules, the claimant shall respond with LMP_not_accepted with the reason code *PDU not allowed*. This is to avoid that the units have different ACOs when they calculate the encryption key. If the encryption mode is not accepted or the encryption key size negotiation results in disagreement the units are allowed to send LMP_au_rand again.

3.6.2 Encryption key size

The next step is to determine the size of the encryption key. In the following we use the same terms as in Baseband Specification Section 14.3.1, on page 159. The master sends LMP_encryption_key_size_req including the suggested key size Lsug, m, which is initially equal to $L_{max, m}$. If $L_{min, s} \leq L_{sug, m}$ and the slave supports $L_{sug, m}$ it responds with LMP_accepted and $L_{sug, m}$ will be used as the key size. If both conditions are not fulfilled the slave sends back LMP_encryption_key_size_req including the slave's suggested key size $L_{sug, s}$. This value is the slave's largest supported key size that is less than $L_{sug, m}$. Then the master performs the corresponding test on the slave's suggestion. This procedure is repeated until a key size agreement is reached or it becomes clear that no such agreement can be reached. If an agreement is reached a unit sends LMP_accepted and the key size in the last LMP_encryption_key_size_req will be used. After this, the encryption is started; see Section 3.6.3 on page 203. If an agreement is not reached a unit sends LMP_not_accepted with the reason code *Unsupported parameter value* and the units are not allowed to communicate using Bluetooth link encryption.".



Sequence 13: Encryption key size negotiation successful.



Sequence 14: Encryption key size negotiation failed.

3.6.3 Start encryption

Finally, encryption is started. The master issues the random number EN_RAND and calculates the encryption key as $K_c=E_3$(current link key, EN_RAND, COF). See Baseband Specification Section 14.2.2.5, on page 155 and 14.2.2.2 for the definition of the COF. The random number must be the same for all slaves if the piconet should support encrypted broadcast. Then the master sends LMP_start_encryption_req, which includes EN_RAND. The slave calculates K_c when this message is received and acknowledges with LMP_accepted.

Sequence 15: Start of encryption.

The start of encryption will be done in three steps:

1. Master is configured to transmit unencrypted packets, but to receive encrypted packets.
2. Slave is configured to transmit and receive encrypted packets.
3. Master is configured to transmit and receive encrypted packets.

Between step 1 and step 2, master-to-slave transmission is possible. This is when LMP_start_encryption_req is transmitted. Step 2 is triggered when the slave receives this message. Between step 2 and step 3, slave-to-master transmission is possible. This is when LMP_accepted is transmitted. Step 3 is triggered when the master receives this message.

3.6.4 Stop encryption

If a unit wants to stop encryption it sends LMP_encryption_mode_req with the parameter encryption mode equal to 0 (no encryption). The other device

responds with LMP_accepted or LMP_not_accepted (the procedure is described in Sequence 12 in section 3.6.1 on page 201). If accepted encryption is stopped by the master sending LMP_stop_encryption_req and the slave responding with LMP_accepted according to Sequence 16.

Sequence 16: Stop of encryption.

Stopping of encryption is then done in three steps, similar to the procedure for starting encryption.

1. Master is configured to transmit encrypted packets, but to receive unencrypted packets.
2. Slave is configured to transmit and receive unencrypted packets.
3. Master is configured to transmit and receive unencrypted packets.

Between step 1 and step 2 master to slave transmission is possible. This is when LMP_stop_encryption_req is transmitted. Step 2 is triggered when the slave receives this message. Between step 2 and step 3 slave to master transmission is possible. This is when LMP_accepted is transmitted. Step 3 is triggered when the master receives this message.

3.6.5 Change encryption mode, key or random number

If the encryption key or encryption random number need to be changed or if the encryption mode needs to be changed between 1 (point-to-point) and 2(point-to-point and broadcast), encryption must first be stopped and then re-started with the new parameters according to the procedures in section 3.6 on page 201, subsections 1-3.

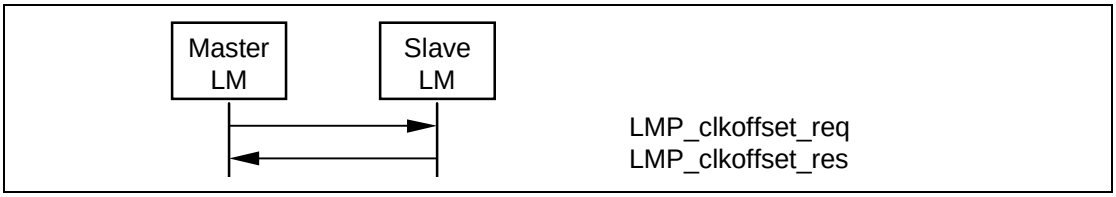
3.7 CLOCK OFFSET REQUEST

When a slave receives the FHS packet, the difference is computed between its own clock and the master's clock included in the payload of the FHS packet. The clock offset is also updated each time a packet is received from the master. The master can request the clock offset at anytime following a successful baseband paging procedure (i.e., before, during or after connection setup).

By saving this clock offset the master knows on what RF channel the slave wakes up to PAGE SCAN after it has left the piconet. This can be used to speed up the paging time the next time the same device is paged.

M/O	PDU	Contents
M	LMP_clkoffset_req	-
M	LMP_clkoffset_res	clock offset

Table 3.7: PDUs used for clock offset request.



Sequence 17: Clock offset requested.

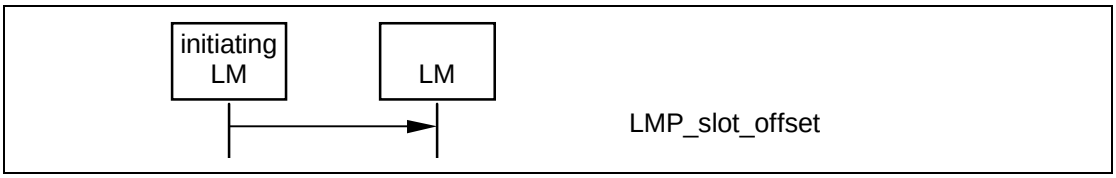
3.8 SLOT OFFSET INFORMATION

With LMP_slot_offset the information about the difference between the slot boundaries in different piconets is transmitted. This PDU carries the parameters slot offset and BD_ADDR. The slot offset is the subtraction of time in µs of the start of the master's TX slot in the piconet where the PDU is transmitted from the time in µs of the start of the master's TX slot in the piconet where the BD_ADDR device is master modulo 1250.

See Section 3.12 on page 208 for the use of LMP_slot_offset in the context of the master-slave switch..

M/O	PDU	Contents
O	LMP_slot_offset	slot offset BD_ADDR

Table 3.8: PDU used for slot offset information.



Sequence 18: Slot offset information is sent.

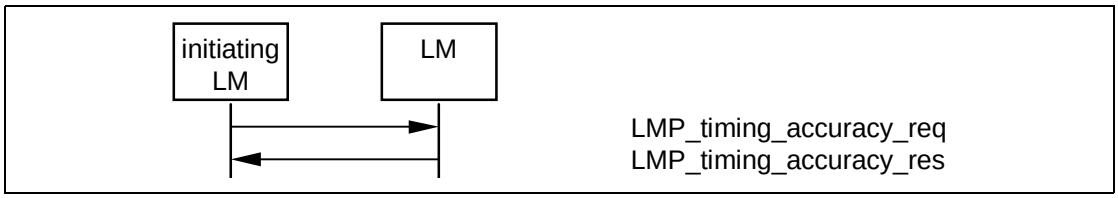
3.9 TIMING ACCURACY INFORMATION REQUEST

LMP supports requests for the timing accuracy. This information can be used to minimize the scan window for a given hold time when returning from hold and to extend the maximum hold time. It can also be used to minimize the scan window when scanning for the sniff mode slots or the park mode beacon packets. The timing accuracy parameters returned are the long term drift measured in ppm and the long term jitter measured in µs of the clock used during hold,

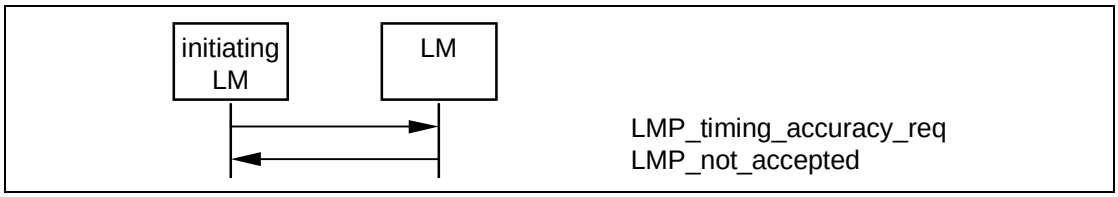
sniff and park mode. These parameters are fixed for a certain device and must be identical when requested several times. Timing accuracy can be requested at anytime following a successful baseband paging procedure, provided this PDU is shown as supported in the supported features list. If the timing accuracy request is not supported, the requesting device must assume worst case values (drift=250ppm and jitter=10µs).

M/O	PDU	Contents
O	LMP_timing_accuracy_req	-
O	LMP_timing_accuracy_res	drift jitter

Table 3.9: PDUs used for requesting timing accuracy information.



Sequence 19: The requested device supports timing accuracy information.



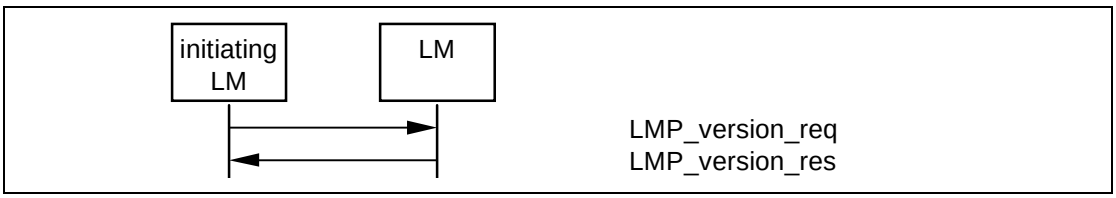
Sequence 20: The requested device does not support timing accuracy information.

3.10 LMP VERSION

LMP supports requests for the version of the LM protocol. The requested device will send a response with three parameters: VersNr, CompId and SubVersNr. VersNr specifies the version of the Bluetooth LMP specification that the device supports. CompId is used to track possible problems with the lower Bluetooth layers. All companies that create a unique implementation of the Link Manager shall have their own CompId. The same company is also responsible for the administration and maintenance of the SubVersNr. It is recommended that each company has a unique SubVersNr for each RF/BB/LM implementation. For a given VersNr and CompId, the values of the SubVersNr must increase each time a new implementation is released. For both CompId and SubVersNr the value 0xFFFF means that no valid number applies. There is no ability to negotiate the version of the LMP. The sequence below is only used to exchange the parameters. LMP version can be requested at anytime following a successful baseband paging procedure.

M/O	PDU	Contents
M	LMP_version_req	VersNr CompId SubVersNr
M	LMP_version_res	VersNr CompId SubVersNr

Table 3.10: PDUs used for LMP version request.



Sequence 21: Request for LMP version.

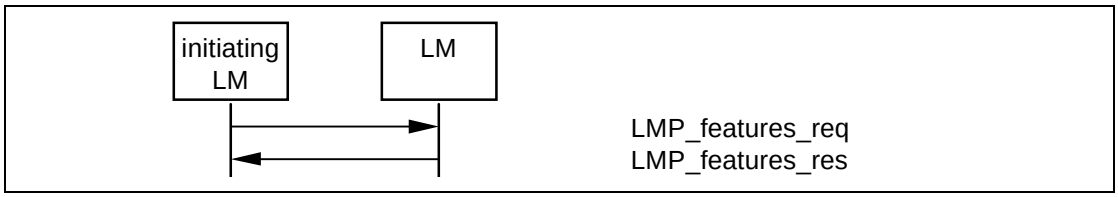
3.11 SUPPORTED FEATURES

The Bluetooth radio and link controller may support only a subset of the packet types and features described in Baseband Specification and Radio Specification. The PDU LMP_features_req and LMP_features_res are used to exchange this information. The supported features can be requested at anytime following a successful baseband paging procedure. A device may not send any packets other than ID, FHS, NULL, POLL, DM1 or DH1 before it is aware of the supported features of the other device. After the features request has been carried out, the intersection of the supported packet types for both sides may also be transmitted. Whenever a request is issued, it must be compatible with the supported features of the other device. For instance, when establishing an SCO link the initiator may not propose to use HV3 packets if

that packet type is not supported by the other device. Exceptions to this rule are LMP_switch_ req and LMP_slot_offset, which can be sent before the requesting side is aware of the other side's features (switch is an optional feature)

M/O	PDU	Contents
M	LMP_features_req	features
M	LMP_features_res	features

Table 3.11: PDUs used for features request.



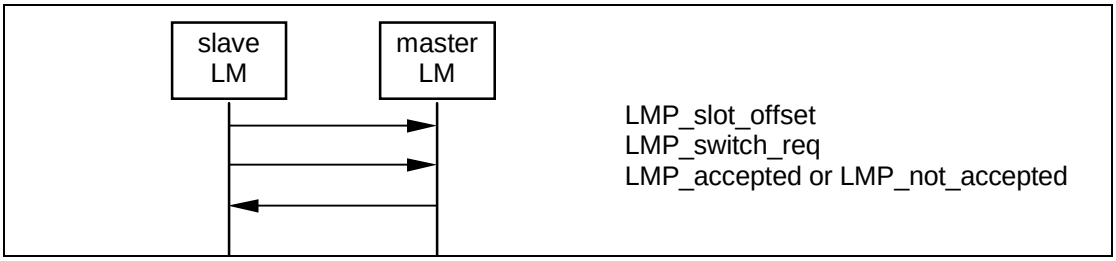
Sequence 22: Request for supported features.

3.12 SWITCH OF MASTER-SLAVE ROLE

Since the paging device always becomes the master of the piconet, a switch of the master-slave role is sometimes needed, see Baseband Specification Section 10.9.3, on page 121.

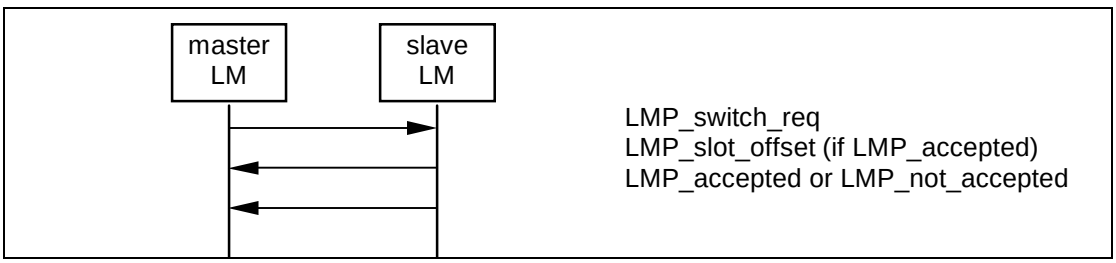
M/O	PDU	Contents
O	LMP_switch_req	switch instant
O	LMP_slot_offset	slot offset BD_ADDR

*PDUs used for master-slave switch.*If the slave initiates the master-slave switch it finalizes the transmission of the current ACL packet with L2CAP information, stops L2CAP transmission and sends LMP_slot_offset immediately followed by LMP_switch_req. If the master accepts the master-slave switch it finalizes the transmission of the current ACL packet with L2CAP information, stops L2CAP transmission and responds with LMP_accepted. When the master-slave switch has been completed on Baseband level (successfully or not) both units re-enable L2CAP transmission. If the master rejects the master-slave switch it responds with LMP_not_accepted and the slave re-enables L2CAP transmission. The transaction ID for all PDUs in the sequence is set to 1.



Sequence 23: Master-slave switch (slave initiated).

If the master initiates the master-slave switch it finalizes the transmission of the current ACL packet with L2CAP information, stops L2CAP transmission and sends LMP_switch_req. If the slave accepts the master-slave switch it finalizes the transmission of the current ACL packet with L2CAP information, stops L2CAP transmission and responds with LMP_slot_offset immediately followed by LMP_accepted. When the master-slave switch has been completed on Baseband level (successfully or not) both units re-enable L2CAP transmission. If the slave rejects the master-slave switch it responds with LMP_not_accepted and the master re-enables L2CAP transmission. The transaction ID for all PDUs in the sequence is set to 0.



Sequence 24: Master-slave switch (master initiated).

The LMP_switch_req PDU contains a parameter, switch instant, which specifies the instant at which the TDD switch is made. This is specified as a Bluetooth clock value of the master's clock, which is available to both devices. This instant is chosen by the sender of the message and should be at least $2*T_{poll}$ or 32 (whichever is greater) slots in the future. The assumption is that the switch instant is always within 12 hours of the current clock value, in order to accommodate clock wrap.

The sender of the LMP_switch_req selects the switch instant and queues the LMP_switch_req to LC for transmission and starts a timer to expire at the switch instant. When the timer expires it initiates the mode switch. In the case of a master initiated switch if the LMP_slot_offset has not been received by the switch instant the master slave switch is carried out without an estimate of the slave's slot offset. If LMP_not_accepted is received before the timer expires then the timer is stopped and the role switch is not initiated.

When the LMP_switch_req is received the switch instant is compared with the current master clock value. If it is in the past then the instant has been passed

and LMP_not_accepted with the reason code Instant passed shall be returned. If it is in the future then LMP_accepted is returned assuming the master-slave switch is allowed and a timer is started to expire at the switch instant. When this timer expires it initiates the mode switch.

Support for LMP_slot_offset is mandatory if LMP_switch_req is supported.

3.13 NAME REQUEST

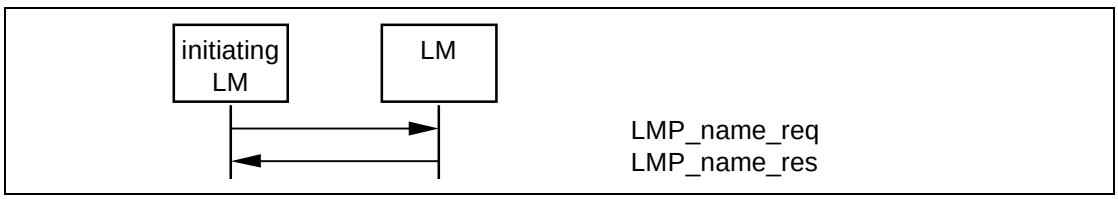
LMP supports name request to another Bluetooth device. The name is a user-friendly name associated with the Bluetooth device and consists of a maximum of 248 bytes coded according to the UTF-8 standard. The name is fragmented over one or more DM1 packets. When the LMP_name_req is sent, a name offset indicates which fragment is expected. The corresponding LMP_name_res carries the same name offset, the name length indicating the total number of bytes in the name of the Bluetooth device and the name fragment, where:

- name fragment(N) = name(N + name offset), if (N + name offset) < name length
- name fragment(N) = 0,otherwise.

Here $0 \leq N \leq 13$. In the first sent LMP_name_req, name offset=0. Sequence 25 is then repeated until the initiator has collected all fragments of the name. The name request can be made at anytime following a successful baseband paging procedure.

M/O	PDU	Contents
M	LMP_name_req	name offset
M	LMP_name_res	name offset name length name fragment

Table 3.12: PDUs used for name request.



Sequence 25: Device's name requested and it responses.

3.14 DETACH

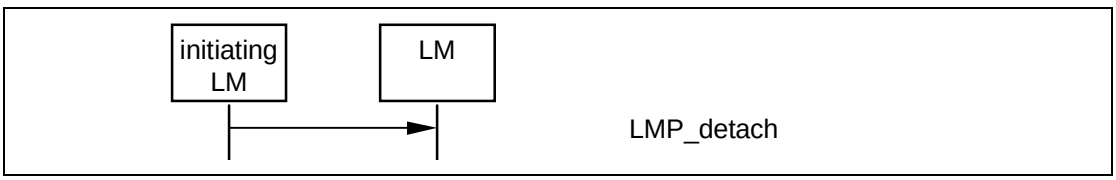
The connection between two Bluetooth devices can be closed anytime by the master or the slave. A reason parameter is included in the message to inform the other party of why the connection is closed.

M/O	PDU	Contents
M	LMP_detach	reason

Table 3.13: PDU used for detach.

The initiating LM first finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission. The initiating LM then queues the LMP_detach for transmission and starts a timer for $6*T_{poll}$ slots where T_{poll} is the poll interval for the connection. If the initiating LM receives the Baseband-level acknowledgement before the timer expires it now starts a timer for $3*T_{poll}$ slots. When this timer expires (and if the initiating LM is the master) the AM_ADDR can be re-used immediately. If the initial timer expires then the initiating LM drops the link and starts a timer for $T_{linksupervisiontimeout}$ slots after which the AM_ADDR can be re-used (if the initiating LM is the master).

When the receiving LM receives the LMP_detach, it starts a timer for $6*T_{poll}$ slots if it is the master and $3*T_{poll}$ if it is the slave. On timer expiration, the link is detached and (if the receiving LM is the master) the AM_ADDR can be re-used immediately. If the receiver never gets the LMP_detach then a link supervision timeout will occur and the link will be detached.



Sequence 26: Connection closed by sending LMP_detach.

3.15 HOLD MODE

The ACL link of a connection between two Bluetooth devices can be placed in hold mode for a specified hold time. During this time no ACL packets will be transmitted from the master. The hold mode is typically entered when there is no need to send data for a relatively long time. The transceiver can then be turned off in order to save power. But the hold mode can also be used if a device wants to discover or be discovered by other Bluetooth devices, or wants to join other piconets. What a device actually does during the hold time is not controlled by the hold message, but it is up to each device to decide.

M/O	PDU	Contents
O	LMP_hold	hold time, hold instant
O	LMP_hold_req	hold time, hold instant

Table 3.14: PDUs used for hold mode.

The LMP_hold and LMP_hold_req PDUs both contain a parameter, hold instant, which specifies the instant at which the hold will become effective. This is specified as a Bluetooth clock value of the master's clock, which is available to both devices. This instant is chosen by the sender of the message and should be at least $6*T_{poll}$ slots in the future. The assumption is that the hold instant is always within 12 hours of the current clock value, in order to accommodate clock wrap.

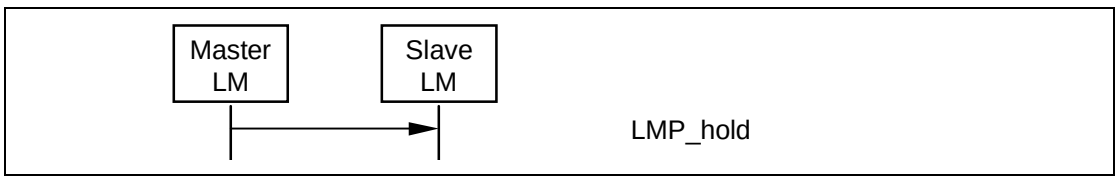
3.15.1 Master forces hold mode

The master can force hold mode if there has previously been a request for hold mode that has been accepted. The hold time included in the PDU when the master forces hold mode cannot be longer than any hold time the slave has previously accepted when there was a request for hold mode.

The master LM first finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission. It selects the hold instant and queues the LMP_hold to its LC for transmission. It then starts a timer to wait until the hold instant occurs. When this timer expires then the connection enters hold mode. Note that the Baseband-level acknowledgement is ignored in this mechanism.

When the slave LM receives the LMP_hold it compares the hold instant with the current master clock value. If it is in the future then it starts a timer to expire at this instant and enters hold mode when it expires.

When the master LM exits from Hold mode it re-enables L2CAP transmission.



Sequence 27: Master forces slave into hold mode.

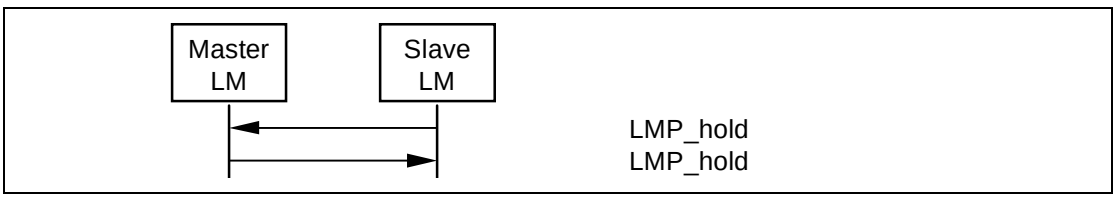
3.15.2 Slave forces hold mode

The slave can force hold mode if there has previously been a request for hold mode that has been accepted. The hold time included in the PDU when the slave forces hold mode cannot be longer than any hold time the master has previously accepted when there was a request for hold mode.

The slave LM first finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission. It selects the hold instant and queues the LMP_hold to its LC for transmission. It then waits for the LMP_hold from the master acting according to the procedure described in Section 3.15.1 .

When the master LM receives the LMP_hold it finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission. It then inspects the hold instant. If this is less than $6*T_{poll}$ slots in the future it should modify the instant so that it is at least $6*T_{poll}$ slots in the future. Then it sends the LMP_hold using the mechanism described in Section 3.15.1 .

When the master and slave LMs exit from Hold mode they re-enable L2CAP transmission.



Sequence 28: Slave forces master into hold mode.

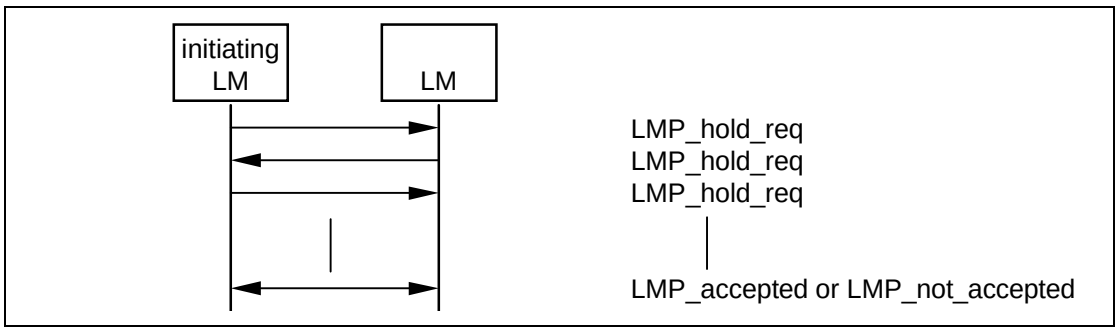
3.15.3 Master or slave requests hold mode

The master or the slave can request to enter hold mode. Upon receipt of the request, the same request with modified parameters can be returned or the negotiation can be terminated. If an agreement is seen LMP_accepted terminates the negotiation and the ACL link is placed in hold mode. If no agreement is seen, LMP_not_accepted with the reason code *unsupported parameter value* terminates the negotiation and hold mode is not entered.

The initiating LM first finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission. On receiving the LMP_hold_req the receiving LM finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission

The LM sending LMP_hold_req selects the hold instant, which should be at least $9*T_{poll}$ slots in the future. If this is a response to a previous LMP_hold_req and the contained hold instant is at least $9*T_{poll}$ slots in the future then this should be used. The LMP_hold_req is then queued to its LC for transmission and a timer is started to expire at this instant and the connection enters hold mode when it expires unless an LMP_not_accepted or LMP_hold_req is received by its LM before that point. If the LM receiving LMP_hold_req agrees to enter hold mode it returns LMP_accepted and starts a timer to expire at the hold instant. When this timer expires it enters hold mode.

When each LM exits from Hold mode it re-enables L2CAP transmission.



Sequence 29: Negotiation for hold mode.

3.16 SNIFF MODE

To enter sniff mode, master and slave negotiate a sniff interval T_{sniff} and a sniff offset, D_{sniff}, which specifies the timing of the sniff slots. The offset determines the time of the first sniff slot; after that the sniff slots follows periodically with the sniff interval T_{sniff}. To avoid problems with a clock wrap-around during the initialization, one of two options is chosen for the calculation of the first sniff slot. A timing control flag in the message from the master indicates this. Note: Only bit1 of the timing control flag is valid.

When the link is in sniff mode the master can only start a transmission in the sniff slot. Two parameters control the listening activity in the slave. The sniff attempt parameter determines for how many slots the slave must listen, beginning at the sniff slot, even if it does not receive a packet with its own AM address. The sniff timeout parameter determines for how many additional slots the slave must listen if it continues to receive only packets with its own AM address.

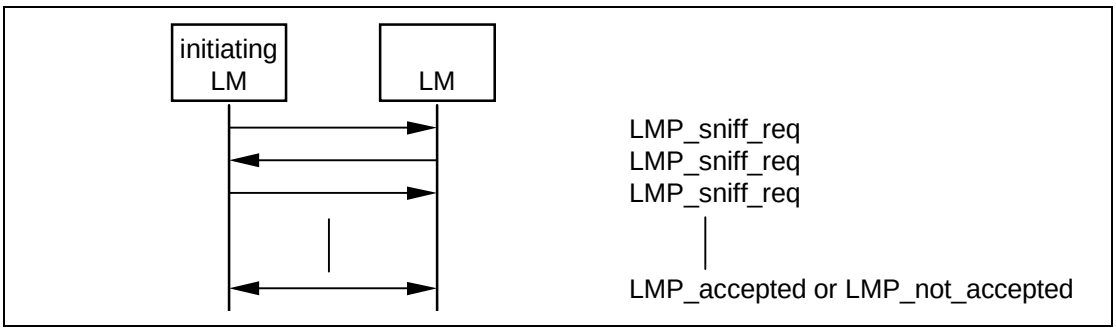
M/O	PDU	Contents
O	LMP_sniff_req	timing control flags D_{sniff} T_{sniff} sniff attempt sniff timeout
O	LMP_unsniff_req	-

Table 3.15: PDUs used for sniff mode.

3.16.1 Master or slave requests sniff mode

The master or the slave can request to enter sniff mode. Upon receipt of the request, the same request with modified parameters can be returned or the negotiation can be terminated. If an agreement is seen LMP_accepted terminates the negotiation and the ACL link is placed in sniff mode. If no agreement

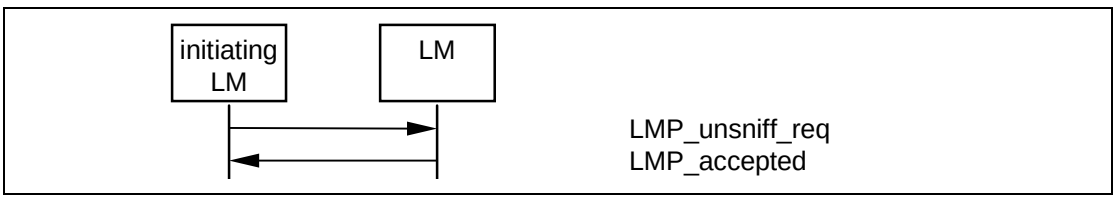
is seen, LMP_not_accepted with the reason code *unsupported parameter value* terminates the negotiation and sniff mode is not entered.



Sequence 30: Negotiation for sniff mode.

3.16.2 Moving a slave from sniff mode to active mode

Sniff mode is ended by sending the PDU LMP_unsniff_req. The requested device must reply with LMP_accepted. If the slave requests it will enter active mode after receiving LMP_accepted. If the master requests, the slave will enter active mode after receiving LMP_unsniff_req.



Sequence 31: Slave moved from sniff mode to active mode.

3.17 PARK MODE

If a slave does not need to participate in the channel, but still should be FH-synchronized, it can be placed in park mode. In this mode the device gives up its AM_ADDR but still re-synchronizes to the channel by waking up at the beacon instants separated by the beacon interval. The beacon interval, a beacon offset and a flag indicating how the first beacon instant is calculated determine the first beacon instant. After this the beacon instants follow periodically at the predetermined beacon interval. At the beacon instant the parked slave can be activated again by the master, the master can change the park mode parameters, transmit broadcast information or let the parked slaves request access to the channel.

All PDUs sent from the master to the parked slaves are broadcast. These PDUs (LMP_set_broadcast_scan_window, LMP_modify_beacon, LMP_unpark_BD_addr_req and LMP_unpark_PM_addr_req) are the only PDUs that can be sent to a slave in park mode and the only PDUs that can be broadcast. To increase reliability for broadcast, the packets are made as short as possible. Therefore the format for these LMP PDUs are somewhat different. The parameters are not always byte-aligned and the length of the PDUs is variable.

The messages for controlling the park mode include many parameters, which are all defined in Baseband Specification Section 10.8.4, on page 112 When a slave is placed in park mode it is assigned a unique PM_ADDR, which can be used by the master to unpark that slave. The all-zero PM_ADDR has a special meaning; it is not a valid PM_ADDR. If a device is assigned this PM_ADDR, it must be identified with its BD_ADDR when it is unparked by the master.

M/O	PDU	Contents
O	LMP_park_req	timing control flags D_B T_B N_B Δ_B PM_ADDR AR_ADDR N_{Bsleep} D_{Bsleep} D_{access} T_{access} $N_{acc\text{-}slots}$ N_{poll} M_{access} access scheme

Table 3.16: PDUs used for park mode.

M/O	PDU	Contents
O	LMP_set_broadcast_scan_window	timing control flags D_B (optional) broadcast scan window
O	LMP_modify_beacon	timing control flags D_B (optional) T_B N_B Δ_B D_{access} T_{access} $N_{acc\text{-}slots}$ N_{poll} M_{access} access scheme
O	LMP_unpark_PM_ADDR_req	timing control flags D_B (optional) AM_ADDR PM_ADDR AM_ADDR (optional) PM_ADDR (optional) (totally 1-7 pairs of AM_ADDR, PM_ADDR)
O	LMP_unpark_BD_ADDR _req	timing control flags D_B (optional) AM_ADDR BD_ADDR AM_ADDR (optional) BD_ADDR (optional)

Table 3.16: PDUs used for park mode.

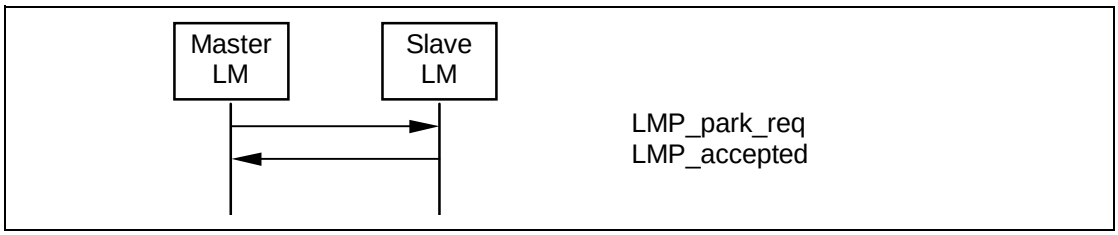
3.17.1 Master requests slave to enter park mode

The master can request park mode. The master finalizes the transmission of the current ACL packet with L2CAP information, stops point-to-point L2CAP transmission and then sends LMP_park_req. If the slave accepts to enter park mode it finalizes the transmission of the current ACL packet with L2CAP information, stops L2CAP transmission and then responds with LMP_accepted.

When the slave queues LMP_accepted it starts a timer for $6*T_{poll}$ slots. If the Baseband-level acknowledgement is received before this timer expires it enters park mode immediately otherwise it enters park mode when the timer expires.

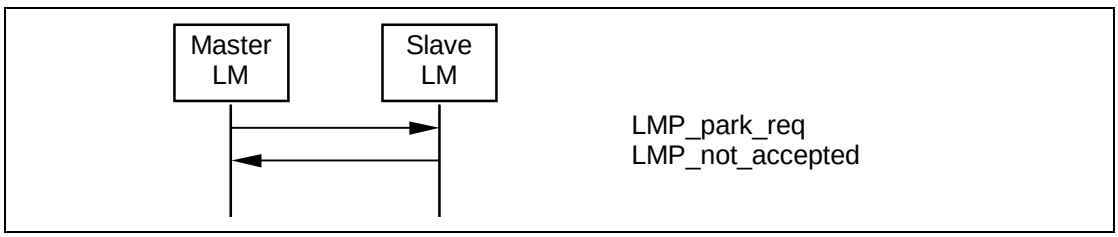
When the master receives LMP_accepted it starts a timer for $6*T_{poll}$ slots. When this timer expires the slave is in park mode and the AM_ADDR can be re-used.

Note: If the master never receives the LMP_accepted then a link supervision timeout will occur.



Sequence 32: Slave accepts to enter park mode.

If the slave rejects to enter park mode it responds with LMP_not_accepted and the master re-enables L2CAP transmission.



Sequence 33: Slave rejects to enter into park mode

3.17.2 Slave requests to enter park mode

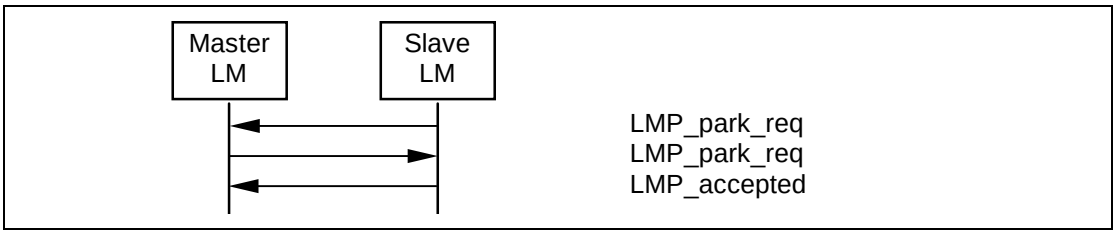
The slave can request park mode. The slave finalizes the transmission of the current ACL packet with L2CAP information, stops L2CAP transmission and then sends LMP_park_req. The parameters PM_ADDR and AR_ADDR are not valid and the other parameters represent suggested values. If the master wants the slave to enter park mode it finalizes the transmission of the current ACL packet with L2CAP information, stops point-to-point L2CAP transmission

and then sends LMP_park_req, where the parameter values may be different from the values in the PDU sent from the slave. If the slave can accept these parameter it responds with LMP_accepted.

When the slave queues LMP_accepted it starts a timer for $6*T_{poll}$ slots. If the Baseband-level acknowledgement is received before this timer expires it enters park mode immediately otherwise it enters park mode when the timer expires.

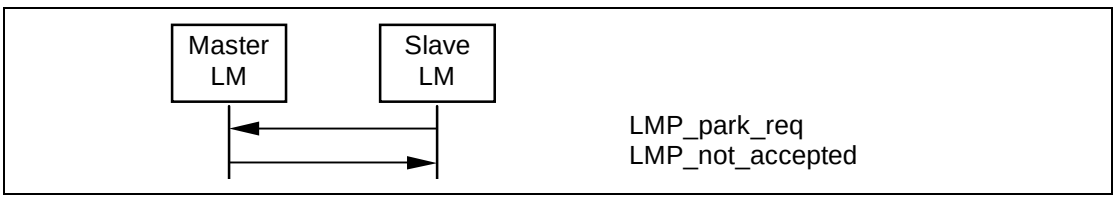
When the master receives LMP_accepted it starts a timer for $6*T_{poll}$ slots. When this timer expires the slave is in park mode and the AM_ADDR can be re-used.

Note: If the master never receives the LMP_accepted then a link supervision timeout will occur.



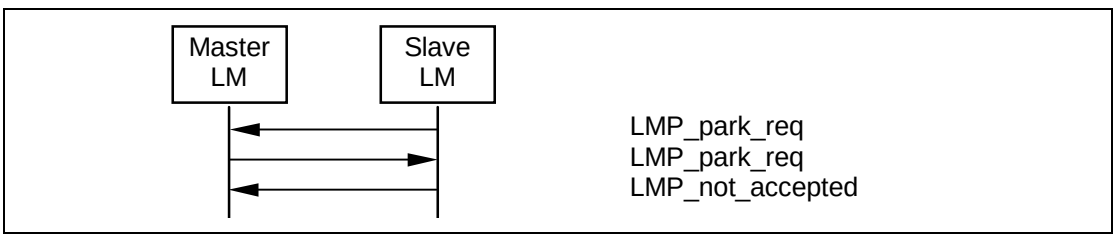
Sequence 34: Slave requests to enter park mode and accepts master's beacon parameters.

If the master does not accept that the slave enters park mode it sends LMP_not_accepted. The slave then re-enables L2CAP transmission.



Sequence 35: Master rejects slave's request to enter park mode

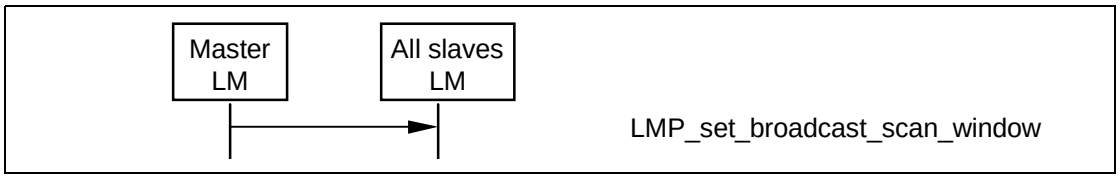
If the slave does not accept the parameters in LMP_park_req sent from the master it responds with LMP_not_accepted and both units re-enable L2CAP transmission.



Sequence 36: Slave requests to enter park mode, but rejects master's beacon parameters.

3.17.3 Master sets up broadcast scan window

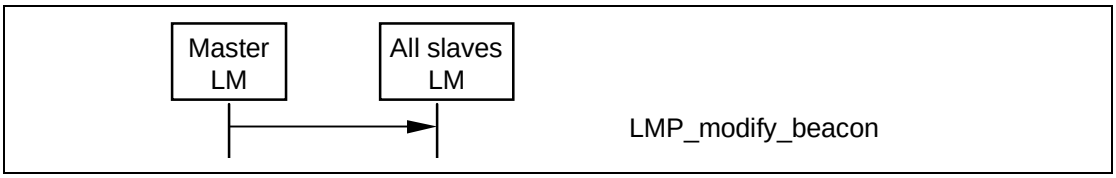
If more broadcast capacity is needed than the beacon train, the master can indicate to the slaves that more broadcast information will follow the beacon train by sending LMP_set_broadcast_scan_window. This message is always sent in a broadcast packet at the beacon slot(s). The scan window starts in the beacon instant and is only valid for the current beacon.



Sequence 37: Master notifies all slaves of increase in broadcast capacity.

3.17.4 Master modifies beacon parameters

When the beacon parameters change the master notifies the parked slaves of this by sending LMP_modify_beacon. This message is always sent in a broadcast packet at the beacon slot(s).



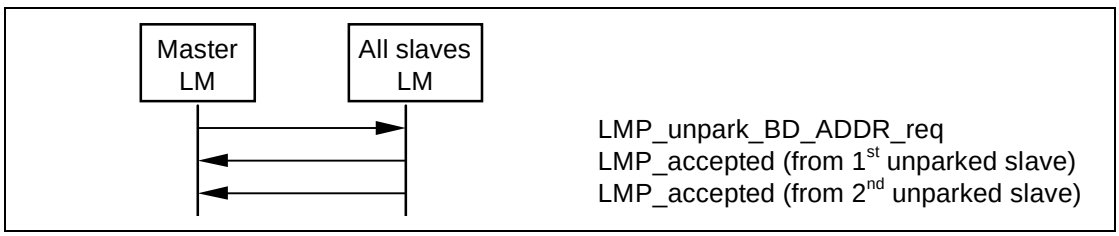
Sequence 38: Master modifies beacon parameters.

3.17.5 Unparking slaves

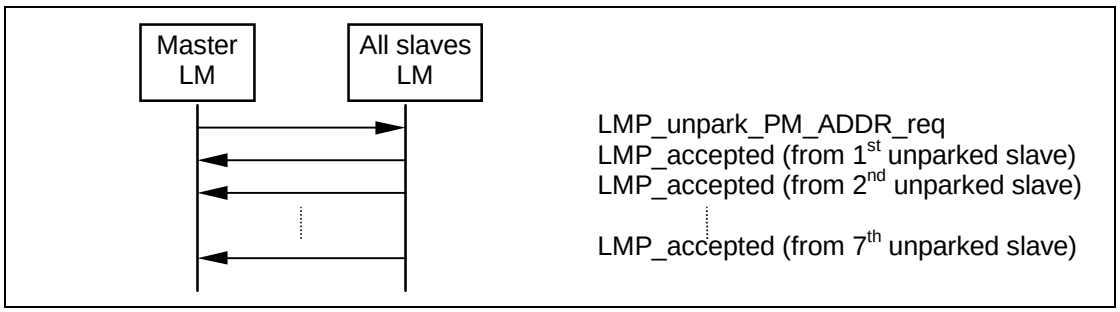
The master can unpark one or many slaves by sending a broadcast LMP message including the PM_ADDR or the BD_ADDR of the device(s) it wishes to unpark at the beacon slot(s). This message also includes the AM_ADDR that the master assigns to the slave(s). After sending this message, the master must check the success of the unpark by polling each unparked slave, i.e. sending POLL packets, so that the slave is granted access to the channel. The unparked slave must then send a response with LMP_accepted. If this message is not received from the slave within a certain time after the master sent the unpark message, the unpark failed and the master must consider the slave as still being in park mode.

One message is used where the parked device is identified with the PM_ADDR, and another message is used where it is identified with the BD_ADDR. Both messages have variable length depending on the number of slaves the master unparks. For each slave the master wishes to unpark an AM_ADDR followed by the PM/BD_ADDR of the device that is assigned this AM_ADDR is included in the payload. If the slaves are identified with the PM_ADDR a maximum of 7 slaves can be unparked with the same message. If they are identified with the BD_ADDR a maximum of 2 slaves can be unparked with the same message.

After a successful unparking, both units re-enable L2CAP transmission.



Sequence 39: Master unparks slaves addressed with their BD_ADDR.



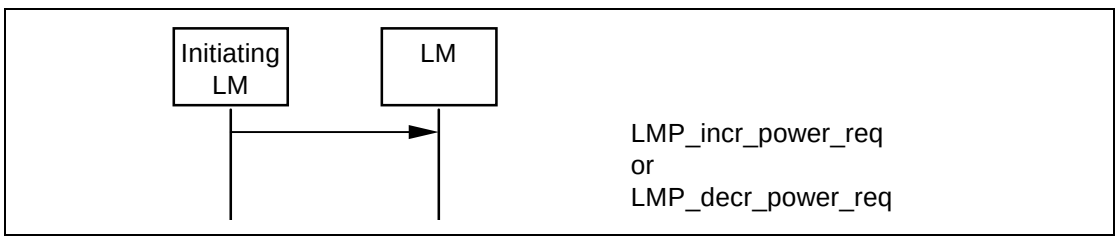
Sequence 40: Master unparks slaves addressed with their PM_ADDR.

3.18 POWER CONTROL

If the RSSI value differs too much from the preferred value of a Bluetooth device, it can request an increase or a decrease of the other device's TX power. The power adjustment requests can be made at anytime following a successful baseband paging procedure. If a device does not support power control requests this is indicated in the supported features list and thus no power control requests shall be sent after the supported features response has been processed. Prior to this time, a power control adjustment might be sent and if the recipient does not support power control it is allowed to send LMP_max_power in response to LMP_incr_power_req and LMP_min_power in response to LMP_decr_power_req. Another possibility is to send LMP_not_accepted with the reason unsupported LMP feature. Upon receipt of this message, the output power is increased or decreased one step. See Radio Specification Section 3.1, on page 22 for the definition of the step size. At the master side the TX power is completely independent for different slaves; a request from one slave can only effect the master's TX power for that same slave.

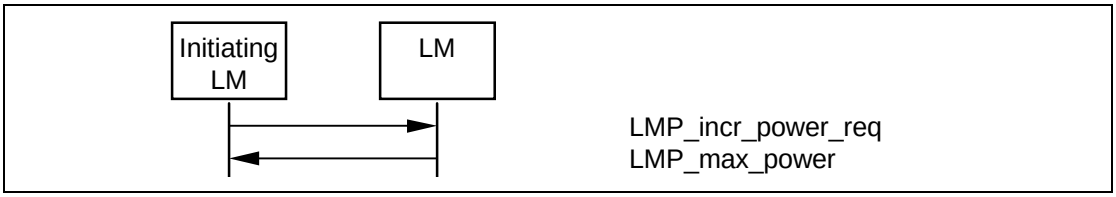
M/O	PDU	Contents
O	LMP_incr_power_req	for future use (1 Byte)
O	LMP_decr_power_req	for future use (1 Byte)
O	LMP_max_power	-
O	LMP_min_power	-

Table 3.17: PDUs used for power control.

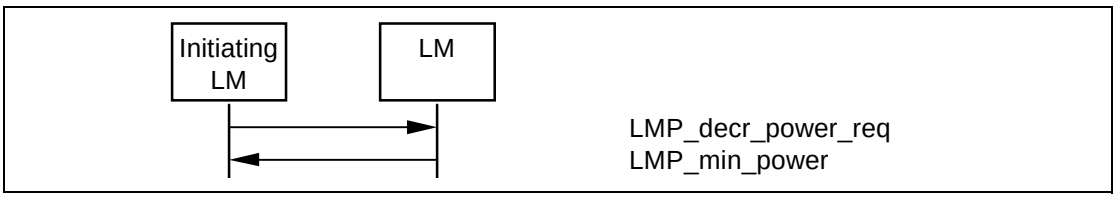


Sequence 41: A device requests a change of the other device's TX power.

If the receiver of LMP_incr_power_req already transmits at maximum power LMP_max_power is returned. The device may then only request an increase again after having requested a decrease at least once. Similarly, if the receiver of LMP_decr_power_req already transmits at minimum power then LMP_min_power is returned and the device may only request a decrease again after having requested an increase at least once.



Sequence 42: The TX power cannot be increased.



Sequence 43: The TX power cannot be decreased.

One byte is reserved in LMP_incr/decr_power_req for future use. It could, for example, be the mismatch between preferred and measured RSSI. The receiver of LMP_incr/decr_power_req could then use this value to adjust to the correct power at once, instead of only changing it one step for each request. The parameter value must be 0x00 for all versions of LMP where this parameter is not yet defined.

3.19 CHANNEL QUALITY-DRIVEN CHANGE BETWEEN DM AND DH

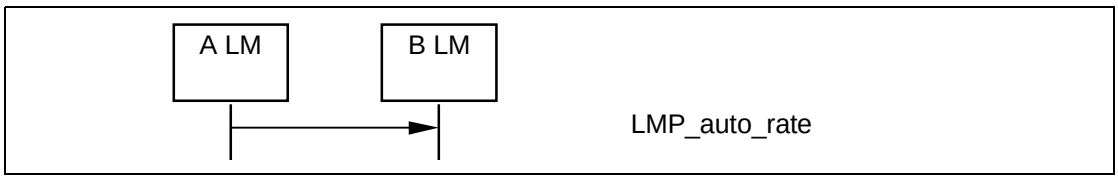
The data throughput for a given packet type depends on the quality of the RF channel. Quality measurements in the receiver of one device can be used to dynamically control the packet type transmitted from the remote device for optimization of the data throughput. If a device A wants the remote device B to have this control it sends LMP_auto_rate once. The device B can then send back LMP_preferred_rate to device A whenever it wishes to change the packet type that A transmits. This PDU has a parameter which determines the preferred coding (with or without 2/3FEC) and the preferred size (in slots) of the packets. Device A is not required to change to the packet type specified by this parameter and may never send a packet that is larger than the maximum allowed number of slots even if the preferred size is greater than this value.

These PDUs can be sent at anytime after connection setup is completed. These PDUs shall not be sent to a given Bluetooth device if the supported fea-

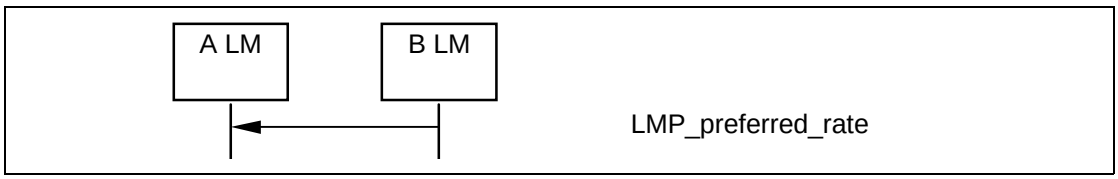
tures list of the device shows that it does not support the channel quality driven change requests.

M/O	PDU	Contents
O	LMP_auto_rate	-
O	LMP_preferred_rate	data rate

Table 3.18: PDUs used for quality driven change of the data rate.



Sequence 44: The left-hand unit is configured to automatically change between DM and DH.



Sequence 45: A wants B to control A's packet type.

3.20 QUALITY OF SERVICE (QoS)

The Link Manager provides Quality of Service capabilities. A poll interval, which is defined as the maximum time between subsequent transmissions from the master to a particular slave on the ACL link, is used to support bandwidth allocation and latency control. The poll interval is guaranteed in the active mode except when there are collisions with page, page scan, inquiry and inquiry scan. The poll interval is also known as T_{poll}. These PDUs can be sent at anytime after connection setup is completed.

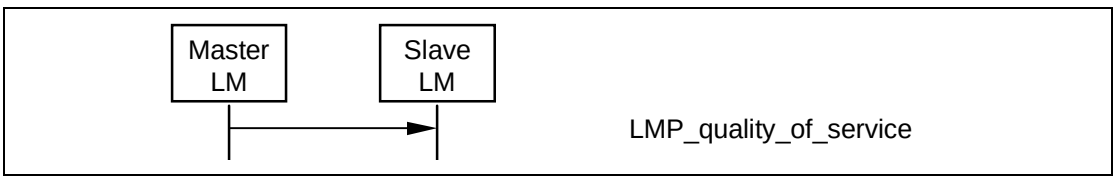
In addition, master and slave negotiate the number of repetitions for broadcast packets (N_{BC}), see Baseband Specification Section 5.3.5, on page 71.

M/O	PDU	Contents
M	LMP_quality_of_service	poll interval N_{BC}
M	LMP_quality_of_service_req	poll interval N_{BC}

Table 3.19: PDUs used for quality of service.

3.20.1 Master notifies slave of the quality of service

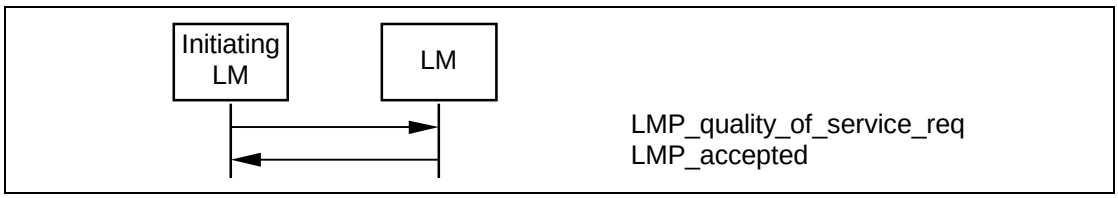
In this case the master notifies the slave of the new poll interval and N_{BC}. The slave cannot reject the notification.



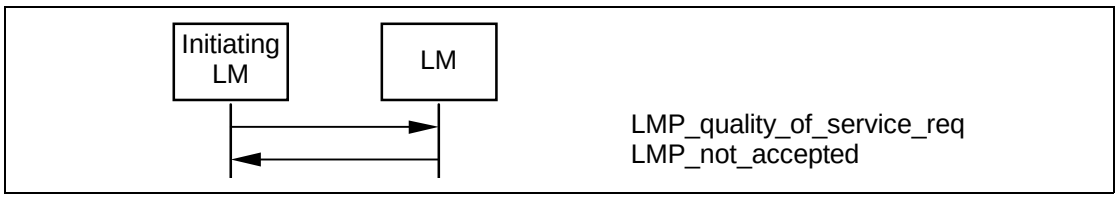
Sequence 46: B changes A's packet type.

3.20.2 Device requests new quality of service

In this case the master or slave requests a new poll interval and N_{BC}.
The parameter N_{BC} is meaningful only when it is sent by a master to a slave. For transmission of LMP_quality_of_service_req PDUs from a slave, this parameter is ignored by the master. The request can be accepted or rejected. This will allow the master and slave to dynamically negotiate the quality of service as needed.



Sequence 47: Device accepts new quality of service



Sequence 48: Device rejects new quality of service.

3.21 SCO LINKS

When a connection has been established between two Bluetooth devices the connection consists of an ACL link. One or more SCO links can then be established. The SCO link reserves slots separated by the SCO interval, T_{sco}. The first slot reserved for the SCO link is defined by T_{sco} and the SCO offset, D_{sco}. After that the SCO slots follows periodically with the SCO interval. To avoid problems with a wrap-around of the clock during initialization of the SCO link, a flag indicating how the first SCO slot should be calculated is included in a message from the master. Note: Only bit0 and bit1 of this field is valid. Each SCO link is distinguished from all other SCO links by an SCO handle. The SCO handle zero is never used.

M/O	PDU	Contents
O	LMP_SCO_link_req	SCO handle timing control flags D_{sco} T_{sco} SCO packet air mode
O	LMP_remove_SCO_link_req	SCO handle reason

Table 3.20: PDUs used for managing the SCO links.

3.21.1 Master initiates an SCO link

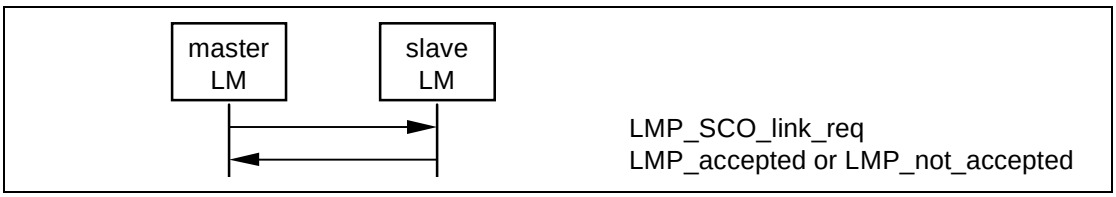
When establishing an SCO link the master sends a request with parameters that specify the timing, packet type and coding that will be used on the SCO link. For each of the SCO packets Bluetooth supports three different voice coding formats on the air-interface: µ-law log PCM, A-law log PCM and CVSD. The air coding by log PCM or CVSD can be deactivated to achieve a transparent synchronous data link at 64 kbits/s.

The slots used for the SCO links are determined by three parameters controlled by the master: T_{sco}, D_{sco} and a flag indicating how the first SCO slot should be calculated. After the first slot, the SCO slots follows periodically with the T_{sco}.

If the slave does not accept the SCO link, but is willing to consider another possible set of SCO parameters, it can indicate what it does not accept in the error reason field of LMP_not_accepted. The master then has the possibility to issue a new request with modified parameters.

The SCO handle in the message must be different from any already existing SCO link(s).

Note: If the SCO packet type is HV1 the LMP_accepted must be sent using the DM1 packet.

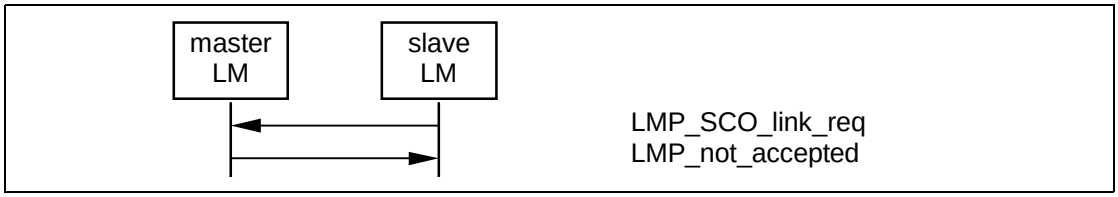


Sequence 49: Master requests an SCO link.

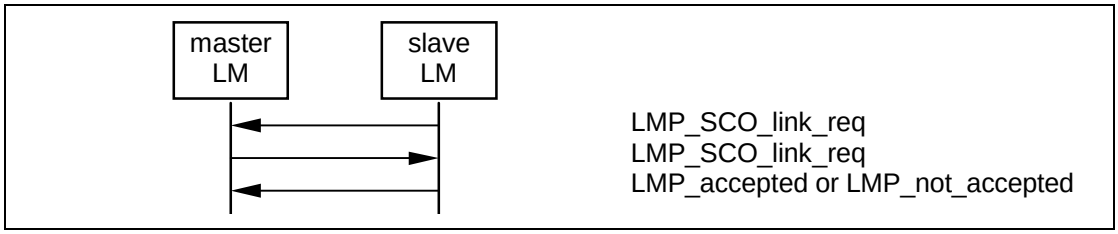
3.21.2 Slave initiates an SCO link

The slave can also initiate the establishment of an SCO link. The slave sends LMP_SCO_link_req, but the parameters timing control flags and D_{sco} are invalid as well as the SCO handle, which must be zero. If the master is not capable of establishing an SCO link, it replies with LMP_not_accepted. Otherwise it sends back LMP_SCO_link_req. This message includes the assigned SCO handle, D_{sco} and the timing control flags. For the other parameters, the master should try to use the same parameters as in the slave request; if the master cannot meet that request, it is allowed to use other values. The slave must then reply with LMP_accepted or LMP_not_accepted.

"Note: If the SCO packet type is HV1 the LMP_accepted must be sent using the DM1 packet.



Sequence 50: Master rejects slave's request for an SCO link.



Sequence 51: Master accepts slave's request for an SCO link.

3.21.3 Master requests change of SCO parameters

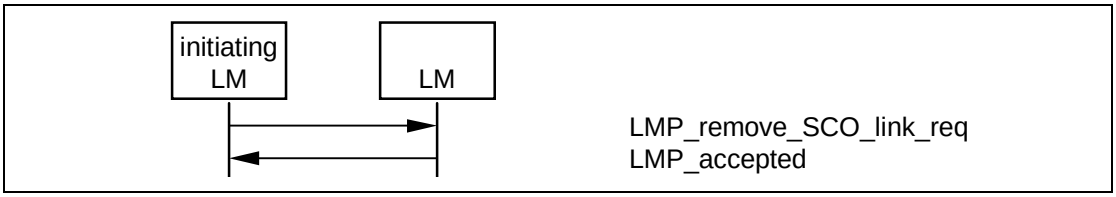
The master sends LMP_SCO_link_req, where the SCO handle is the handle of the SCO link the master wishes to change parameters for. If the slave accepts the new parameters, it replies with LMP_accepted and the SCO link will change to the new parameters. If the slave does not accept the new parameters, it replies with LMP_not_accepted and the SCO link is left unchanged. When the slave replies with LMP_not_accepted it shall indicate in the error reason parameter what it does not accept. The master can then try to change the SCO link again with modified parameters. The sequence is the same as in Section 3.21.1 on page 227.

3.21.4 Slave requests change of SCO parameters

The slave sends LMP_SCO_link_req, where the SCO handle is the handle of the SCO link the slave wishes to change parameters for. The parameters timing control flags and D_{sco} are not valid in this message. If the master does not accept the new parameters it replies with LMP_not_accepted and the SCO link is left unchanged. If the master accepts the new parameters it replies with LMP_SCO_link_req, where it must use the same parameters as in the slave request. When receiving this message the slave replies with LMP_not_accepted if it does not accept the new parameters. The SCO link is then left unchanged. If the slave accepts the new parameters it replies with LMP_accepted and the SCO link will change to the new parameters. The sequence is the same as in Section 3.21.2 on page 227.

3.21.5 Remove an SCO link

Master or slave can remove the SCO link by sending a request including the SCO handle of the SCO link to be removed and a reason indicating why the SCO link is removed. The receiving party must respond with LMP_accepted.



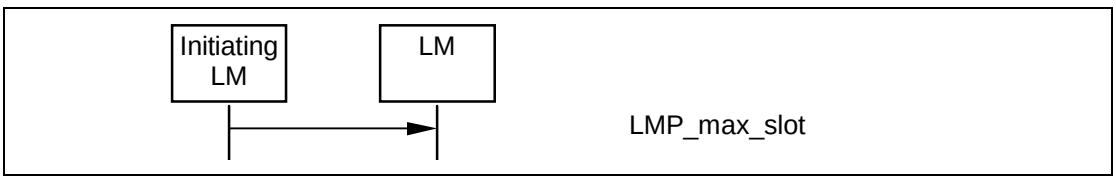
Sequence 52: SCO link removed.

3.22 CONTROL OF MULTI-SLOT PACKETS

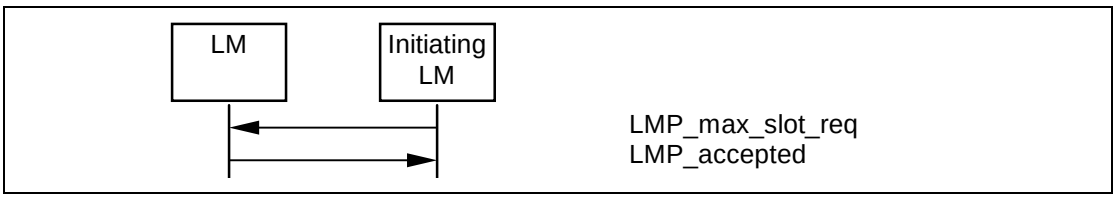
The number of slots used by a device can be limited. A device allows the remote device to use a maximal number of slots by sending the PDU LMP_max_slot providing max slots as parameter. Each device can request to use a maximal number of slots by sending the PDU LMP_max_slot_req providing max slots as parameter. After a new connection, as a result of page, page scan, master-slave switch or unpark, the default value is 1 slot. Two PDUs are used for the control of multi-slot packets.These PDUs can be sent at anytime after connection setup is completed.

M/O	PDU	Contents
M	LMP_max_slot	max slots
M	LMP_max_slot_req	max slots

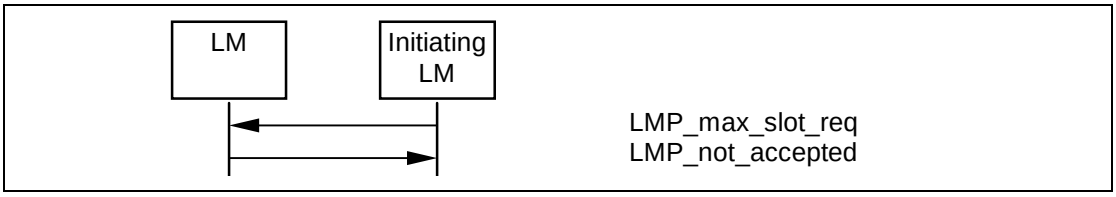
Table 3.21: PDUs used to control the use of multi-slot packets.



Sequence 53: Device allows Remote Device to use a maximum number of slots.



Sequence 54: Device requests a maximum number of slots. Remote Device accepts.



Sequence 55: Device requests a maximum number of slots. Remote Device rejects.

3.23 PAGING SCHEME

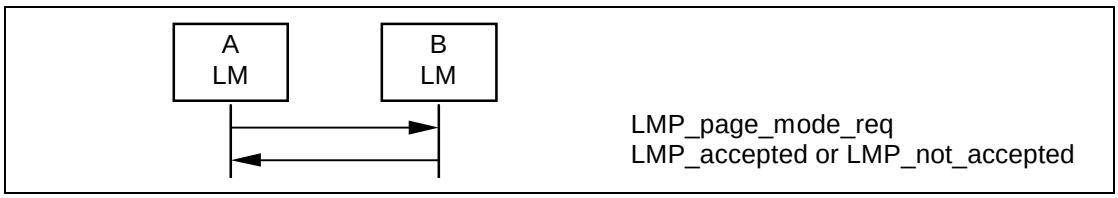
In addition to the mandatory paging scheme, Bluetooth defines optional paging schemes; see “Appendix VII” on page 1025. LMP provides a means to negotiate the paging scheme, which is to be used the next time a unit is paged.

M/O	PDU	Contents
O	LMP_page_mode_req	paging scheme paging scheme settings
O	LMP_page_scan_mode_req	paging scheme paging scheme settings

Table 3.22: PDUs used to request paging scheme.

3.23.1 Page mode

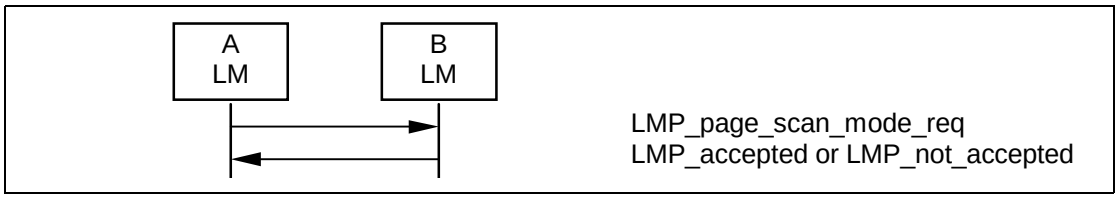
This procedure is initiated from device A and negotiates the paging scheme used when device A pages device B. Device A proposes a paging scheme including the parameters for this scheme and device B can accept or reject. On rejection the old setting is not changed. A request to switch back to the mandatory scheme may be rejected.



Sequence 56: Negotiation for page mode.

3.23.2 Page scan mode

This procedure is initiated from device A and negotiates the paging scheme used when device B pages device A. Device A proposes a paging scheme including the parameters for this scheme and device B can accept or reject. On rejection the old setting is not changed. A request to switch to the mandatory scheme must be accepted.



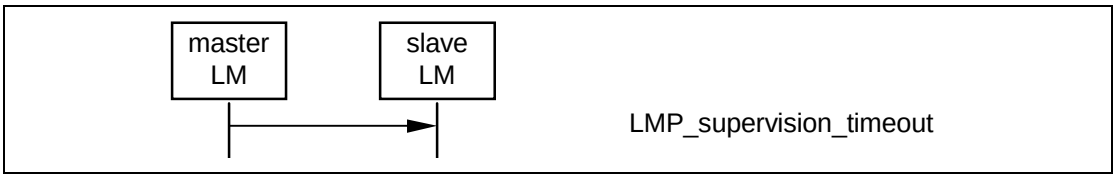
Sequence 57: Negotiation for page scan mode

3.24 LINK SUPERVISION

Each Bluetooth link has a timer that is used for link supervision. This timer is used to detect link loss caused by devices moving out of range, a device's power-down, or other similar failure cases. The scheme for link supervision is described in Baseband Specification Section 10.11, on page 124. An LMP procedure is used to set the value of the supervision timeout.

M/O	PDU	Contents
M	LMP_supervision_timeout	supervision timeout

Table 3.23: PDU used to set the supervision timeout.



Sequence 58: Setting the link supervision timeout.

4 CONNECTION ESTABLISHMENT

After the paging procedure, the master must poll the slave with a max poll interval as defined in Table 5.5 on page 245. LMP procedures with for clock offset request, LMP version, supported features, name request and detach can then be carried out.

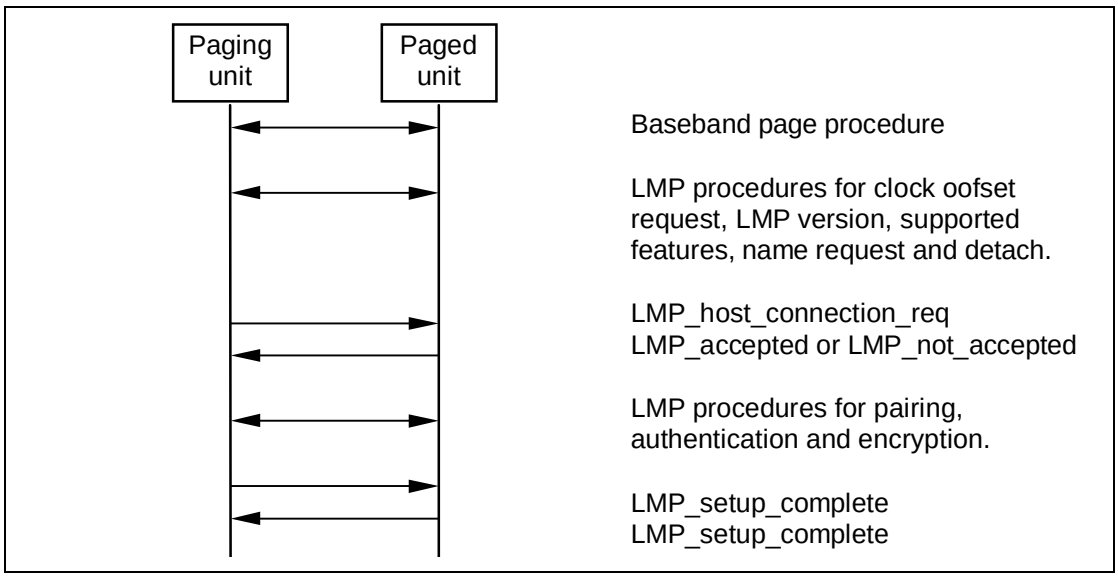


Figure 4.1: Connection establishment.

When the paging device wishes to create a connection involving layers above LM, it sends LMP_host_connection_req. When the other side receives this message, the host is informed about the incoming connection. The remote device can accept or reject the connection request by sending LMP_accepted or LMP_not_accepted. Alternatively, if the slave needs a master-slave switch, see section 3.12 on page 208, it sends LMP_slot_offset and LMP_switch_req after it has received LMP_host_connection_req. When the master-slave switch has been successfully completed, the old slave will reply with LMP_accepted or LMP_not_accpted to LMP_host_connection_req (with the transaction ID set to 0).

If LMP_host_connection_req is accepted, LMP security procedures (pairing, authentication and encryption) can be invoked. When a device is not going to initiate any more security procedures during connection establishment it sends LMP_setup_complete. When both devices have sent LMP_setup_complete the first packet on a logical channel different from LMP can then be transmitted.

M/O	PDU	Contents
M	LMP_host_connection_req	-
M	LMP_setup_complete	-

Table 4.1: PDUs used for connection establishment.

Note: the transaction ID shall be 0 if LMP_setup_complete is sent from the master and 1 if it is sent from the slave.

5 SUMMARY OF PDUs

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_accepted	2	3	DM1/ DV	m ↔ s	op code	2
LMP_au_rand	17	11	DM1	m ↔ s	random number	2-17
LMP_auto_rate	1	35	DM1/ DV	m ↔ s	-	
LMP_clkoffset_req	1	5	DM1/ DV	m → s	-	
LMP_clkoffset_res	3	6	DM1/ DV	m ← s	clock offset	2–3
LMP_comb_key	17	9	DM1	m ↔ s	random number	2-17
LMP_decr_power_req	2	32	DM1/ DV	m ↔ s	for future use	2
LMP_detach	2	7	DM1/ DV	m ↔ s	reason	2
LMP_encryption_key_size_req	2	16	DM1/ DV	m ↔ s	key size	2
LMP_encryption_mode_req	2	15	DM1/ DV	m ↔ s	encryption mode	2
LMP_features_req	9	39	DM1/ DV	m ↔ s	features	2-9
LMP_features_res	9	40	DM1/ DV	m ↔ s	features	2-9
LMP_host_connection_req	1	51	DM1/ DV	m ↔ s	-	
LMP_hold	7	20	DM1/ DV	m ↔ s	hold time, hold instant	4-7
LMP_hold_req	7	21	DM1/ DV	m ↔ s	hold time, hold instant	4-7
LMP_incr_power_req	2	31	DM1/ DV	m ↔ s	for future use	2
LMP_in_rand	17	8	DM1	m ↔ s	random number	2-17
LMP_max_power	1	33	DM1/ DV	m ↔ s	-	

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_max_slot	2	45	DM1/ DV	m ↔ s	max slots	2
LMP_max_slot_req	2	46	DM1/ DV	m ↔ s	max slots	2
LMP_min_power	1	34	DM1/ DV	m ↔ s	-	
LMP_modify_beacon	11 or 13	28	DM1	m → s	timing control flags	2
					D_B	3-4
					T_B	5-6
					N_B	7
					Δ_B	8
					D_{access}	9
					T_{access}	10
					$N_{acc\text{-}slots}$	11
					N_{poll}	12
					M_{access}	13:0-3
					access scheme	13:4-7
LMP_name_req	2	1	DM1/ DV	m ↔ s	name offset	2
LMP_name_res	17	2	DM1	m ↔ s	name offset	2
					name length	3
					name fragment	4-17
LMP_not_accepted	3	4	DM1/ DV	m ↔ s	op code	2
					reason	3
LMP_page_mode_req	3	53	DM1/ DV	m ↔ s	paging scheme	2
					paging scheme settings	3
LMP_page_scan_mode_ req	3	54	DM1/ DV	m ↔ s	paging scheme	2
					paging scheme settings	3

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_park_req	17	25	DM	m → s	timing control flags	2
					D_B	3-4
					T_B	5-6
					N_B	7
					Δ_B	8
					PM_ADDR	9
					AR_ADDR	10
					N_{Bsleep}	11
					D_{Bsleep}	12
					D_{access}	13
					T_{access}	14
					$N_{acc-slots}$	15
					N_{poll}	16
					M_{access}	17:0-3
					access scheme	17:4-7
LMP_preferred_rate	2	36	DM1/ DV	m ↔ s	data rate	2
LMP_quality_of_service	4	41	DM1/ DV	m → s	poll interval	2-3
					N_{BC}	4
LMP_quality_of_service_ req	4	42	DM1/ DV	m ↔ s	poll interval	2-3
					N_{BC}	4
LMP_remove_SCO_link_ req	3	44	DM1/ DV	m ↔ s	SCO handle	2
					reason	3
LMP_SCO_link_req	7	43	DM1/ DV	m ↔ s	SCO handle	2
					timing control flags	3
					D_{sco}	4
					T_{sco}	5
					SCO packet	6
					air mode	7

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_set_broadcast_ scan_window	4 or 6	27	DM1	m → s	timing control flags	2
					D_B	3-4
					broadcast scan window	5-6
LMP_setup_complete	1	49	DM1	m ↔ s	-	
LMP_slot_offset	9	52	DM1/ DV	m ↔ s	slot offset	2-3
					BD_ADDR	4-9
LMP_sniff_req	10	23	DM1	m ↔ s	timing control flags	2
					D_{sniff}	3-4
					T_{sniff}	5-6
					sniff attempt	7-8
					sniff timeout	9-10
LMP_sres	5	12	DM1/ DV	m ↔ s	authentication response	2-5
LMP_start_encryption_req	17	17	DM1	m → s	random number	2-17
LMP_stop_encryption_req .	1	18	DM1/ DV	m → s	-	
LMP_supervision_timeou t	3	55	DM1/ DV	m →s	supervision timeout	2-3
LMP_switch_req	5	19	DM1/ DV	m ↔ s	switch instant	2-5
LMP_temp_rand	17	13	DM1	m → s	random number	2-17
LMP_temp_key	17	14	DM1	m → s	key	2-17
LMP_timing_accuracy_req	1	47	DM1/ DV	m ↔ s	-	
LMP_timing_accuracy_res	3	48	DM1/ DV	m ↔ s	drift	2
					jitter	3
LMP_unit_key	17	10	DM1	m ↔ s	key	2-17

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_unpark_BD_ADDR_req	variable	29	DM1	m → s	timing control flags	2
					D_B	3-4
					AM_ADDR 1st unpark	5:0-2
					AM_ADDR 2nd unpark	5:4-6
					BD_ADDR 1st unpark	6-11
					BD_ADDR 2nd unpark	12-17
LMP_unpark_PM_ADDR_req	variable	30	DM1	m → s	timing control flags	2
					D_B	3-4
					AM_ADDR 1st unpark	5:0-3
					AM_ADDR 2nd unpark	5:4-7
					PM_ADDR 1st unpark	6
					PM_ADDR 2nd unpark	7
LMP_unsniff_req	1	24	DM1/ DV	m ↔ s	-	
LMP_use_semi_ permanent_key	1	50	DM1/ DV	m → s	-	
LMP_version_req	6	37	DM1/ DV	m ↔ s	VersNr	2
					CompId	3-4
					SubVersNr	5-6
LMP_version_res	6	38	DM1/ DV	m ↔ s	VersNr	2
					CompId	3-4
					SubVersNr	5-6

Table 5.1: Coding of the different LM PDUs.

Note1: For LMP_set_broadcast_scan_window, LMP_modify_beacon, LMP_unpark_BD_ADDR_req and LMP_unpark_PM_ADDR_req the parameter D_B is optional. This parameter is only present if bit0 of *timing control flags* is 1. If the parameter is not included, the position in payload for all parameters following D_B are decreased by 2.

Note2: For LMP_unpark_BD_ADDR the AM_ADDR and the BD_ADDR of the 2nd unparked slave are optional. If only one slave is unparked *AM_ADDR 2nd unpark* should be zero and *BD_ADDR 2nd unpark* is left out.

Note3: For LMP_unpark_PM_ADDR the AM_ADDR and the PM_ADDR of the 2nd – 7th unparked slaves are optional. If N slaves are unparked, the fields up to and including the Nth unparked slave are present. If N is odd, the *AM_ADDR (N+1)th unpark* must be zero. The length of the message is x + 3N/2 if N is even and x + 3(N+1)/2 -1 if N is odd, where x = 2 or 4 depending on if the D_B is incluDed Or Not (See Note1).

5.1 DESCRIPTION OF PARAMETERS

Name	Length (bytes)	Type	Unit	Detailed
access scheme	1	u_int4		0: polling technique 1-15: Reserved
air mode	1	u_int8		0: μ-law log 1: A-law log 2: CVSD 3: transparent data 4-255: Reserved
AM_ADDR	1	u_int4		
AR_ADDR	1	u_int8		
authentication response	4	multiple bytes		
BD_ADDR	6	multiple bytes		
broadcast scan window	2	u_int16	slots	
clock offset	2	u_int16	1.25ms	($CLKN_{16-2}$ slave - $CLKN_{16-2}$ master) mod 2^{15} MSbit of second byte not used.
CompId	2	u_int16		see Bluetooth Assigned Numbers (http://www.bluetooth.org/assigned-numbers.htm)
D_{access}	1	u_int8	slots	
D_B	2	u_int16	slots	
D_{Bsleep}	1	u_int8	slots	

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed
data rate	1	u_int8		bit0 = 0: use FEC bit0 = 1: do not use FEC bit1-2=0: No packet-size preference available bit1-2=1: use 1-slot packets bit1-2=2: use 3-slot packets bit1-2=3: use 5-slot packets bit3-7: Reserved
drift	1	u_int8	ppm	
D_{sco}	1	u_int8	slots	
D_{sniff}	2	u_int16	slots	
encryption mode	1	u_int8		0: no encryption 1: point-to-point encryption 2: point-to-point and broadcast encryption 3 -255: Reserved
features	8	multiple bytes		See Table 5.3 on page 243
hold instant	4	u_int32	slots	Bits 27:1 of the master Bluetooth clock value
hold time	2	u_int16	slots	
jitter	1	u_int8	µs	
key	16	multiple bytes		
key size	1	u_int8	byte	
M_{access}	1	u_int4		number of access windows
max slots	1	u_int8	slots	
$N_{acc-slots}$	1	u_int8	slots	
name fragment	14	multiple bytes		UTF-8 characters.
name length	1	u_int8	bytes	
name offset	1	u_int8	bytes	
N_B	1	u_int8		

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed
N_{BC}	1	u_int8		
N_{Bsleep}	1	u_int8	slots	
N_{poll}	1	u_int8	slots	
op code	1	u_int8		
paging scheme	1	u_int8		0: mandatory scheme 1: optional scheme I 2: optional scheme II 3: optional scheme III 4-255: Reserved
paging scheme settings	1	u_int8		For mandatory scheme: 0: R0 1: R1 2: R2 3-255: Reserved For optional scheme 1: 0: Reserved 1: R1 2: R2 3-255: Reserved
PM_ADDR	1	u_int8		
poll interval	2	u_int16	slots	
random number	16	multiple bytes		
reason	1	u_int8		See Table 5.4 on page 244.
SCO handle	1	u_int8		
SCO packet	1	u_int8		0: HV1 1: HV2 2: HV3 3-255: Reserved
slot offset	2	u_int16	µs	0 ≤ slot offset < 1250
sniff attempt	2	u_int16	slots	Number of receive slots
sniff timeout	2	u_int16	slots	Number of receive slots
SubVersNr	2	u_int16		Defined by each company
supervision time-out	2	u_int16	slots	0 means an infinite time-out

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed
switch instant	4	u_int32	slots	Bits 27:1 of the master Bluetooth clock value
T_{access}	1	u_int8	slots	
T_B	2	u_int16	slots	
timing control flags	1	u_int8		bit0 = 0: no timing change bit0 = 1: timing change bit1 = 0: use initialization 1 bit1 = 1: use initialization 2 bit2 = 0: access window bit2 = 1: no access window bit3-7: Reserved
T_{sco}	1	u_int8	slots	
T_{sniff}	2	u_int16	slots	
VersNr	1	u_int8		See Bluetooth Assigned Numbers, (http://www.bluetooth.org/assigned-numbers.htm)
Δ_B	1	u_int8	slots	

Table 5.2: Parameters in LM PDUs.

5.1.1 Coding of features

This parameter is a bitmap with information about the Bluetooth radio-, baseband- and LMP features which a device supports. The bit shall be one if the feature is supported. In addition to the bitmap information the feature parameter has a 3-bit field denoted flow control lag. This is defined as the total amount of L2CAP data that can be sent following the receipt of a valid payload header with the payload header flow bit set to 0 and is in units of 256 Byte. See further in Baseband Specification Section 4.5.2, on page 62. The feature parameter bits that are not defined in Table 5.3 shall be zero.

Byte	Bit	Supported feature
0	0	3-slot packets
	1	5-slot packets
	2	encryption
	3	slot offset
	4	timing accuracy
	5	switch
	6	hold mode
	7	sniff mode
1	0	park mode
	1	RSSI
	2	channel quality driven data rate
	3	SCO link
	4	HV2 packets
	5	HV3 packets
	6	u-law log
	7	A-law log
	0	CVSD
	1	paging scheme
	2	power control
2	3	transparent SCO data
	4	Flow control lag (bit0)
	5	Flow control lag (bit1)
	6	Flow control lag (bit2)

Table 5.3: Coding of the parameter features.

5.1.2 List of error reasons

The following table contains the codes of the different error reasons used in LMP.

Reason	Description
0x05	Authentication Failure
0x06	Key Missing
0x0A	Max Number Of SCO Connections To A Device (The maximum number of SCO connections to a particle device has been reached. All allowed SCO connection handles to that device are used.)
0x0D	Host Rejected due to limited resources (The host at the remote side has rejected the connection because the remote host did not have enough additional resources to accept the connection.)
0x0E	Host Rejected due to security reasons (The host at the remote side has rejected the connection because the remote host determined that the local host did not meet its security criteria.)
0x0F	Host Rejected due to remote device is only a personal device (The host at the remote side has rejected the connection because the remote host is a personal device and will only accept the connection from one particle remote host.)
0x10	Host Timeout (Used at connection accept timeout, the host did not respond to an incoming connection attempt before the connection accept timer expired.)
0x13	Other End Terminated Connection: User Ended Connection
0x14	Other End Terminated Connection: Low Resources
0x15	Other End Terminated Connection: About to Power Off
0x16	Connection Terminated by Local Host
0x17	Repeated Attempts (An authentication or pairing attempt is made too soon after a previously failed authentication or pairing attempt.)
0x18	Pairing Not Allowed
0x19	Unknown LMP PDU
0x1A	Unsupported LMP Feature
0x1B	SCO Offset Rejected
0x1C	SCO Interval Rejected
0x1D	SCO Air Mode Rejected
0x1E	Invalid LMP Parameters
0x1F	Unspecified Error
0x20	Unsupported parameter value
0x21	Switch not allowed
0x23	LMP Error Transaction Collision
0x24	PDU not allowed
0x25	Encryption mode not acceptable

Table 5.4: List of error reasons.

Reason	Description
0x26	Unit key used
0x27	QoS not supported
0x28	Instant passed
0x29	Pairing with unit key not supported

Table 5.4: List of error reasons.

5.2 DEFAULT VALUES

The Bluetooth device must use these values before anything else has been negotiated:

Parameter	Value
drift	250
jitter	10
max slots	1
poll interval	40

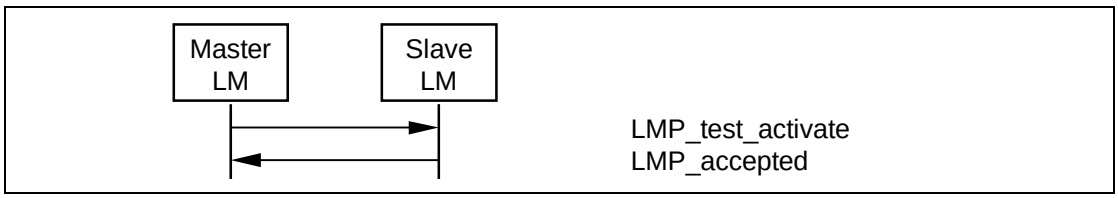
Table 5.5: Default values.

6 TEST MODES

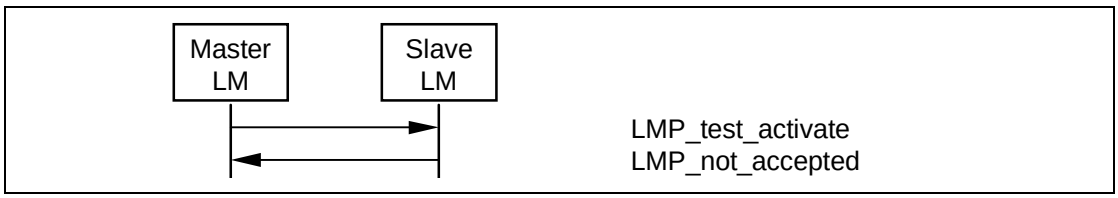
LMP has PDUs to support different Bluetooth test modes, which are used for certification and compliance testing of the Bluetooth radio and baseband. See “Bluetooth Test Mode” on page 827 for a detailed description of these test modes.

6.1 ACTIVATION AND DEACTIVATION OF TEST MODE

The test mode is activated by sending LMP_test_activate to the device under test (DUT). The DUT is always the slave. The link manager must be able to receive this message anytime. If entering test mode is locally enabled in the DUT it responds with LMP_accepted and test mode is entered. Otherwise the DUT responds with LMP_not_accepted and the DUT remains in normal operation. The reason code in LMP_not_accepted shall be *PDU not allowed*.



Sequence 59: Activation of test mode successful.

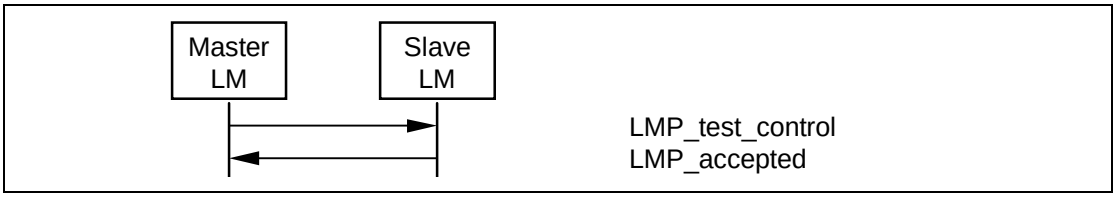


Sequence 60: Activation of test mode fails. Slave is not allowed to enter test mode.

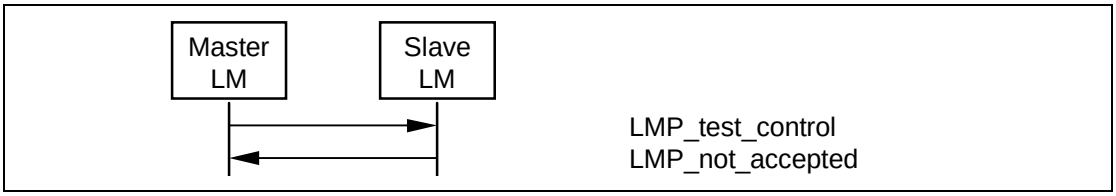
The test mode can be deactivated in two ways. Sending LMP_test_control with the test scenario set to "exit test mode" exits the test mode and the slave returns to normal operation still connected to the master. Sending LMP_detach to the DUT ends the test mode and the connection.

6.2 CONTROL OF TEST MODE

When the DUT has entered test mode, the PDU LMP_test_control can be sent to the DUT to start a specific test. This PDU is acknowledged with LMP_accepted. If a device that is not in test mode receives LMP_test_control it responds with LMP_not_accepted, where the reason code shall be *PDU not allowed*.



Sequence 61: Control of test mode successful.



Sequence 62: Control of test mode rejected since slave is not in test mode.

6.3 SUMMARY OF TEST MODE PDUs

The PDUs used for test purposes are summarized in the following table. For a detailed description of the parameters, see Bluetooth Test Mode Table 3.2 on page 842.

M/O	LMP PDU	Length	op code	Packet type	Possible direction	Contents	Position in payload
M	LMP_test_activate	1	56	DM1/DV	m → s	-	
M	LMP_test_control	10	57	DM1	m → s	test scenario	2
						hopping mode	3
						TX frequency	4
						RX frequency	5
						power control mode	6
						poll period	7
						packet type	8
						length of test data	9-10

Table 6.1: Test mode PDUs.

7 ERROR HANDLING

If the Link Manager receives a PDU with unrecognized opcode, it responds with LMP_not_accepted with the reason code *unknown LMP PDU.* The opcode parameter that is echoed back is the unrecognized opcode.

If the Link Manager receives a PDU with invalid parameters, it responds with LMP_not_accepted with the reason code *invalid LMP parameters*.

If the maximum response time, see Section 1 on page 189, is exceeded or if a link loss is detected (see Baseband Specification Section 10.11, on page 124), the party that waits for the response shall conclude that the procedure has terminated unsuccessfully.

Erroneous LMP messages can be caused by errors on the channel or systematic errors at the transmit side. To detect the latter case, the LM should monitor the number of erroneous messages and disconnect if it exceeds a threshold, which is implementation-dependent.

Since LMP PDUs are not interpreted in real time, collision situations can occur where both LMs initiate the same procedure and both cannot be completed. In this situation, the master shall reject the slave-initiated procedure by sending LMP_not_accepted with the reason code 'LMP Error Transaction Collision'. The master-initiated procedure shall then be completed.

When the Link Manager receives a PDU that is not allowed, if the PDU normally expects a PDU reply, for example LMP_host_connection_req or LMP_unit_key, the PDU LMP_not_accepted with the reason code “PDU not allowed” will be returned. If the PDU normally doesn’t expect a reply, for example LMP_sres or LMP_temp_key, the PDU will be ignored.

8 LIST OF FIGURES

Figure 1.1: Link Manager's place on the global scene. 191
Figure 2.1: Payload body when LM PDUs are sent. 193
Figure 3.1: Symbols used in sequence diagrams. 195

Sequence 1: Authentication. Claimant has link key. 196
Sequence 2: Authentication fails. Claimant has no link key. 197
Sequence 3: Pairing accepted. Responder has a variable PIN.Initiator has a variable or fixed PIN. 198
Sequence 4: Responder has a fixed PIN and initiator has a variable PIN. 198
Sequence 5: Both devices have a fixed PIN. 198
Sequence 6: Claimant rejects pairing. 199
Sequence 7: Creation of the link key. 199
Sequence 8: Successful change of the link key. 200
Sequence 9: Change of the link key not possible since the other unit uses a unit key. 200
Sequence 10: Change to a temporary link key. 201
Sequence 11: Link key changed to the semi-permanent link key. 202
Sequence 12: Negotiation for encryption mode. 203
Sequence 13: Encryption key size negotiation successful. 204
Sequence 14: Encryption key size negotiation failed. 205
Sequence 15: Start of encryption. 205
Sequence 16: Stop of encryption. 206
Sequence 17: Clock offset requested. 207
Sequence 18: Slot offset information is sent. 207
Sequence 19: The requested device supports timing accuracy information. 208
Sequence 20: The requested device does not support timing accuracy information. 208
Sequence 21: Request for LMP version. 209
Sequence 22: Request for supported features. 210
Sequence 23: Master-slave switch (slave initiated). 211
Sequence 24: Master-slave switch (master initiated). 211
Sequence 25: Device's name requested and it responses. 212
Sequence 26: Connection closed by sending LMP_detach. 213
Sequence 27: Master forces slave into hold mode. 214
Sequence 28: Slave forces master into hold mode. 215
Sequence 29: Negotiation for hold mode. 216
Sequence 30: Negotiation for sniff mode. 217
Sequence 31: Slave moved from sniff mode to active mode. 217
Sequence 32: Slave accepts to enter park mode. 220

Sequence 33: Slave rejects to enter into park mode 220
Sequence 34: Slave requests to enter park mode and accepts master's beacon parameters. 221
Sequence 35: Master rejects slave's request to enter park mode 221
Sequence 36: Slave requests to enter park mode, but rejects master's beacon parameters. 221
Sequence 37: Master accepts and places slave in park mode. 222
Sequence 38: Master rejects to place slave in park mode. 222
Sequence 39: Master notifies all slaves of increase in broadcast capacity. 222
Sequence 40: Master modifies beacon parameters. 223
Sequence 41: Master unparks slaves addressed with their BD_ADDR. ... 223
Sequence 42: Master unparks slaves addressed with their PM_ADDR. ... 224
Sequence 43: A device requests a change of the other device's TX power. 224
Sequence 44: The TX power cannot be increased. 225
Sequence 45: The TX power cannot be decreased. 225
Sequence 46: The left-hand unit is configured to automatically change between DM and DH. 226
Sequence 47: A wants B to control A's packet type. 226
Sequence 48: B changes A's packet type. 227
Sequence 49: Device accepts new quality of service 228
Sequence 50: Device rejects new quality of service. 228
Sequence 51: Master requests an SCO link. 229
Sequence 52: Master rejects slave's request for an SCO link. 230
Sequence 53: Master accepts slave's request for an SCO link. 230
Sequence 54: SCO link removed. 231
Sequence 55: Master allows slave to use a maximal number of slots. 231
Sequence 56: Device allows remote device to use a maximal number of slots. 231
Sequence 57: Slave requests to use a maximal number of slots. Master rejects. 231
Sequence 58: Negotiation for page mode. 232
Sequence 59: Negotiation for page scan mode 232
Sequence 60: Setting the link supervision timeout. 233
Figure 4.1: Connection establishment. 234
Sequence 61: Activation of test mode successful. 248
Sequence 62: Activation of test mode fails. Slave is not allowed to enter test mode. 248
Sequence 63: Control of test mode successful. 249
Sequence 64: Control of test mode rejected since slave is not in test mode. 249

9 LIST OF TABLES

Table 2.1: Logical channel L_CH field contents. 193
Table 3.1: General response messages. 195
Table 3.2: PDUs used for authentication. 196
Table 3.3: PDUs used for pairing 197
Table 3.4: PDUs used for change of link key. 200
Table 3.5: PDUs used to change the current link key. 201
Table 3.6: PDUs used for handling encryption. 203
Table 3.8: PDU used for slot offset information. 207
Table 3.7: PDUs used for clock offset request. 207
Table 3.9: PDUs used for requesting timing accuracy information. 208
Table 3.10: PDUs used for LMP version request. 209
Table 3.11: PDUs used for features request. 210
Table 3.12: PDUs used for name request. 212
Table 3.13: PDU used for detach. 213
Table 3.14: PDUs used for hold mode. 214
Table 3.15: PDUs used for sniff mode. 216
Table 3.16: PDUs used for park mode. 219
Table 3.17: PDUs used for power control. 224
Table 3.18: PDUs used for quality driven change of the data rate. 226
Table 3.19: PDUs used for quality of service. 227
Table 3.20: PDUs used for managing the SCO links. 228
Table 3.21: PDUs used to control the use of multi-slot packets. 231
Table 3.22: PDUs used to request paging scheme. 232
Table 3.23: PDU used to set the supervision timeout. 233
Table 4.1: PDUs used for connection establishment. 235
Table 5.1: Coding of the different LM PDUs. 236
Table 5.2: Parameters in LM PDUs. 241
Table 5.3: Coding of the parameter features. 245
Table 5.4: List of error reasons. 246
Table 5.5: Default values. 247
Table 6.1: Test mode PDUs. 249

Part D

LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL SPECIFICATION

This document describes the Bluetooth logical link control and adaptation protocol (L2CAP). This protocol supports higher level protocol multiplexing, packet segmentation and reassembly, and the conveying of quality of service information. This document is part of the Bluetooth Specification. This document describes the protocol state machine, packet format and composition, and a test interface required for the Bluetooth test and certification program.

CONTENTS

1 Introduction 259
1.1 L2CAP Functional Requirements 260
1.2 Assumptions 262
1.3 Scope 262
2 General Operation 263
2.1 Channel Identifiers 263
2.2 Operation Between Devices 263
2.3 Operation Between Layers 264
2.4 Segmentation and Reassembly 265
2.4.1 Segmentation Procedures 266
2.4.2 Reassembly Procedures 266
3 State Machine 268
3.1 Events 269
3.1.1 Lower-Layer Protocol (LP) to L2CAP events 269
3.1.2 L2CAP to L2CAP Signalling events 270
3.1.3 L2CAP to L2CAP Data events 271
3.1.4 Upper-Layer to L2CAP events 271
3.1.5 Timer events 272
3.2 Actions 273
3.2.1 L2CAP to Lower Layer actions 273
3.2.2 L2CAP to L2CAP Signalling actions 274
3.2.3 L2CAP to L2CAP Data actions 274
3.2.4 L2CAP to Upper Layer actions 274
3.3 Channel Operational States 275
3.4 Mapping Events to Actions 276
4 Data Packet Format 282
4.1 Connection-oriented Channel 282
4.2 Connectionless Data Channel 283
5 Signalling 285
5.1 Command Reject (code 0x01) 287
5.2 Connection Request (code 0x02) 288
5.3 Connection Response (code 0x03) 289
5.4 Configuration Request (code 0x04) 290
5.5 Configure Response (code 0x05) 292
5.6 Disconnection Request (code 0x06) 295
5.7 Disconnection Response (code 0x07) 296
5.8 Echo Request (code 0x08) 296
5.9 Echo Response (code 0x09) 297
5.10 Information Request (CODE 0x0A) 297
5.11 Information Response (CODE 0x0B) 298

6 Configuration Parameter Options 299
6.1 Maximum Transmission Unit (MTU) 299
6.2 Flush Timeout Option 300
6.3 Quality of Service (QoS) Option 301
6.4 Configuration Process 303
6.4.1 Request Path 303
6.4.2 Response Path 304
6.4.3 Configuration State Machine 304
7 Service Primitives 305
7.1 Event Indication 305
7.1.1 L2CA_ConnectInd Callback 306
7.1.2 L2CA_ConfigInd Callback 306
7.1.3 L2CA_DisconnectInd Callback 306
7.1.4 L2CA_QoSViolationInd Callback 306
7.2 Connect 306
7.3 Connect Response 308
7.4 Configure 309
7.5 Configuration Response 311
7.6 Disconnect 312
7.7 Write 313
7.8 Read 314
7.9 Group Create 315
7.10 Group Close 315
7.11 Group Add Member 316
7.12 Group Remove Member 317
7.13 Get Group Membership 318
7.14 Ping 319
7.15 GetInfo 320
7.16 Disable Connectionless Traffic 321
7.17 Enable Connectionless Traffic 322
8 Summary 323
9 References 324
10 List of Figures 325
11 List of Tables 326
Terms and Abbreviations 327
Appendix A: Configuration MSCs 328
Appendix B: Implementation Guidelines 331

1 INTRODUCTION

This section of the Bluetooth Specification defines the Logical Link Control and Adaptation Layer Protocol, referred to as L2CAP. L2CAP is layered over the Baseband Protocol and resides in the data link layer as shown in Figure 1.1. L2CAP provides connection-oriented and connectionless data services to upper layer protocols with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions. L2CAP permits higher level protocols and applications to transmit and receive L2CAP data packets up to 64 kilobytes in length.

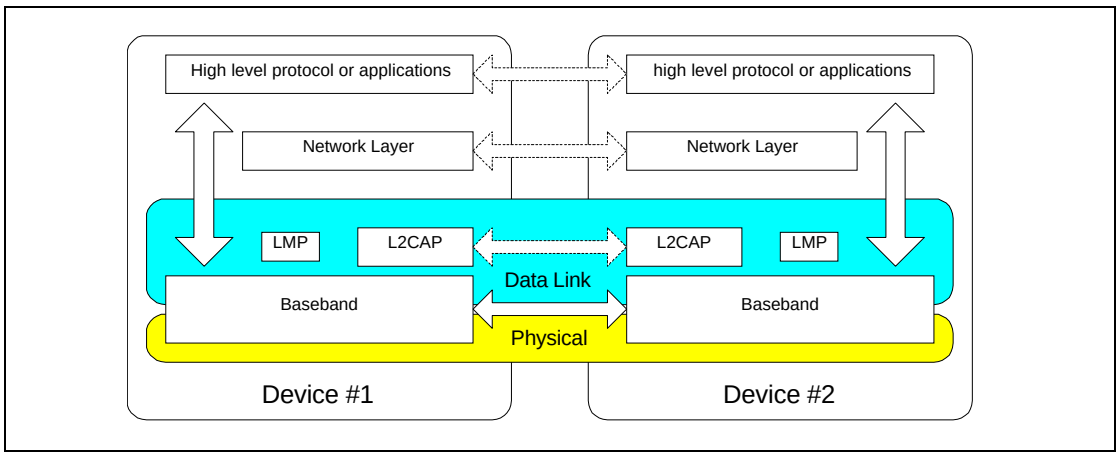


Figure 1.1: L2CAP within protocol layers

The "Baseband Specification" on page 33 defines two link types: Synchronous Connection-Oriented (SCO) links and Asynchronous Connection-Less (ACL) links. SCO links support real-time voice traffic using reserved bandwidth. ACL links support best effort traffic. The L2CAP Specification is defined for only ACL links and no support for SCO links is planned.

For ACL links, use of the AUX1 packet on the ACL link is prohibited. This packet type supports no data integrity checks (no CRC). Because L2CAP depends on integrity checks in the Baseband to protect the transmitted information, AUX1 packets must never be used to transport L2CAP packets.

The format of the ACL payload header is shown below. Figure 1.2 on page 258 displays the payload header used for single-slot packets and Figure 1.3 displays the header used in multi-slot packets. The only difference is the size of the length field. The packet type (a field in the Baseband header) distinguishes single-slot packets from multi-slot packets.

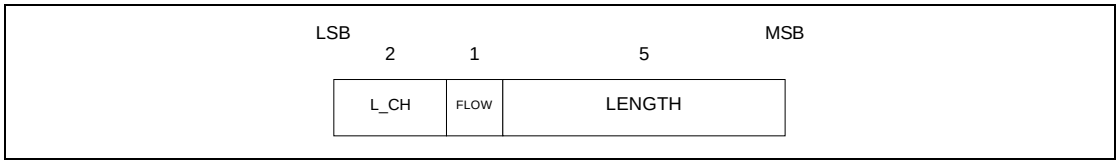


Figure 1.2: ACL Payload Header for single-slot packets

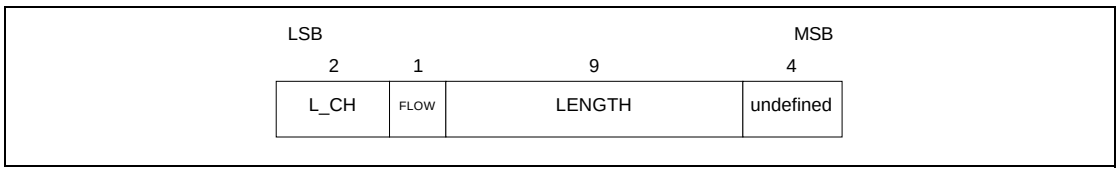


Figure 1.3: ACL Payload Header for multi-slot packets

The 2-bit logical channel (L_CH) field, defined in Table 1.1, distinguishes L2CAP packets from Link Manager Protocol (page 183) packets. The remaining code is reserved for future use.

L_CH code	Logical Channel	Information
00	RESERVED	Reserved for future use
01	L2CAP	Continuation of L2CAP packet
10	L2CAP	Start of L2CAP packet
11	LMP	Link Manager Protocol

Table 1.1: Logical channel L_CH field contents

The FLOW bit in the ACL header is managed by the Link Controller (LC), a Baseband implementation entity, and is normally set to1 (‘flow on’). It is set to 0 (‘flow off’) when no further L2CAP traffic shall be sent over the ACL link. Sending an L2CAP packet with the FLOW bit set to 1 resumes the flow of incoming L2CAP packets. This is described in more detail in “Baseband Specification” on page 33.

1.1 L2CAP FUNCTIONAL REQUIREMENTS

The functional requirements for L2CAP include protocol multiplexing, segmentation and reassembly (SAR), and group management. Figure 1.4 illustrates how L2CAP fits into the Bluetooth Protocol Stack. L2CAP lies above the Baseband Protocol (page 33) and interfaces with other communication protocols such as the Bluetooth Service Discovery Protocol (SDP, page 331), RFCOMM (page 393), and Telephony Control (TCS, page 443). Voice-quality channels for audio and telephony applications are usually run over Baseband SCO links. Packetized audio data, such as IPTelephony, may be sent using communication protocols running over L2CAP.

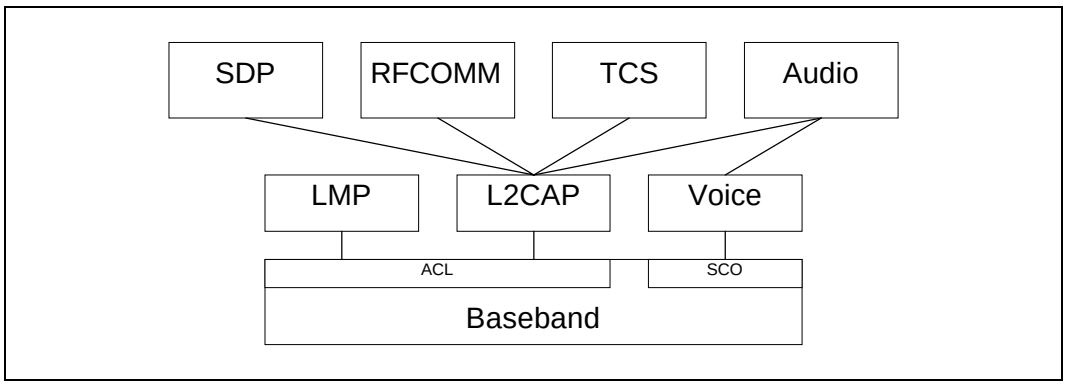


Figure 1.4: L2CAP in Bluetooth Protocol Architecture

Essential protocol requirements for L2CAP include simplicity and low overhead. Implementations of L2CAP must be applicable for devices with limited computational resources. L2CAP should not consume excessive power since that significantly sacrifices power efficiency achieved by the Bluetooth Radio. Memory requirements for protocol implementation should also be kept to a minimum.

The protocol complexity should be acceptable to personal computers, PDAs, digital cellular phones, wireless headsets, joysticks and other wireless devices supported by Bluetooth. Furthermore, the protocol should be designed to achieve reasonably high bandwidth efficiency.

- *Protocol Multiplexing*

  L2CAP must support protocol multiplexing because the Baseband Protocol does not support any 'type' field identifying the higher layer protocol being multiplexed above it. L2CAP must be able to distinguish between upper layer protocols such as the Service Discovery Protocol (page 331), RFCOMM (page 393), and Telephony Control (page 443).

- *Segmentation and Reassembly*

  Compared to other wired physical media, the data packets defined by the Baseband Protocol (page 33) are limited in size. Exporting a maximum transmission unit (MTU) associated with the largest Baseband payload (341 bytes for DH5 packets) limits the efficient use of bandwidth for higher layer protocols that are designed to use larger packets. Large L2CAP packets must be segmented into multiple smaller Baseband packets prior to their transmission over the air. Similarly, multiple received Baseband packets may be reassembled into a single larger L2CAP packet following a simple integrity check (described in Section 2.4.2 on page 264). The Segmentation and Reassembly (SAR) functionality is absolutely necessary to support protocols using packets larger than those supported by the Baseband.

- *Quality of Service*

  The L2CAP connection establishment process allows the exchange of information regarding the quality of service (QoS) expected between two Blue-

tooth units. Each L2CAP implementation must monitor the resources used by the protocol and ensure that QoS contracts are honoured.

- *Groups*

  Many protocols include the concept of a group of addresses. The Baseband Protocol supports the concept of a piconet, a group of devices synchronously hopping together using the same clock. The L2CAP group abstraction permits implementations to efficiently map protocol groups on to piconets. Without a group abstraction, higher level protocols would need to be exposed to the Baseband Protocol and Link Manager functionality in order to manage groups efficiently.

1.2 ASSUMPTIONS

The protocol is designed based on the following assumptions:

1. The ACL link between two units is set up using the Link Manager Protocol (page 183). The Baseband provides orderly delivery of data packets, although there might be individual packet corruption and duplicates. No more than 1 ACL link exists between any two devices.
2. The Baseband always provides the impression of full-duplex communication channels. This does not imply that all L2CAP communications are bi-directional. Multicasts and unidirectional traffic (e.g., video) do not require duplex channels.
3. L2CAP provides a reliable channel using the mechanisms available at the Baseband layer. The Baseband always performs data integrity checks when requested and resends data until it has been successfully acknowledged or a timeout occurs. Because acknowledgements may be lost, timeouts may occur even after the data has been successfully sent. The Baseband protocol uses a 1-bit sequence number that removes duplicates. Note that the use of Baseband broadcast packets is prohibited if reliability is required since all broadcasts start the first segment of an L2CAP packet with the same sequence bit.

1.3 SCOPE

The following features are outside the scope of L2CAP's responsibilities:

- L2CAP does not transport audio designated for SCO links.
- L2CAP does not enforce a reliable channel or ensure data integrity, that is, L2CAP performs no retransmissions or checksum calculations.
- L2CAP does not support a reliable multicast channel. See Section 4.2.
- L2CAP does not support the concept of a global group name.

2 GENERAL OPERATION

The Logical Link Control and Adaptation Protocol (L2CAP) is based around the concept of *'channels'*. Each one of the end-points of an L2CAP channel is referred to by a *channel identifier.*

2.1 CHANNEL IDENTIFIERS

Channel identifiers (CIDs) are local names representing a logical channel end-point on the device. Identifiers from 0x0001 to 0x003F are reserved for specific L2CAP functions. The null identifier (0x0000) is defined as an illegal identifier and must never be used as a destination end-point. Implementations are free to manage the remaining CIDs in a manner best suited for that particular implementation, with the provision that the same CID is not reused as a local L2CAP channel endpoint for multiple simultaneous L2CAP channels between a local device and some remote device. Table 2.1 summarizes the definition and partitioning of the CID name space.

CID assignment is relative to a particular device and a device can assign CIDs independently from other devices (unless it needs to use any of the reserved CIDs shown in the table below). Thus, even if the same CID value has been assigned to (remote) channel endpoints by several remote devices connected to a single local device, the local device can still uniquely associate each remote CID with a different device.

CID	Description
0x0000	Null identifier
0x0001	Signalling channel
0x0002	Connectionless reception channel
0x0003-0x003F	Reserved
0x0040-0xFFFF	Dynamically allocated

Table 2.1: CID Definitions

2.2 OPERATION BETWEEN DEVICES

Figure 2.1 on page 262 illustrates the use of CIDs in a communication between corresponding peer L2CAP entities in separate devices. The connection-oriented data channels represent a connection between two devices, where a CID identifies each endpoint of the channel. The connectionless channels restrict data flow to a single direction. These channels are used to support a channel 'group' where the CID on the source represents one or more remote devices. There are also a number of CIDs reserved for special purposes. The signalling channel is one example of a reserved channel. This channel is used to create and establish connection-oriented data channels and to negotiate changes in the characteristics of these channels. Support for a signalling chan-

nel within an L2CAP entity is mandatory. Another CID is reserved for all incoming connectionless data traffic. In the example below, a CID is used to represent a group consisting of device #3 and #4. Traffic sent from this channel ID is directed to the remote channel reserved for connectionless data traffic.

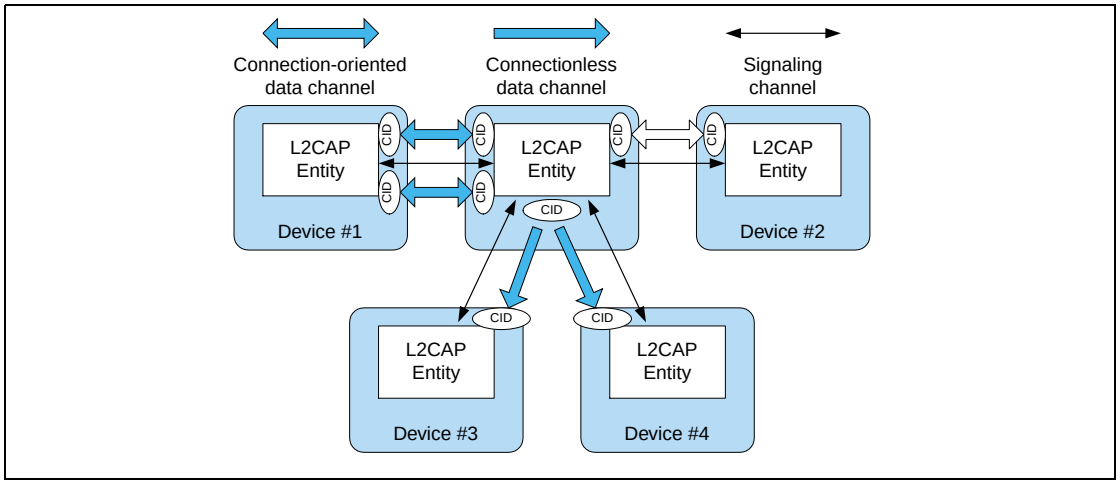


Figure 2.1: Channels between devices

Table 2.2 describes the various channels and their source and destination identifiers. An 'allocated' channel is created to represent the local endpoint and should be in the range 0x0040 to 0xFFFF. Section 3 on page 266 describes the state machine associated with each connection-orientedchannel. Section 4.1 on page 280 describes the packet format associated with bi-directional channels and Section 4.2 on page 281 describes the packet format associated with uni-directional channels.

Channel Type	Local CID	Remote CID
Connection-oriented	Dynamically allocated	Dynamically allocated
Connectionless data	Dynamically allocated	0x0002 (fixed)
Signalling	0x0001 (fixed)	0x0001 (fixed)

Table 2.2: Types of Channel Identifiers

2.3 OPERATION BETWEEN LAYERS

L2CAP implementations should follow the general architecture described below. L2CAP implementations must transfer data between higher layer protocols and the lower layer protocol. This document lists a number of services that should be exported by any L2CAP implementation. Each implementation must also support a set of signalling commands for use between L2CAP implementations. L2CAP implementations should also be prepared to accept certain types of events from lower layers and generate events to upper layers. How these events are passed between layers is an implementation-dependent process.

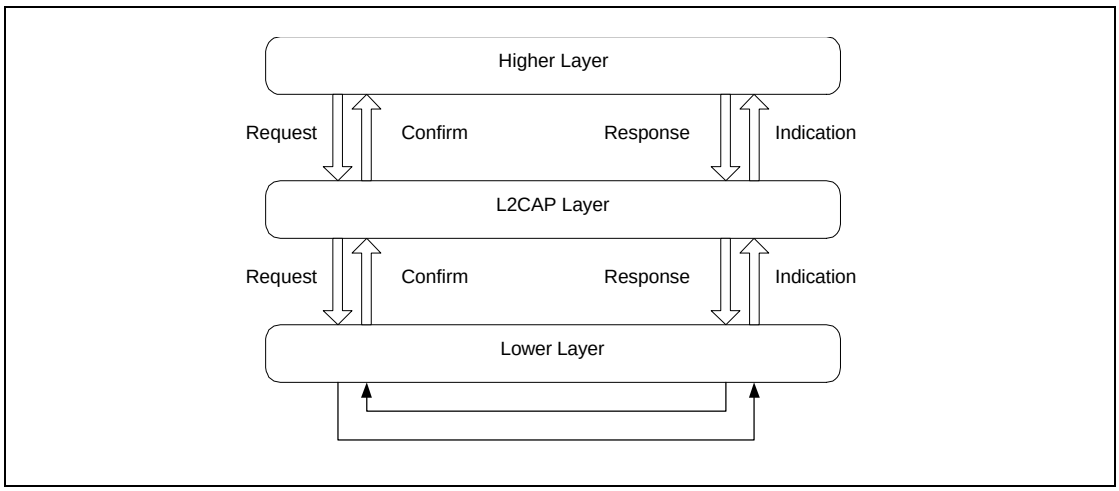


Figure 2.2: L2CAP Architecture

2.4 SEGMENTATION AND REASSEMBLY

Segmentation and reassembly (SAR) operations are used to improve efficiency by supporting a maximum transmission unit (MTU) size larger than the largest Baseband packet. This reduces overhead by spreading the network and transport packets used by higher layer protocols over several Baseband packets. All L2CAP packets may be segmented for transfer over Baseband packets. The protocol does not perform any segmentation and reassembly operations but the packet format supports adaptation to smaller physical frame sizes. An L2CAP implementation exposes the outgoing (i.e., the remote host's receiving) MTU and segments higher layer packets into 'chunks' that can be passed to the Link Manager via the Host Controller Interface (HCI), whenever one exists. On the receiving side, an L2CAP implementation receives 'chunks' from the HCI and reassembles those chunks into L2CAP packets using information provided through the HCI and from the packet header.

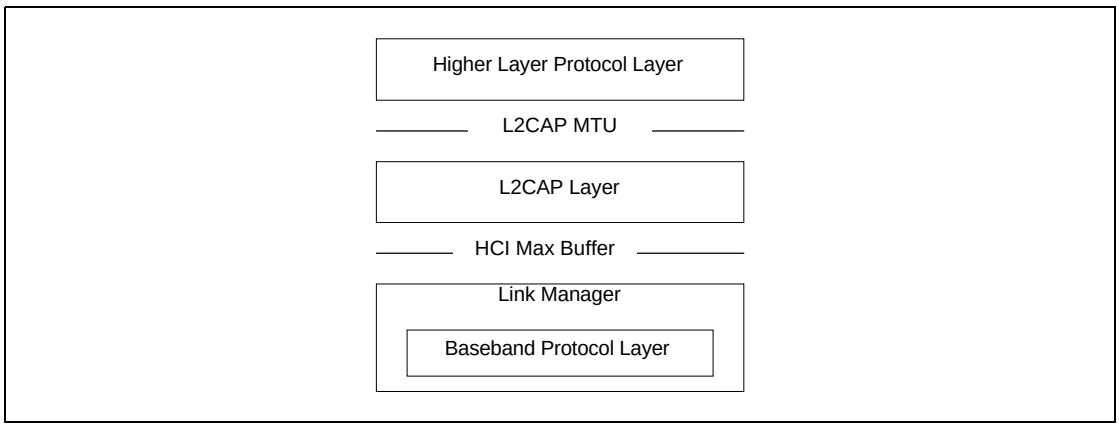


Figure 2.3: L2CAP SAR Variables

Segmentation and Reassembly is implemented using very little overhead in Baseband packets. The two L_CH bits defined in the first byte of Baseband

payload (also called the frame header) are used to signal the start and continuation of L2CAP packets. L_CH shall be ’10’ for the first segment in an L2CAP packet and ’01’ for a continuation segment. An example use of SAR is shown in Figure 2.4.

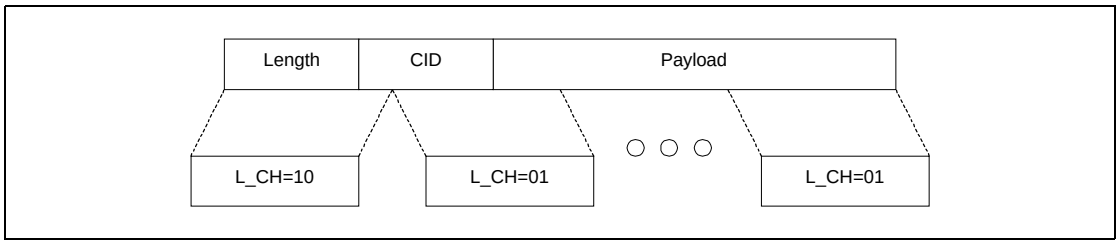


Figure 2.4: L2CAP segmentation

2.4.1 Segmentation Procedures

The L2CAP maximum transmission unit (MTU) will be exported using an implementation specific service interface. It is the responsibility of the higher layer protocol to limit the size of packets sent to the L2CAP layer below the MTU limit. An L2CAP implementation will segment the packet into protocol data units (PDUs) to send to the lower layer. If L2CAP runs directly over the Baseband Protocol, an implementation may segment the packet into Baseband packets for transmission over the air. If L2CAP runs above the host controller interface (typical scenario), an implementation may send block-sized chunks to the host controller where they will be converted into Baseband packets. All L2CAP segments associated with an L2CAP packet must be passed through to the Baseband before any other L2CAP packet destined to the same unit may be sent.

2.4.2 Reassembly Procedures

The Baseband Protocol delivers ACL packets in sequence and protects the integrity of the data using a 16-bit CRC. The Baseband also supports reliable connections using an automatic repeat request (ARQ) mechanism. As the Baseband controller receives ACL packets, it either signals the L2CAP layer on the arrival of each Baseband packets, or accumulates a number of packets before the receive buffer fills up or a timer expires before signalling the L2CAP layer.

L2CAP implementations must use the length field in the header of L2CAP packets, see Section 4 on page 280, as a consistency check and discard any L2CAP packets that fail to match the length field. If channel reliability is not needed, packets with improper lengths may be silently discarded. For reliable channels, L2CAP implementations must indicate to the upper layer that the channel has become unreliable. Reliable channels are defined by having an infinite flush timeout value as specified in Section 6.2 on page 298.

Figure 2.5 on pa g e265illustrates the use of segmentation and reassembly operations to transmit a single higher layer PDU. Note that while there is a one-to-one mapping between a high layer PDU and an L2CAP packet, the segment

size used by the segmentation and reassembly routines is left to the implementation and may differ from the sender to the receiver.

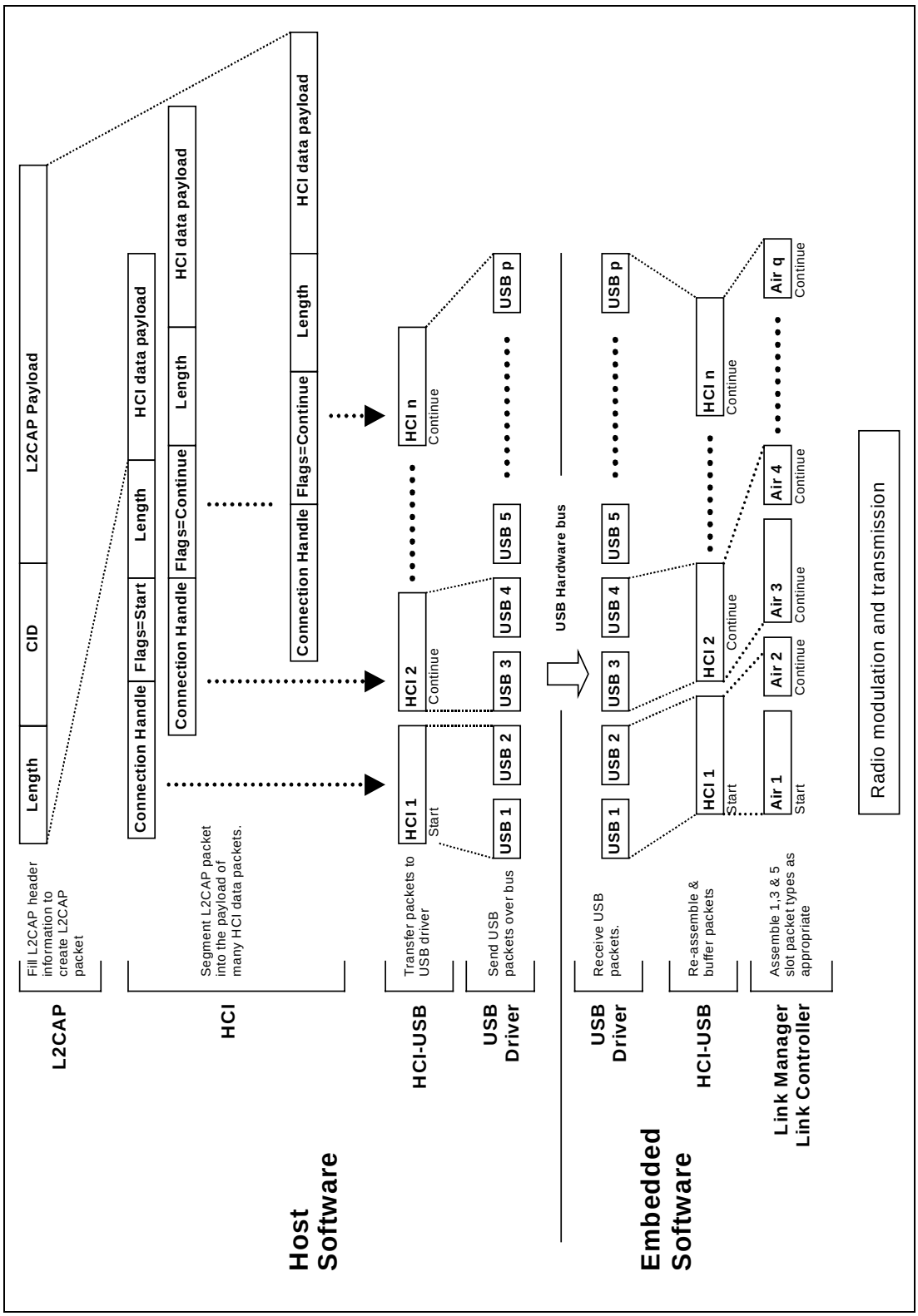


Figure 2.5: Segmentation and Reassembly Services in a unit with an HCI[1]

3 STATE MACHINE

This section describes the L2CAP connection-oriented channel state machine. The section defines the states, the events causing state transitions, and the actions to be performed in response to events. This state machine is only pertinent to bi-directional CIDs and is not representative of the signalling channel or the uni-directional channel.

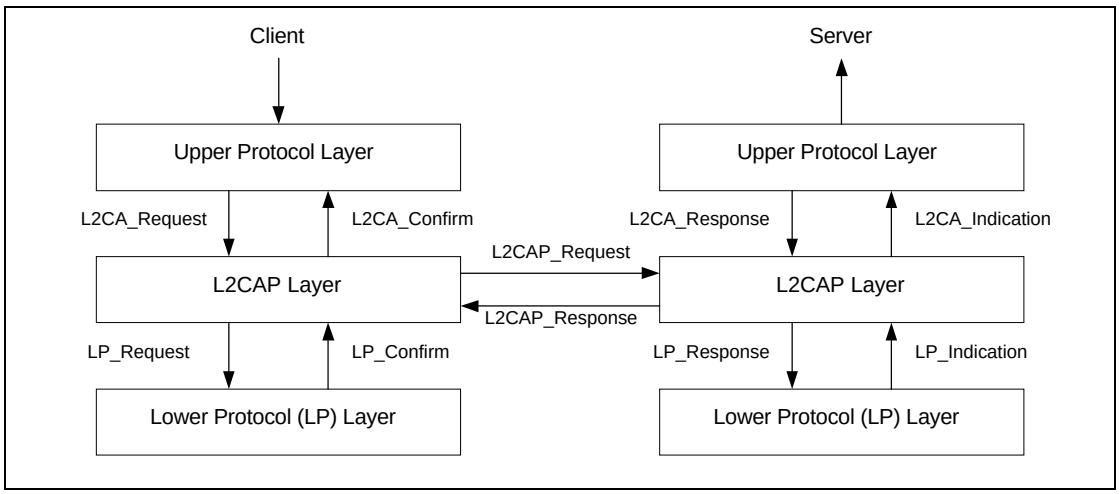


Figure 3.1: L2CAP Layer Interactions

Figure 3.1 illustrates the events and actions performed by an implementation of the L2CAP layer. Client and Server simply represent the initiator of the request and the acceptor of the request respectively. An application-level Client would both initiate and accept requests. The naming convention is as follows. The interface between two layers (vertical interface) uses the prefix of the lower layer offering the service to the higher layer, e.g., L2CA. The interface between two entities of the same layer (horizontal interface) uses the prefix of the protocol (adding a P to the layer identification), e.g., L2CAP. Events coming from above are called Requests (Req) and the corresponding replies are called Confirms (Cfm). Events coming from below are called Indications (Ind) and the corresponding replies are called Responses (Rsp). Responses requiring further processing are called Pending (Pnd). The notation for Confirms and Responses assumes positive replies. Negative replies are denoted by a 'Neg' suffix such as L2CAP_ConnectCfmNeg.

While Requests for an action always result in a corresponding Confirmation (for the successful or unsuccessful satisfaction of the action), Indications do not always result into corresponding Responses. The latter is especially true, if the Indications are informative about locally triggered events, e.g., seeing the

1. For simplicity, the stripping of any additional HCI and USB specific information fields prior to the creation of the baseband packets (Air_1, Air_2, etc.) is not shown in the figure.

LP_QoSViolationInd in Section 3.1.1 on page 267, or *L2CA_TimeOutInd* in Section 3.2.4 on page 272.

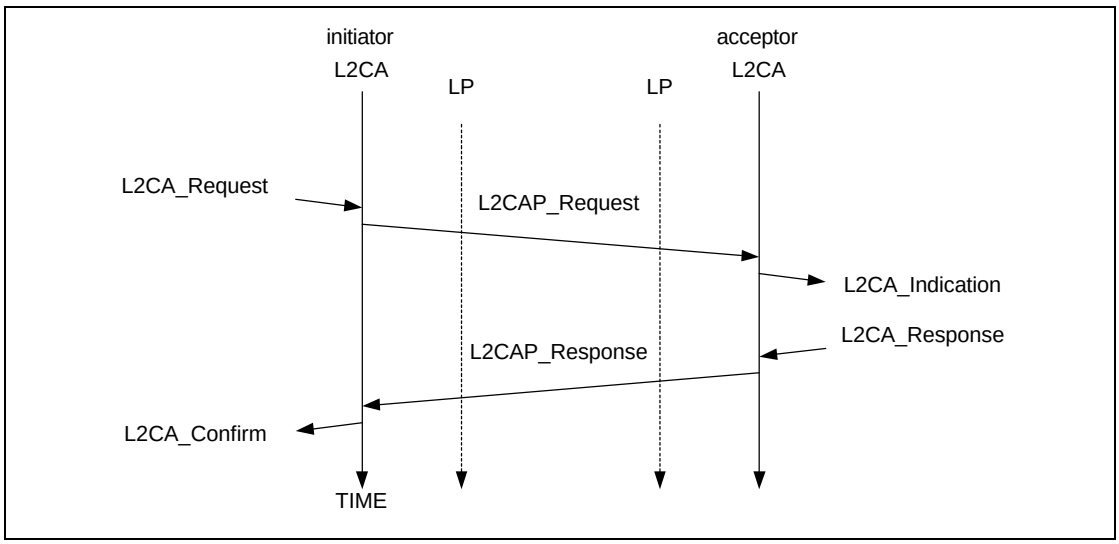


Figure 3.2: MSC of Layer Interactions

Figure 3.2 uses a message sequence chart (MSC) to illustrate the normal sequence of events. The two outer vertical lines represent the L2CA interface on the initiator (the device issuing a request) and the acceptor (the device responding to the initiator's request). Request commands at the L2CA interface result in Requests defined by the protocol. When the protocol communicates the request to the acceptor, the remote L2CA entity presents the upper protocol with an Indication. When the acceptor's upper protocol responds, the response is packaged by the protocol and communicated back the to initiator. The result is passed back to the initiator's upper protocol using a Confirm message.

3.1 EVENTS

Events are all incoming messages to the L2CA layer along with timeouts. Events are partitioned into five categories: Indications and Confirms from lower layers, Requests and Responses from higher layers, data from peers, signal Requests and Responses from peers, and events caused by timer expirations.

3.1.1 Lower-Layer Protocol (LP) to L2CAP events

- *LP_ConnectCfm*

  Confirms the request (see *LP_ConnectReq* in Section 3.2.1) to establish a lower layer (Baseband) connection. This includes passing the authentication challenge if authentication is required to establish the physical link.

- *LP_ConnectCfmNeg*

  Confirms the failure of the request (see *LP_ConnectReq* in Section 3.2.1) to establish a lower layer (Baseband) connection failed. This could be because

the device could not be contacted, refused the request, or the LMP authentication challenge failed.

- *LP_ConnectInd*

  Indicates the lower protocol has successfully established connection. In the case of the Baseband, this will be an ACL link. An L2CAP entity may use to information to keep track of what physical links exist.

- *LP_DisconnectInd*

  Indicates the lower protocol (Baseband) has been shut down by LMP commands or a timeout event.

- *LP_QoSCfm*

  Confirms the request (see *LP_QoSReq* in Section 3.2.1) for a given quality of service.

- *LP_QoSCfmNeg*

  Confirms the failure of the request (see *LP_QoSReq* in Section 3.2.1) for a given quality of service.

- *LP_QoSViolationInd*

  Indicates the lower protocol has detected a violation of the QoS agreement specified in the previous *LP_QoSReq* (see Section 3.2.1).

3.1.2 L2CAP to L2CAP Signalling events

L2CAP to L2CAP signalling events are generated by each L2CAP entity following the exchange of the corresponding L2CAP signalling PDUs, see Section 5. L2CAP signalling PDUs, like any other L2CAP PDUs, are received from a lower layer via a lower protocol indication event. For simplicity of the presentation, we avoid a detailed description of this process, and we assume that signalling events are exchanged directly between the L2CAP peer entities as shown in Figure 3.1 on page 266.

- *L2CAP_ConnectReq*

  A Connection Request packet has been received.

- *L2CAP_ConnectRsp*

  A Connection Response packet has been received with a positive result indicating that the connection has been established.

- *L2CAP_ConnectRspPnd*

  A Connection Response packet has been received indicating the remote endpoint has received the request and is processing it.

- *L2CAP_ConnectRspNeg*

  A Connection Response packet has been received, indicating that the connection could not be established.

- *L2CAP_ConfigReq*

  A Configuration Request packet has been received indicating the remote endpoint wishes to engage in negotiations concerning channel parameters.

- *L2CAP_ConfigRsp*

  A Configuration Response packet has been received indicating the remote endpoint agrees with all the parameters being negotiated.

- *L2CAP_ConfigRspNeg*

  A Configuration Response packet has been received indicating the remote endpoint does not agree to the parameters received in the response packet.

- *L2CAP_DisconnectReq*

  A Disconnection Request packet has been received and the channel must initiate the disconnection process. Following the completion of an L2CAP channel disconnection process, an L2CAP entity should return the corresponding local CID to the pool of 'unassigned' CIDs.

- *L2CAP_DisconnectRsp*

  A Disconnection Response packet has been received. Following the receipt of this signal, the receiving L2CAP entity may return the corresponding local CID to the pool of unassigned CIDs. There is no corresponding negative response because the Disconnect Request must succeed.

3.1.3 L2CAP to L2CAP Data events

- *L2CAP_Data*

  A Data packet has been received.

3.1.4 Upper-Layer to L2CAP events

- *L2CA_ConnectReq*

  Request from upper layer for the creation of a channel to a remote device.

- *L2CA_ConnectRsp*

  Response from upper layer to the indication of a connection request from a remote device (see L2CA_ConnectInd in Section 3.2.4).

- *L2CA_ConnectRspNeg*

  Negative response (rejection) from upper layer to the indication of a connection request from a remote device (see *L2CA_ConnectInd* in Section 3.2.4).

- *L2CA_ConfigReq*

  Request from upper layer to (re)configure the channel.

- *L2CA_ConfigRsp*

  Response from upper layer to the indication of a (re) configuration request (see *L2CA_ConfigInd* in Section 3.2.4).

- *L2CA_ConfigRspNeg*

  A negative response from upper layer to the indication of a (re) configuration request (see *L2CA_ConfigInd* in Section 3.2.4).

- *L2CA_DisconnectReq*

  Request from upper layer for the immediate disconnection of a channel.

- *L2CA_DisconnectRsp*

  Response from upper layer to the indication of a disconnection request (see *L2CA_DisconnectInd* in Section 3.2.4). There is no corresponding negative response, the disconnect indication must always be accepted.

- *L2CA_DataRead*

  Request from upper layer for the transfer of received data from L2CAP entity to upper layer.

- *L2CA_DataWrite*

  Request from upper layer for the transfer of data from the upper layer to L2CAP entity for transmission over an open channel.

3.1.5 Timer events

- *RTX*

  The Response Timeout eXpired (RTX) timer is used to terminate the channel when the remote endpoint is unresponsive to signalling requests. This timer is started when a signalling request (see Section 5 on page 283) is sent to the remote device. This timer is disabled when the response is received. If the initial timer expires, a duplicate Request message may be sent or the channel identified in the request may be disconnected. If a duplicate Request message is sent, the RTX timeout value must be reset to a new value at least double the previous value.

  Implementations have the responsibility to decide on the maximum number of Request retransmissions performed at the L2CAP level before terminating the channel identified by the Requests. The one exception is the signalling CID that should never be terminated. The decision should be based on the flush timeout of the signalling link. The longer the flush timeout, the more retransmissions may be performed at the physical layer and the reliability of the channel improves, requiring fewer retransmissions at the L2CAP level. For example, if the flush timeout is infinite, no retransmissions should be performed at the L2CAP level. When terminating the channel, it is not necessary to send a L2CAP DisconnectReq and enter disconnection state.Channels should be transitioned directly to the Closed state.

- The value of this timer is implementation-dependent but the minimum initial value is 1 second and the maximum initial value is 60 seconds. One RTX timer MUST exist for each outstanding signalling request, including each Echo Request. The timer disappears on the final expiration, when the response is received, or the physical link is lost. The maximum elapsed time between the initial start of this timer and the initiation of channel termination (if no response is received) is 60 seconds.*ERTX*

The Extended Response Timeout eXpired (ERTX) timer is used in place of the RTX timer when it is suspected the remote endpoint is performing additional processing of a request signal. This timer is started when the remote endpoint responds that a request is pending, e.g., when an *L2CAP_ConnectRspPnd* event is received. This timer is disabled when the formal response is received or the physical link is lost. If the initial timer expires, a duplicate Request may be sent or the channel may be disconnected. If a duplicate Request is sent, the particular ERTX timer disappears, replaced by a new RTX timer and the whole timing procedure restarts as described previously for the RTX timer.

The value of this timer is implementation-dependent but the minimum initial value is 60 seconds and the maximum initial value is 300 seconds. Similar to RTX, there MUST be at least one ERTX timer for each outstanding request that received a Pending response. There should be at most one (RTX or ERTX) associated with each outstanding request. The maximum elapsed time between the initial start of this timer and the initiation of channel termination (if no response is received) is 300 seconds. When terminating the channel, it is not necessary to send a L2CAP DisconnectReq and enter disconnection state. Channels should be transitioned directly to the Closed state.

3.2 ACTIONS

Actions are partitioned into five categories: Confirms and Indications to higher layers, Request and Responses to lower layers, Requests and Responses to peers, data transmission to peers, and setting timers.

3.2.1 L2CAP to Lower Layer actions

- *LP_ConnectReq*

  L2CAP requests the lower protocol to create a connection. If a physical link to the remote device does not exist, this message must be sent to the lower protocol to establish the physical connection. Since no more than a single ACL link between two devices is assumed, see Section 1.2 on page 260, additional L2CAP channels between these two devices must share the same baseband ACL link.
  Following the processing of the request, the lower layer returns with an *LP_ConnectCfm* or an *LP_ConnectCfmNeg* to indicate whether the request has been satisfied or not, respectively.

- *LP_QoSReq*

  L2CAP requests the lower protocol to accommodate a particular QoS parameter set. Following the processing of the request, the lower layer returns with an *LP_QoSCfm* or an *LP_QoSCfmNeg* to indicate whether the request has been satisfied or not, respectively

- *LP_ConnectRsp*

  A positive response accepting the previous connection indication request (see *LP_ConnectInd* in Section 3.1.1).

- *LP_ConnectRspNeg*

  A negative response denying the previous connection indication request (see *LP_ConnectInd* in Section 3.1.1).

3.2.2 L2CAP to L2CAP Signalling actions

This section contains the same names identified in Section 3.1.2 except the actions refer to the transmission, rather than reception, of these messages.

3.2.3 L2CAP to L2CAP Data actions

This section is the counterpart of Section 3.1.3. Data transmission is the action performed here.

3.2.4 L2CAP to Upper Layer actions

- *L2CA_ConnectInd*

  Indicates a Connection Request has been received from a remote device (see *L2CA_ConnectReq* in Section 3.1.4).

- *L2CA_ConnectCfm*

  Confirms that a Connection Request has been accepted (see (*L2CAP_ConnectReq* in Section 3.1.4) following the receipt of a Connection message from the remote device.

- *L2CA_ConnectCfmNeg*

  Negative confirmation (failure) of a Connection Request (see *L2CA_ConnectReq* in Section 3.1.4). An RTX timer expiration (see Section 3.1.5 and *L2CA_TimeOutInd* below) for an outstanding Connect Request can substitute for a negative Connect Response and result in this action.

- *L2CA_ConnectPnd*

  Confirms that a Connection Response (pending) has been received from the remote device.

- *L2CA_ConfigInd*

  Indicates a Configuration Request has been received from a remote device.

- *L2CA_ConfigCfm*

  Confirms that a Configuration Request has been accepted (see *L2CA_ConfigReq* in Section 3.1.4) following the receipt of a Configuration Response from the remote device.

- *L2CA_ConfigCfmNeg*

  Negative confirmation (failure) of a Configuration Request (see *L2CA_ConfigReq* in Section 3.1.4). An RTX timer expiration (see Section 3.1.5 and *L2CA_TimeOutInd* below) for an outstanding Connect Request can substitute for a negative Connect Response and result in this action.

- *L2CA_DisconnectInd*

  Indicates a Disconnection Request has been received from a remote device or the remote device has been disconnected because it has failed to respond to a signalling request. See Section 3.1.5

- *L2CA_DisconnectCfm*

  Confirms that a Disconnect Request has been processed by the remote device (see *L2CA_DisconnectReq* in Section 3.1.4) following the receipt of a Disconnection Response from the remote device. An RTX timer expiration (see Section 3.1.5 and *L2CA_TimeOutInd* below) for an outstanding Disconnect Request can substitute for a Disconnect Response and result in this action. Upon receiving this event the upper layer knows the L2CAP channel has been terminated. There is no corresponding negative confirm.

- *L2CA_TimeOutInd*

  Indicates that a RTX or ERTX timer has expired. This indication will occur an implementation-dependant number of times before the L2CAP implementation will give up and send a L2CA_DisconnectInd.

- *L2CA_QoSViolationInd*

  Indicates that the quality of service agreement has been violated.

3.3 CHANNEL OPERATIONAL STATES

- *CLOSED*

  In this state, there is no channel associated with this CID. This is the only state when a link level connection (Baseband*)* may not exist. Link disconnection forces all other states into the CLOSED state.

- *W4_L2CAP_CONNECT_RSP*

  In this state, the CID represents a local end-point and an L2CAP_ConnectReq message has been sent referencing this endpoint and it is now waiting for the corresponding L2CAP_ConnectRsp message.

- *W4_L2CA_CONNECT_RSP*

  In this state, the remote end-point exists and an *L2CAP_ConnectReq* has been received by the local L2CAP entity. An L2CA_ConnectInd has been sent to the upper layer and the part of the local L2CAP entity processing the received *L2CAP_ConnectReq* waits for the corresponding response. The response may require a security check to be performed.

- *CONFIG*

  In this state, the connection has been established but both sides are still negotiating the channel parameters. The Configuration state may also be entered when the channel parameters are being renegotiated. Prior to entering the CONFIG state, all outgoing data traffic should be suspended since the traffic parameters of the data traffic are to be renegotiated. Incoming data traffic must be accepted until the remote channel endpoint has entered the CONFIG state.

  In the CONFIG state, both sides must issue L2CAP_ConfigReq messages – if only defaults are being used, a null message should be sent, see Section 5.4 on page 288. If a large amount of parameters need to be negotiated, multiple messages may be sent to avoid any MTU limitations and negotiate incrementally – see Section 6 on page 297 for more details.

  Moving from the CONFIG state to the OPEN state requires both sides to be ready. An L2CAP entity is ready when it has received a positive response to its final request and it has positively responded to the final request from the remote device.

- *OPEN*

  In this state, the connection has been established and configured, and data flow may proceed.

- *W4_L2CAP_DISCONNECT_RSP*

  In this state, the connection is shutting down and an L2CAP_DisconnectReq message has been sent. This state is now waiting for the corresponding response.

- *W4_L2CA_DISCONNECT_RSP*

  In this state, the connection on the remote endpoint is shutting down and an L2CAP_DisconnectReq message has been received. An L2CA_DisconnectInd has been sent to the upper layer to notify the owner of the CID that the remote endpoint is being closed. This state is now waiting for the corresponding response from the upper layer before responding to the remote endpoint.

3.4 MAPPING EVENTS TO ACTIONS

Table 3.1 defines the actions taken in response to events that occur in a particular state. Events that are not listed in the table, nor have actions marked N/C (for no change), are assumed to be errors and silently discarded.

Data input and output events are only defined for the Open and Configuration states. Data may not be received during the initial Configuration state, but may be received when the Configuration state is re-entered due to a reconfiguration process. Data received during any other state should be silently discarded.

Event	Current State	Action	New State
LP_ConnectCfm	CLOSED	Flag physical link as up and initiate the L2CAP connection.	CLOSED
LP_ConnectCfmNeg	CLOSED	Flag physical link as down and fail any outstanding service connection requests by sending an L2CA_ConnectCfmNeg message to the upper layer.	CLOSED
LP_ConnectInd	CLOSED	Flag link as up.	CLOSED
LP_DisconnectInd	CLOSED	Flag link as down.	CLOSED
LP_DisconnectInd	Any except CLOSED	Send upper layer L2CA_DisconnectInd message.	CLOSED
LP_QoSViolationInd	Any but OPEN	Discard	N/C
LP_QoSViolationInd	OPEN	Send upper layer L2CA_QoSViolationInd message. If service level is guaranteed, terminate the channel.	OPEN or W4_L2CA_DISCONNECT_RSP
L2CAP_ConnectReq	CLOSED. (CID dynamically allocated from free pool.)	Send upper layer L2CA_ConnectInd. Optionally: Send peer L2CAP_ConnectRspPnd	W4_L2CA_CONNECT_RSP
L2CAP_ConnectRsp	W4_L2CAP_CONNECT_RSP	Send upper layer L2CA_ConnectCfm message. Disable RTX timer.	CONFIG
L2CAP_ConnectRspPnd	W4_L2CAP_CONNECT_RSP	Send upper layer L2CA_ConnectPnd message. Disable RTX timer and start ERTX timer.	N/C
L2CAP_ConnectRspNeg	W4_L2CAP_CONNECT_RSP	Send upper layer L2CA_ConnectCfmNeg message. Return CID to free pool. Disable RTX/ERTX timers.	CLOSED
L2CAP_ConfigReq	CLOSED	Send peer L2CAP_Reject message.	N/C
L2CAP_ConfigReq	CONFIG	Send upper layer L2CA_ConfigInd message.	N/C
L2CAP_ConfigReq	OPEN	Suspend data transmission at a convenient point. Send upper layer L2CA_ConfigInd message.	CONFIG

Table 3.1: L2CAP Channel State Machine

Event	Current State	Action	New State
L2CAP_ConfigRsp	CONFIG	Send upper layer L2CA_ConfigCfm message. Disable RTX timer. If an L2CAP_ConfigReq message has been received and positively responded to, then enter OPEN state, otherwise remain in CONFIG state.	N/C or OPEN
L2CAP_ConfigRsp Neg	CONFIG	Send upper layer L2CA_ConfigCfmNeg message. Disable RTX timer.	N/C
L2CAP_Disconnect Req	CLOSED	Send peer L2CAP_DisconnectRsp message.	N/C
L2CAP_Disconnect Req	Any except CLOSED	Send upper layer L2CA_ DisconnectInd message.	W4_L2CA_ DISCONNECT _RSP
L2CAP_Disconnect Rsp	W4_L2CAP_ DISCONNECT_ RSP	Send upper layer L2CA_DisconnectCfm message. Disable RTX timer.	CLOSED
L2CAP_Data	OPEN or CONFIG	If complete L2CAP packet received, send upper layer L2CA_Read confirm.	N/C
L2CA_ConnectReq	CLOSED (CID dynamically allocated from free pool)	Send peer L2CAP_ConnectReq message. Start RTX timer.	W4_L2CAP_ CONNECT_RSP
L2CA_ConnectRsp	W4_L2CA_CONNECT_RSP	Send peer L2CAP_ ConnectRsp message.	CONFIG
L2CA_ConnectRsp Neg	W4_L2CA_CONNECT_RSP	Send peer L2CAP_ConnectRspNeg message. Return CID to free pool.	CLOSED
L2CA_ConfigReq	CLOSED	Send upper layer L2CA_ ConfigCfmNeg message.	N/C
L2CA_ConfigReq	CONFIG	Send peer L2CAP_ConfigReq message. Start RTX timer.	N/C
L2CA_ConfigReq	OPEN	Suspend data transmission at a convenient point. Send peer L2CAP_ConfigReq message. Start RTX timer.	CONFIG

Table 3.1: L2CAP Channel State Machine

Event	Current State	Action	New State
L2CA_ConfigRsp	CONFIG	Send peer L2CAP_ConfigRsp message. If all outstanding L2CAP_ConfigReq messages have received positive responses then move in OPEN state. Otherwise, remain in CONFIG state.	N/C or OPEN
L2CA_ConfigRspNeg	CONFIG	Send peer L2CAP_ ConfigRspNeg message.	N/C
L2CA_Disconnect Req	OPEN or CONFIG	Send peer L2CAP_DisconnectReq message. Start RTX timer.	W4_L2CAP_ DISCONNECT _RSP
L2CA_DisconnectRsp	W4_L2CA_ DISCONNECT_ RSP	Send peer L2CAP_DisconnectRsp message. Return CID to free pool.	CLOSED
L2CA_DataRead	OPEN	If payload complete, transfer payload to InBuffer.	OPEN
L2CA_DataWrite	OPEN	Send peer L2CAP_Data message.	OPEN
Timer_RTX	Any	Send upper layer L2CA_TimeOutInd message. If final expiration, return CID to free pool and go to CLOSE state, else resend Request.	N/C or CLOSED
Timer_ERTX	Any	Send upper layer L2CA_TimeOutInd message. If final expiration, return CID to free pool and go to CLOSE state, else resend Request.	N/C or CLOSED

Table 3.1: L2CAP Channel State Machine

Figure 3.3 illustrates a simplified state machine and typical transition path taken by an initiator and acceptor. The state machine shows what events cause state transitions and what actions are also taken while the transitions occur. Not all the events listed in Table 3.1 are included in the simplified State Machine to avoid cluttering the figure.

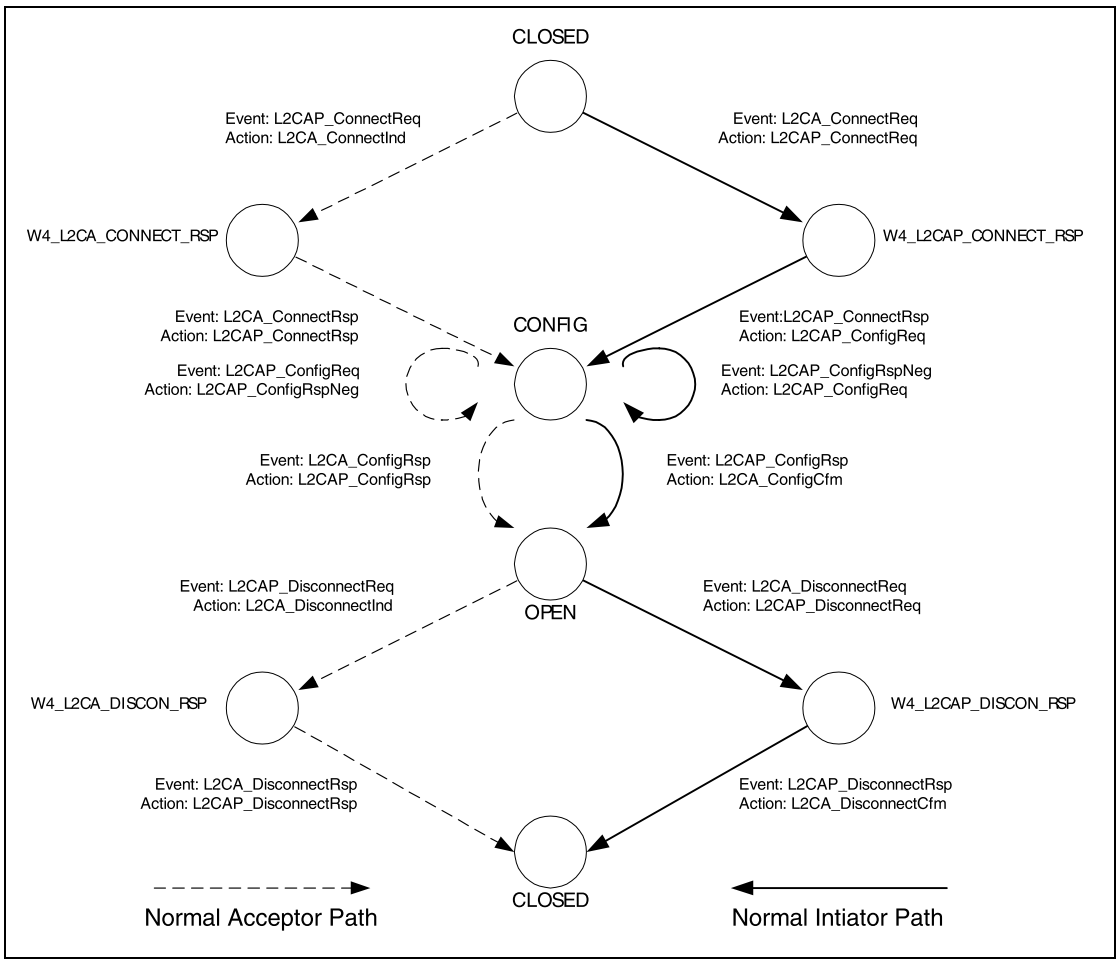


Figure 3.3: State Machine Example

Figure 3.4 presents another illustration of the events and actions based around the messages sequences being communicated between two devices. In this example, the initiator is creating the first L2CAP channel between two devices. Both sides start in the CLOSED state. After receiving the request from the upper layer, the entity requests the lower layer to establish a physical link. If no physical link exists, LMP commands are used to create the physical link between the devices. Once the physical link is established, L2CAP signals may be sent over it.

Figure 3.4 is an example and not all setup sequences will be identical to the one illustrated below.

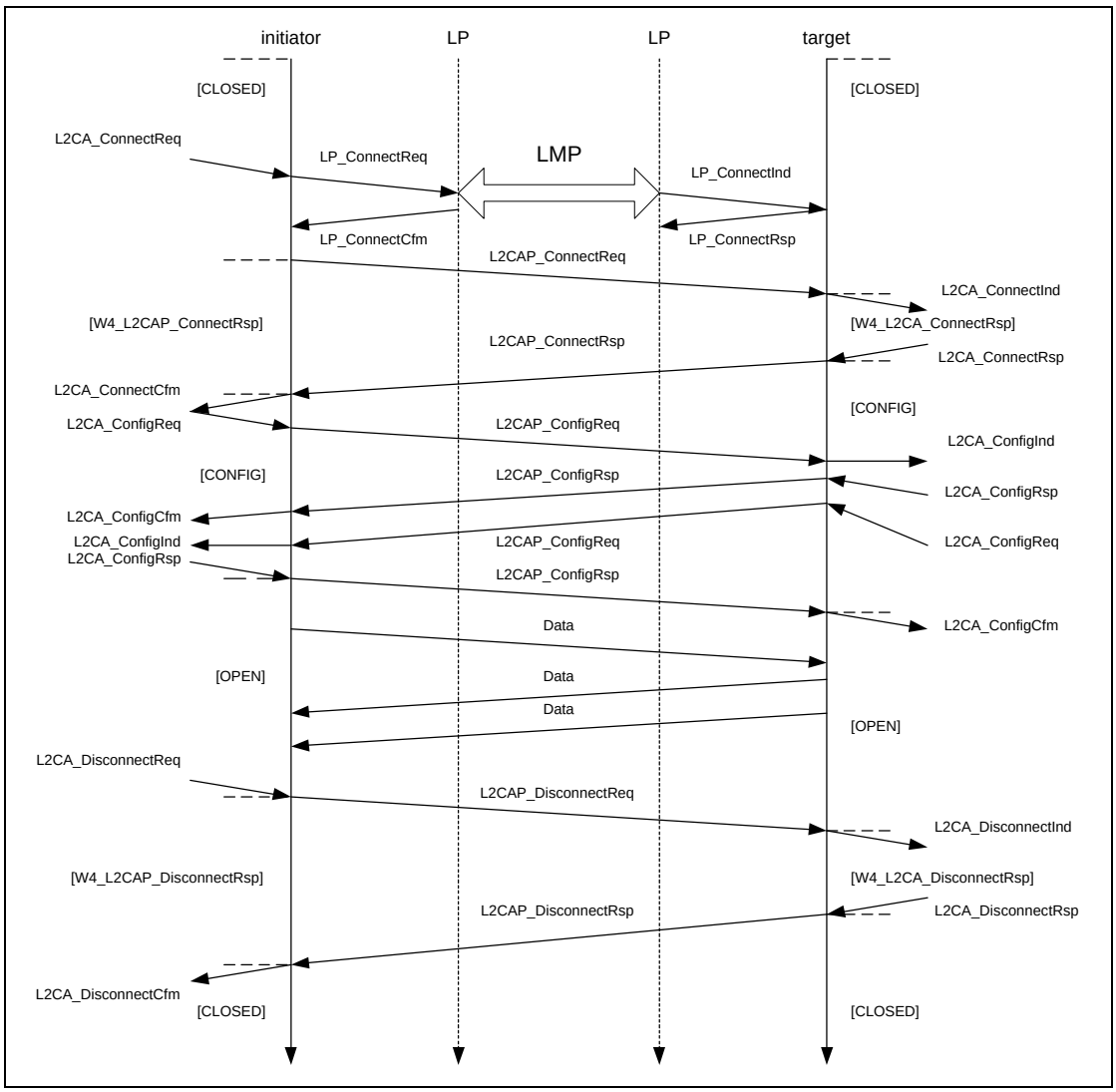


Figure 3.4: Message Sequence Chart of Basic Operation

4 DATA PACKET FORMAT

L2CAP is packet-based but follows a communication model based on *channels*. A channel represents a data flow between L2CAP entities in remote devices. Channels may be connection-oriented or connectionless.
All packet fields use Little Endian byte order.

4.1 CONNECTION-ORIENTED CHANNEL

Figure 4.1 illustrates the format of the L2CAP packet (also referred to as the L2CAP PDU) within a connection-oriented channel.

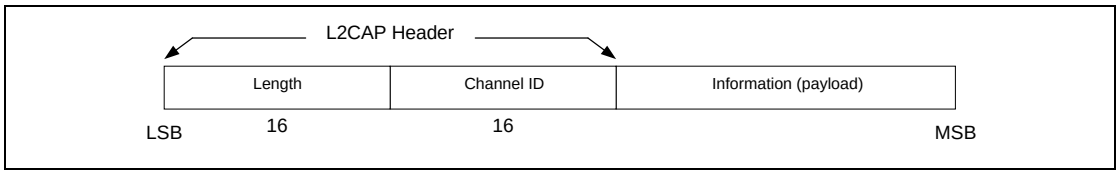


Figure 4.1: L2CAP Packet (field sizes in bits)

The fields shown are:

- *Length: 2 octets (16 bits)*

Length indicates the size of information payload in bytes, excluding the length of the L2CAP header. The length of an information payload can be up to 65535 bytes. The Length field serves as a simple integrity check of the reassembled L2CAP packet on the receiving end.

- *Channel ID: 2 octets*

  The channel ID identifies the destination channel endpoint of the packet. The scope of the channel ID is relative to the device the packet is being sent to.

- *Information: 0 to 65535 octets*

  This contains the payload received from the upper layer protocol (outgoing packet), or delivered to the upper layer protocol (incoming packet). The minimum supported MTU for connection-oriented packets (MTU_{cno}) is negotiated during channel configuration (see Section 6.1 on page 297). The minimum supported MTU for the signalling packet (MTU_{sig}) is 48 bytes (see Section 5 on page 283).

4.2 CONNECTIONLESS DATA CHANNEL

In addition to connection-oriented channels, L2CAP also exports the concept of a group-oriented channel. Data sent to the 'group' channel is sent to all members of the group in a best-effort manner. Groups have no quality of service associated with them. Group channels are unreliable; L2CAP makes no guarantee that data sent to the group successfully reaches all members of the group. If reliable group transmission is required, it must be implemented at a higher layer.

Transmissions to a group must be non-exclusively sent to all members of that group. The local device cannot be a member of the group, and higher layer protocols are expected to loopback any data traffic being sent to the local device. Non-exclusive implies non-group members may receive group transmissions and higher level (or link level) encryption can be used to support private communication.

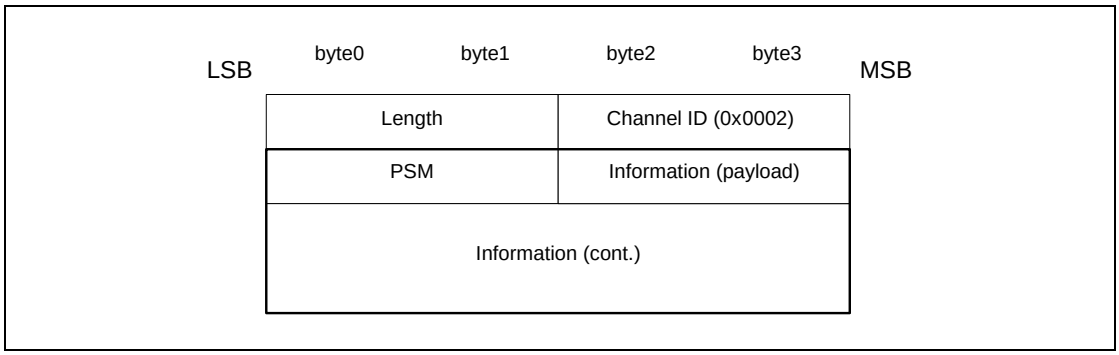


Figure 4.2: Connectionless Packet

The fields shown are:

- *Length: 2 octets*

  Length indicates the size of information payload plus the PSM field in bytes, excluding the length of the L2CAP header.

- *Channel ID: 2 octets*

  Channel ID (0x0002) reserved for connectionless traffic.

- *Protocol/Service Multiplexer (PSM): 2 octets (minimum)*

  The PSM field is based on the ISO 3309 extension mechanism for address fields. All content of the PSM field, referred to as the PSM value, must be ODD, that is, the least significant bit of the least significant octet must be '1'. Also, all PSM values must be assigned such that the least significant bit of the most significant octet equals '0'. This allows the PSM field to be extended beyond 16 bits. The PSM value definitions are specific to L2CAP and assigned by the Bluetooth SIG. For more information on the PSM field see Section 5.2 on page 286.

- *Information: 0 to 65533 octets*

  The payload information to be distributed to all members of the group. Implementations must support a minimum connectionless MTU (MTU_{cnl}) of 670 octets, unless explicitly agreed upon otherwise, e.g., for single operation devices that are built to comply to a specific Bluetooth profile that dictates the use of a specific MTU for connectionless traffic that is less than MTU_{cnl}.

The L2CAP group service interface provides basic group management mechanisms including creating a group, adding members to a group, and removing members from a group. There are no pre-defined groups such as ‘all radios in range’.

5 SIGNALLING

This section describes the signalling commands passed between two L2CAP entities on remote devices. All signalling commands are sent to CID 0x0001. The L2CAP implementation must be able to determine the Bluetooth address (BD_ADDR) of the device that sent the commands. Figure 5.1 illustrates the general format of all L2CAP packets containing signalling commands. Multiple commands may be sent in a single (L2CAP) packet and packets are sent to CID 0x0001. Commands take the form of Requests and Responses. All L2CAP implementations must support the reception of signalling packets whose MTU (MTU_{sig}) does not exceed 48 bytes. L2CAP implementations should not use signalling packets beyond this size without first testing whether the implementation can support larger signalling packets. Implementations must be able to handle the reception of multiple commands in an L2CAP packet as long as the MTU is not exceeded.

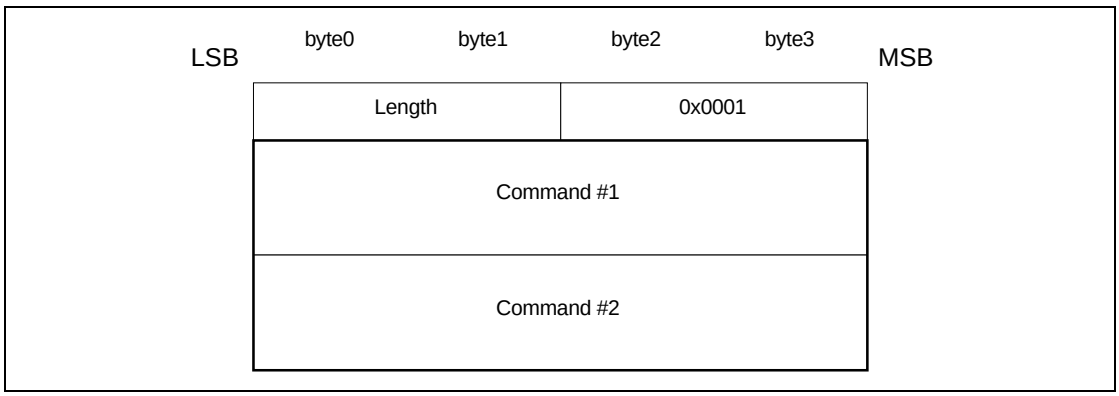


Figure 5.1: Signalling Command Packet Format

Figure 5.2 displays the general format of all signalling commands.

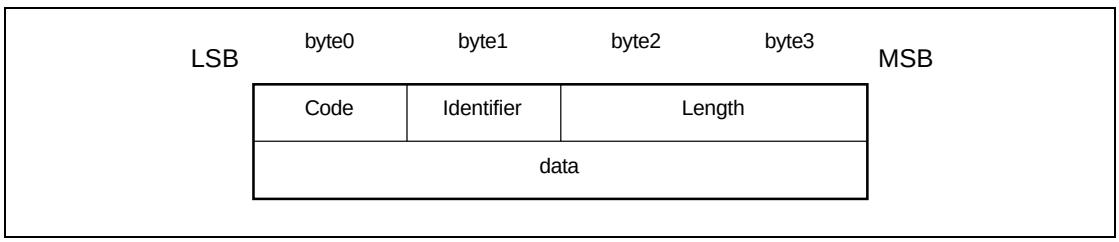


Figure 5.2: Command format

The fields shown are:

- *Code: 1 octet*

  The Code field is one octet long and identifies the type of command. When a packet is received with an unknown Code field, a Command Reject packet (defined in Section 5.1 on page 285) is sent in response.

  Up-to-date values of assigned Codes are specified in the latest Bluetooth 'Assigned Numbers' document. Table 5.1 on pag e284 lists the codes

defined by this document. All codes are specified with the most significant bit in the left-most position.

Code	Description
0x00	RESERVED
0x01	Command reject
0x02	Connection request
0x03	Connection response
0x04	Configure request
0x05	Configure response
0x06	Disconnection request
0x07	Disconnection response
0x08	Echo request
0x09	Echo response
0x0A	Information request
0x0B	Information response

Table 5.1: Signalling Command Codes

- *Identifier: 1 octet*

  The Identifier field is one octet long and helps matching a request with the reply. The requesting device sets this field and the responding device uses the same value in its response. A different Identifier must be used for each original command. Identifiers should not be recycled until a period of 360 seconds has elapsed from the initial transmission of the command using the identifier. On the expiration of a RTX or ERTX timer, the same identifier should be used if a duplicate Request is re-sent as stated in Section 3.1.5 on page 270. A device receiving a duplicate request should reply with a duplicate response. A command response with an invalid identifier is silently discarded. Signalling identifier 0x0000 is defined to be an illegal identifier and shall never be used in any command.

- *Length: 2 octets*

  The Length field is two octets long and indicates the size in octets of the data field of the command only, i.e., it does not cover the Code, Identifier, and Length fields.

- *Data: 0 or more octets*

  The Data field is variable in length and discovered using the Length field. The Code field determines the format of the Data field.

5.1 COMMAND REJECT (CODE 0x01)

A Command Reject packet is sent in response to a command packet with an unknown command code or when sending the corresponding Response is inappropriate. Figure 5.3 displays the format of the packet. The Identifier should match the Identifier of the packet containing the unidentified code field. Implementations must always send these packets in response to unidentified signalling packets. Command Reject packets should not be sent in response to an identified Response packet.

When multiple commands are included in an L2CAP packet and the packet exceeds the MTU of the receiver, a single Command Reject packet is sent in response. The identifier should match the first Request command in the L2CAP packet. If only Responses are recognized, the packet shall be silently discarded.

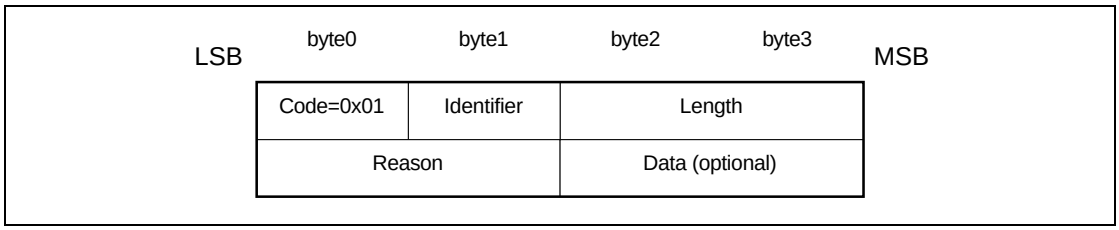


Figure 5.3: Command Reject Packet

- *Length = 0x0002 or more octets*
- *Reason: 2 octets*

  The Reason field describes why the Request packet was rejected.

Reason value	Description
0x0000	Command not understood
0x0001	Signalling MTU exceeded
0x0002	Invalid CID in request
Other	Reserved

Table 5.2: Reason Code Descriptions

- *Data: 0 or more octets*

  The length and content of the Data field depends on the Reason code. If the Reason code is 0x0000, “Command not understood”, no Data field is used. If the Reason code is 0x0001, “Signalling MTU Exceeded”, the 2-octet Data field represents the maximum signalling MTU the sender of this packet can accept.

  If a command refers to an invalid channel then the Reason code 0x0002 will be returned. Typically a channel is invalid because it does not exist. A 4-octet data field on the command reject will contain the local (first) and remote (second) channel endpoints (relative to the sender of the Command Reject) of the disputed channel. The latter endpoints are obtained from the corresponding rejected command. If the rejected command contains only

one of the channel endpoints, the other one is replaced by the null CID 0x0000.

Reason value	Data Length	Data value
0x0000	0 octets	N/A
0x0001	2 octets	Actual MTU
0x0002	4 octets	Requested CID

Table 5.3: Reason Data values

5.2 CONNECTION REQUEST (CODE 0x02)

Connection request packets are sent to create a channel between two devices. The channel connection must be established before configuration may begin. Figure 5.4 illustrates a Connection Request packet.

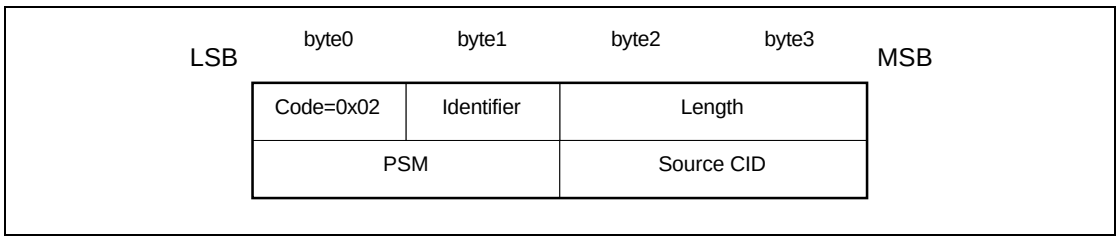


Figure 5.4: Connection Request Packet

- *Length = 0x0004 or more octets*
- *Protocol/Service Multiplexor (PSM): 2 octets (minimum)*

  The PSM field is two octets (minimum) in length. The structure of the PSM field is based on the ISO 3309 extension mechanism for address fields. All PSM values must be ODD, that is, the least significant bit of the least significant octet must be ’1’. Also, all PSM values must be assigned such that the least significant bit of the most significant octet equals ’0’. This allows the PSM field to be extended beyond 16 bits. PSM values are separated into two ranges. Values in the first range are assigned by the Bluetooth SIG and indicate protocols. The second range of values are dynamically allocated and used in conjunction with the Service Discovery Protocol (SDP). The dynamically assigned values may be used to support multiple implementations of a particular protocol

PSM value	Description
0x0001	Service Discovery Protocol
0x0003	RFCOMM
0x0005	Telephony Control Protocol
<0x1000	RESERVED

Table 5.4: Defined PSM Values

[0x1001-0xFFFF]	DYNAMICALLY ASSIGNED

Table 5.4: Defined PSM Values

- *Source CID (SCID): 2 octets*

  The source local CID is two octets in length and represents a channel end-point on the device sending the request. Once the channel has been configured, data packets flowing tothe sender of the request must be send to this CID. In this section, the Source CID represents the channel endpoint on the device sending the request and receiving the response.

5.3 CONNECTION RESPONSE (CODE 0x03)

When a unit receives a Connection Request packet, it must send a Connection Response packet. The format of the connection response packet is shown in Figure 5.5.

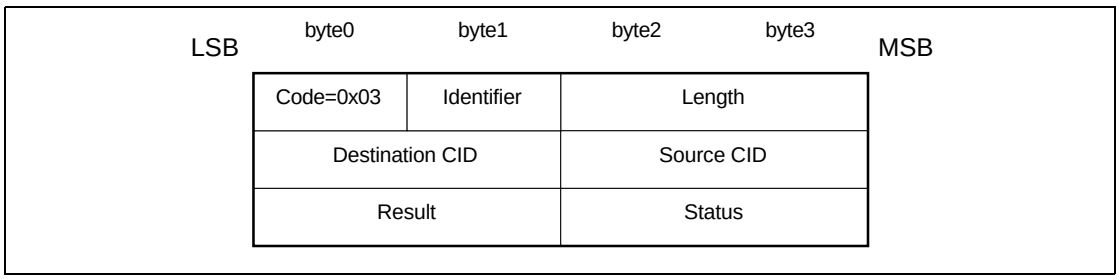


Figure 5.5: Connection Response Packet

- *Length = 0x0008 octets*
- *Destination Channel Identifier (DCID): 2 octets*

  The field contains the channel end-point on the device sending this Response packet. In this section, the Destination CID represents the channel endpoint on the device receiving the request and sending the response.
- *Source Channel Identifier (SCID): 2 octets*

  The field contains the channel end-point on the device receiving this Response packet.

- *Result: 2 octets*

  The result field indicates the outcome of the connection request. The result value of 0x0000 indicates success while a non-zero value indicates the connection request failed or is pending. A logical channel is established on the receipt of a successful result. Table 5.5 defines values for this field. If the result field is not zero. The DCID and SCID fields should be ignored when the result field indicates the connection was refused.

Value	Description
0x0000	Connection successful.
0x0001	Connection pending
0x0002	Connection refused – PSM not supported.
0x0003	Connection refused – security block.
0x0004	Connection refused – no resources available.
Other	Reserved.

Table 5.5: Result values

- *Status: 2 octets*

  Only defined for Result = Pending. Indicates the status of the connection.

Value	Description
0x0000	No further information available
0x0001	Authentication pending
0x0002	Authorization pending
Other	Reserved

Table 5.6: Status values

5.4 CONFIGURATION REQUEST (CODE 0x04)

Configuration Request packets are sent to establish an initial logical link transmission contract between two L2CAP entities and also to re-negotiate this contract whenever appropriate. During a re-negotiation session, all data traffic on the channel should be suspended pending the outcome of the negotiation. Each configuration parameter in a Configuration Request is related exclusively either with the outgoing or the incoming data traffic but not both of them. In Section 6 on page 297, the various configuration parameters and their relation to the outgoing or incoming data traffic are presented. If an L2CAP entity receives a Configuration Request while it is waiting for a response it must not block sending the Configuration Response, otherwise the configuration process may deadlock.

If no parameters need to be negotiated, no options need to be inserted and the C-bit should be cleared. L2CAP entities in remote devices MUST negotiate all

parameters defined in this document whenever the default values are not acceptable. Any missing configuration parameters are assumed to have their most recently (mutually) explicitly or implicitly accepted values. Event if all default values are acceptable, a Configuration Request packet with no options MUST be sent. Implicitly accepted values are any default values for the configuration parameters specified in this document that have not been explicitly negotiated for the specific channel under configuration.

Each configuration parameter is one-directional and relative to the direction implied by the sender of a Configuration Request. If a device needs to establish the value of a configuration parameter in the opposite direction than the one implied by a Configuration Request, a new Configuration Request with the desired value of the configuration parameter in it needs to be sent in the direction opposite the one used for the original ConfigurationRequest.

The decision on the amount of time (or messages) spent arbitrating the channel parameters before terminating the negotiation is left to the implementation but it shall not last more than 120 seconds.

Figure 5.6 defines the format of the Configuration Request packet.

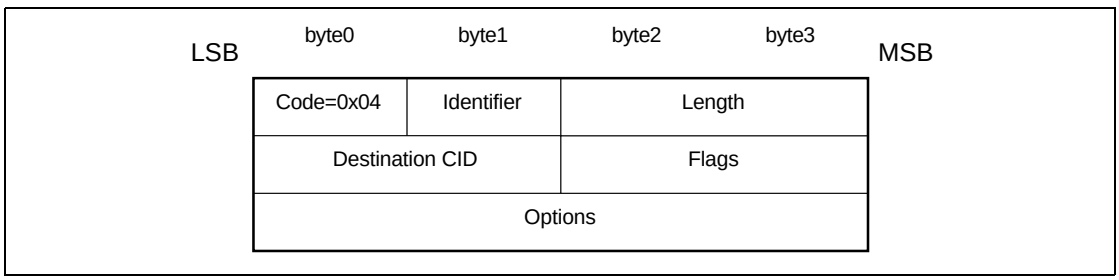


Figure 5.6: Configuration Request Packet

- *Length = 0x0004 or more octets*
- *Destination CID (DCID): 2 octets*

  The field contains the channel end-point on the device receiving this Request packet.
- *Flags: 2 octets*

  Figure 5.7 display the two-octet Flags field. Note the most significant bit is shown on the left.

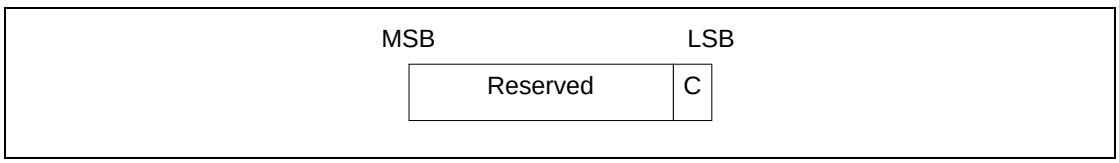


Figure 5.7: Configuration Request Flags field format

C - continuation flag. When all configuration options cannot fit into the receiver's MTU_{sig}, the options must be passed in multiple configuration command packets. If all options fit into the receiver's MTU, then the continuation bit should not be used. Each Configuration Request should contain in integral number of options - partially formed options must not be sent. Each Request will be tagged with a different Identifier and be matched with a Response with the same Identifier.

When used in the Configuration Request, the continuation flag indicates the responder should expect to receive multiple request packets. The responder must reply to each request packet. The responder may reply to each Configuration Request with a Configuration Response containing the same option(s) present in the Request, except for those error conditions more appropriate for a Command Reject, or the responder may reply with a "Success" Configuration Response packet containing no options, delaying those options until the full Request has been received. The Configuration Request packet with the configuration flag cleared shall be treated as the Configuration Request event in the channel state machine.

When used in the Configuration Response, the continuation flag must be set if the flag is set in the Request. If the configuration flag is set in the Response when the matching Request does not set the flag, it indicates the responder has additional options to send to the requestor. In this situation, the requestor shall send null-option Configuration Requests (with cleared C-flag) to the responder until the responder replies with a Configuration Response where the continuation flag is clear. The Configuration Response packet with the configuration flag cleared shall be treated as the Configuration Response event in the channel state machine.

The result of the configuration transaction is the union of all the result values. All the result values must succeed for the configuration transaction to succeed.

Other flags are reserved and should be cleared. L2CAP implementations should ignore these bits.

- *Configuration Options*

  The list of the parameters and their values to be negotiated. These are defined in Section 6 on page 297. Configuration Requests may contain no options (referred to as an empty or null configuration request) and can be used to request a response. For an empty configuration request the length field is set to 0x0004.

5.5 CONFIGURE RESPONSE (CODE 0X05)

Configure Response packets MUST be sent in reply to Configuration Request packets except when the error condition id covered by a Command Reject response. Each configuration parameter value (if any is present) in a Configuration Response reflects an ’adjustment’ to a configuration parameter value that has been sent (or, in case of default values, implied) in the corresponding Configuration Request. Thus, for example, if a configuration parameter in a

Configuration Request relates to traffic flowing from device A to device B, the sender of the Configuration Response will only adjust (if needed) this value again for the same traffic flowing from device A to device B. The options sent in the Response depend on the value in the Result field. Figure 5.8 defines the format of the Configuration Response packet.

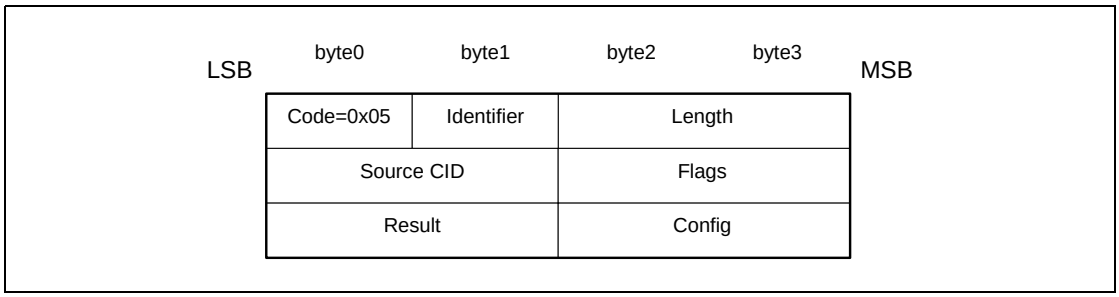


Figure 5.8: Configuration Response Packet

- *Length = 0x0006 or more octets*
- *Source CID (SCID): 2 octets*

  The field contains the channel end-point on the device receiving this Response packet. The device receiving the Response must check that the Identifier field matches the same field in the corresponding configuration request command and the SCID matches its local CID paired with the original DCID.
- *Flags: 2 octets*

  Figure 5.9 displays the two-octet Flags field. Note the most significant bit is shown on the left.

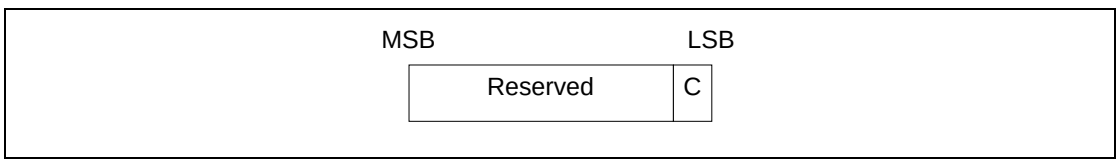


Figure 5.9: Configuration Response Flags field format

C – more configuration responses will follow when set to 1. This flag indicates that the parameters included in the response are a partial subset of parameters being sent by the device sending the Response packet.

Other flags are reserved and should be cleared. L2CAP implementations should ignore these bits.

- *Result: 2 octets*

  The Result field indicates whether or not the Request was acceptable. See Table 5.7 for possible result codes.

Result	Description
0x0000	Success
0x0001	Failure – unacceptable parameters
0x0002	Failure – rejected (no reason provided)
0x0003	Failure – unknown options
Other	RESERVED

Table 5.7: Configuration Response Result codes

- *Configuration Options*

  This field contains the list of parameters being negotiated. These are defined in Section 6 on page 297. On a successful result, these parameters contain the return values for any wild card parameters (see Section 6.3 on page 299) contained in the request.

  On an unacceptable parameters failure (Result=0x0001) the rejected parameters should be sent in the response with the values that would have been accepted if sent in the original request. Any missing configuration parameters are assumed to have their most recently (mutually) accepted values and they too can be included in the Configuration Response if need to be changed. Recall that, each configuration parameter is one-directional and relative to the direction implied by the sender of a Configuration Request. Thus, if the sender of the Configuration Response needs to establish the value of a configuration parameter in the opposite direction than the one implied by an original Configuration Request, a new Configuration Request with the desired value of the configuration parameter in it needs to be sent in the direction opposite the one used for the original Connection Request.

  On an unknown option failure (Result=0x0003), the option types not understood by the recipient of the Request must be included in the Response. Note that hints (defined in Section 6 on page 297), those options in the Request that are skipped if not understood, must not be included in the Response and must not be the sole cause for rejecting the Request.

  The decision on the amount of time (or messages) spent arbitrating the channel parameters before terminating the negotiation is left to the implementation.

5.6 DISCONNECTION REQUEST (CODE 0x06)

Terminating an L2CAP channel requires that a disconnection request packet be sent and acknowledged by a disconnection response packet. Disconnection is requested using the signalling channel since all other L2CAP packets sent to the destination channel automatically get passed up to the next protocol layer. Figure 5.10 displays a disconnection packet request. The receiver must ensure both source and destination CIDs match before initiating a connection disconnection. Once a Disconnection Request is issued, all incoming data in transit on this L2CAP channel will be discarded and any new additional outgoing data is not allowed. Once a disconnection request for a channel has been received, all data queued to be sent out on that channel may be discarded.

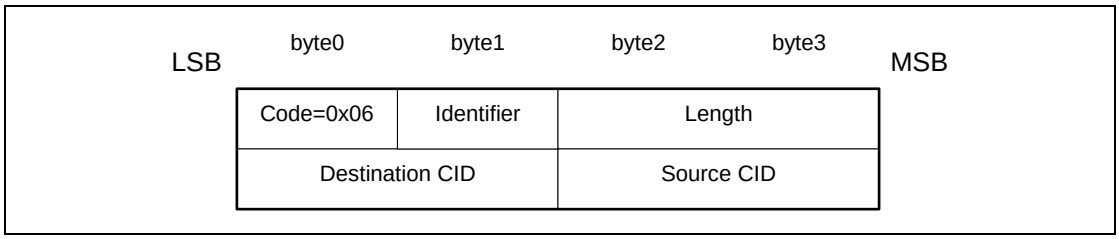


Figure 5.10: Disconnection Request Packet

- *Length = 0x0004 octets*
- *Destination CID (DCID): 2 octets*

  This field specifies the end-point of the channel to be shutdown on the device receiving this request.
- *Source CID (SCID): 2 octets*

  This field specifies the end-point of the channel to be shutdown on the device sending this request.

  The SCID and DCID are relative to the sender of this request and must match those of the channel to be disconnected. If the DCID is not recognized by the receiver of this message, a CommandReject message with 'invalid CID' result code must be sent in response. If the receivers finds a DCID match but the SCID fails to find the same match, the request should be silently discarded.

5.7 DISCONNECTION RESPONSE (CODE 0x07)

Disconnection responses should be sent in response to each disconnection request.

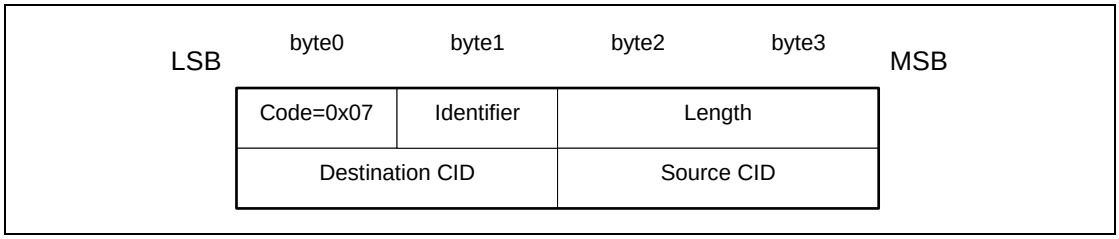


Figure 5.11: Disconnection Response Packet

- *Length = 0x0004 octets*
- *Destination CID (DCID): 2 octets*

  This field identifies the channel end-point on the device sending the response.
- *Source CID (SCID): 2 octets*

  This field identifies the channel end-point on the device receiving the response.

  The DCID and the SCID (which are relative to the sender of the request), and the Identifier fields must match those of the corresponding disconnection request command. If the CIDs do not match, the response should be silently discarded at the receiver.

5.8 ECHO REQUEST (CODE 0x08)

Echo requests are used to solicit a response from a remote L2CAP entity. These requests may be used for testing the link or passing vendor specific information using the optional data field. L2CAP entities MUST respond to well-formed Echo Request packets with an Echo Response packet. The Data field is optional and implementation-dependent. L2CAP entities should ignore the contents of this field.

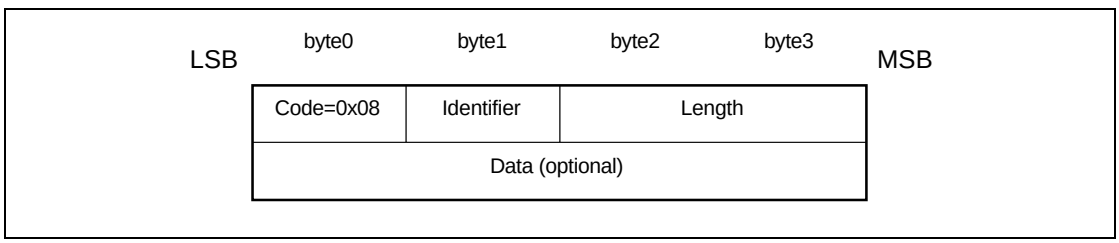


Figure 5.12: Echo Request Packet

5.9 ECHO RESPONSE (CODE 0x09)

Echo responses are sent upon receiving Echo Request packets. The identifier in the response MUST match the identifier sent in the Request. The optional and implementation-dependent data field may contain the contents of the data field in the Request, different data, or no data at all.

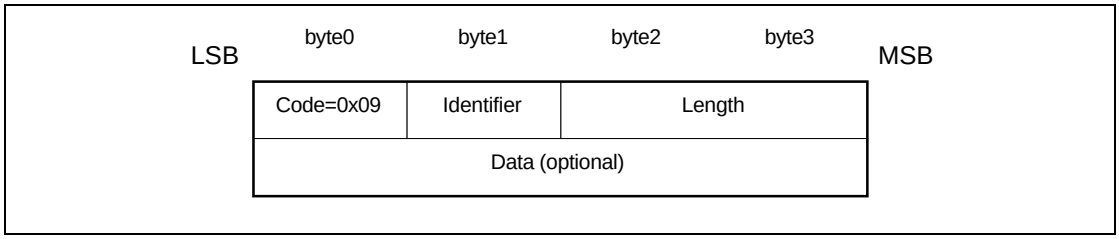


Figure 5.13: Echo Response Packet

5.10 INFORMATION REQUEST (CODE 0X0A)

Information requests are used to solicit implementation-specific information from a remote L2CAP entity. L2CAP entities MUST respond to well-formed Information Request packets with an Information Response packet.

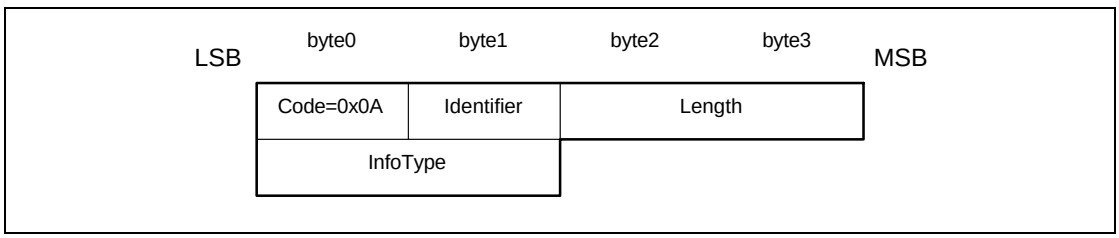


Figure 5.14: Information Request Packet

- *Length = 0x0002 octets*
- *InfoType: 2 octets*

  The InfoType defines the type of implementation-specific information being solicited.

Value	Description
0x0001	Connectionless MTU
Other	Reserved

Table 5.8: InfoType definitions

5.11 INFORMATION RESPONSE (CODE 0X0B)

Information responses are sent upon receiving Information Request packets. The identifier in the response MUST match the identifier sent in the Request. The optional data field may contain the contents of the data field in the Request, different data, or no data at all.

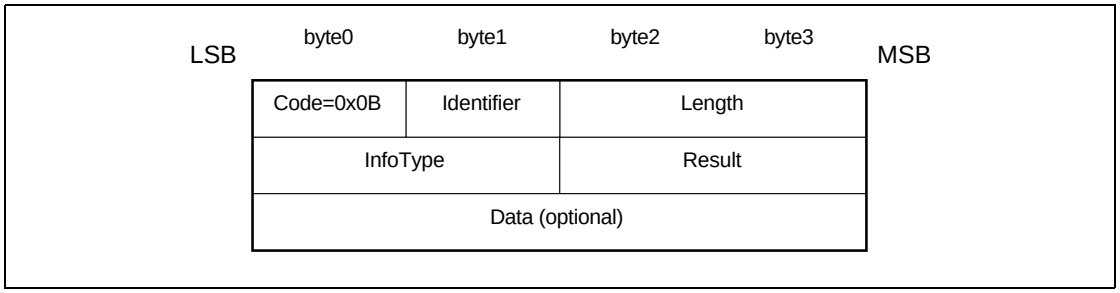


Figure 5.15: Information Response Packet

- *InfoType: 2 octets*

  Same value sent in the request.

- *Result: 2 octets*

  The Result contains information about the success of the request. If result is "Success", the data field contains the information as specified in Table 5.10. If result is "Not supported", no data should be returned.

Value	Description
0x0000	Success
0x0001	Not supported
Other	Reserved

Table 5.9: Information Response Result values

- *Data: 0 or more octets*

  The contents of the Data field depends on the InfoType. For the Connection MTU request, the data field contains the remote entity's 2-octet acceptable connectionless MTU.

InfoType	Data	Data Length (in octets)
0x0001	Connectionless MTU	2

Table 5.10: Information Response Data fields

6 CONFIGURATION PARAMETER OPTIONS

Options are a mechanism to extend the ability to negotiate different connection requirements. Options are transmitted in the form of information elements comprised an option type, an option length, and one or more option data fields. Figure 6.1 illustrates the format of an option.

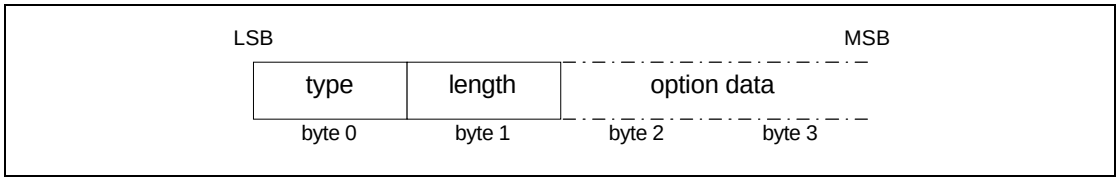


Figure 6.1: Configuration option format

- *Type: 1 octet*

  The option type field defines the parameters being configured. The most significant bit of the type determines the action taken if the option is not recognized. The semantics assigned to the bit are defined below.

  0 - option must be recognized; refuse the configuration request

  1 - option is a hint; skip the option and continue processing

- *Length: 1 octet*

  The length field defines the number of octets in the option payload. So an option type with no payload has a length of 0.

- *Option data*

  The contents of this field are dependent on the option type.

6.1 MAXIMUM TRANSMISSION UNIT (MTU)

This option specifies the payload size the sender is capable of accepting. The type is 0x01, and the payload length is 2 bytes, carrying the two-octet MTU size value as the only information element (see Figure 6.2 on page298).

Since all L2CAP implementations are capable to support a minimum L2CAP packet size, see Section 4 on page 280, MTU is not really a negotiated value but rather an informational parameter to the remote device that the local device can accommodate in this channel an MTU larger than the minimum required. In the unlikely case that the remote device is only willing to send L2CAP packets in this channel that are larger than the MTU announced by the local device, then this Configuration Request will receive a negative response in which the remote device will include the value of MTU that is indented to transmit. In this case, it is implementation specific on whether the local device will continue the configuration process or even maintain this channel.

The remote device in its positive Configuration Response will include the actual MTU to be used on this channel for traffic flowing into the local device which is

minimum{ MTU in configReq, outgoing MTU capability of remote device }. The MTU to be used on this channel but for the traffic flowing in the opposite direction will be established when the remote device (with respect to this discussion) sends its own Configuration Request as explained in Section 5.4 on page 288.

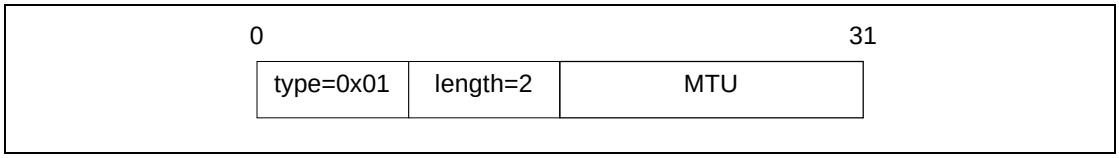


Figure 6.2: MTU Option Format

- Maximum Transmission Unit (MTU) Size: 2 octets

  The MTU field represents the largest L2CAP packet payload, in bytes, that the originator of the Request can accept for that channel. The MTU is asymmetric and the sender of the Request shall specify the MTU it can receive on this channel if it differs from the default value. L2CAP implementations must support a minimum MTU size of 48 bytes. The default value is 672 bytes[1].

6.2 FLUSH TIMEOUT OPTION

This option is used to inform the recipient of the amount of time the originator's link controller / link manager will attempt to successfully transmit an L2CAP segment before giving up and flushing the packet. The type is 0x02 and the payload size is 2 octets.

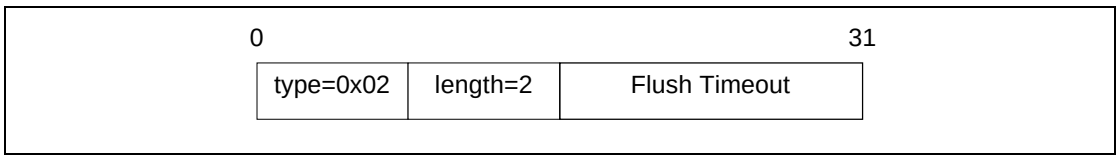


Figure 6.3: Flush Timeout

- *Flush Timeout*

  This value represents units of time measured in milliseconds. The value of 1 implies no retransmissions at the Baseband level should be performed since the minimum polling interval is 1.25 ms. The value of all 1's indicates an infinite amount of retransmissions. This is also referred to as 'reliable channel'. In this case, the link manager shall continue retransmitting a segment until physical link loss occurs. This is an asymmetric value and the sender of the Request shall specify its flush timeout value if it differs from the default value of 0xFFFF.

1. The default MTU was selected based on the payload carried by two Baseband DH5 packets (2*341=682) minus the Baseband ACL headers (2*2=4) and L2CAP header (6).

6.3 QUALITY OF SERVICE (QOS) OPTION

This option specifies a flow specification (flowSpec) similar to RFC 1363 [1]. If no QoS configuration parameter is negotiated the link should assume the default parameters discussed below. The QoS option is type 0x03.

When included in a Configuration Request, this option describes the outgoing traffic flow from the device sending the request to the device receiving it. When included in a positive Configuration Response, this option describes the incoming traffic flow agreement as seen from the device sending the response. When included in a negative Configuration Response, this option describes the preferred incoming traffic flow from the perspective of the device sending the response.

L2CAP implementations are only required to support 'Best Effort' service, support for any other service type is optional. Best Effort does not require any guarantees. If no QoS option is placed in the request, Best Effort must be assumed. If any QoS guarantees are required then a QoS configuration request must be sent.

The remote device places information that depends on the value of the result field, see Section 5.5 on page 290, in its Configuration Response. If the request was for Guaranteed Service, the response shall include specific values for any wild card parameters (see Token Rate and Token Bucket Size descriptions) contained in the request. If the result is "Failure – unacceptable parameters", the response may include a list of outgoing flowspec parameters and parameter values that would make a new Connection Request from the local device acceptable by the remote device. Both explicitly referenced in a Configuration Request or implied configuration parameters can be included in a Configuration Response. Recall that any missing configuration parameters from a Configuration Request are assumed to have their most recently (mutually) accepted values. For both Best effort and Guaranteed service, when the QoS option appears in the Configuration Response, "do not cares" shall be present where they appeared in the Configuration Request.

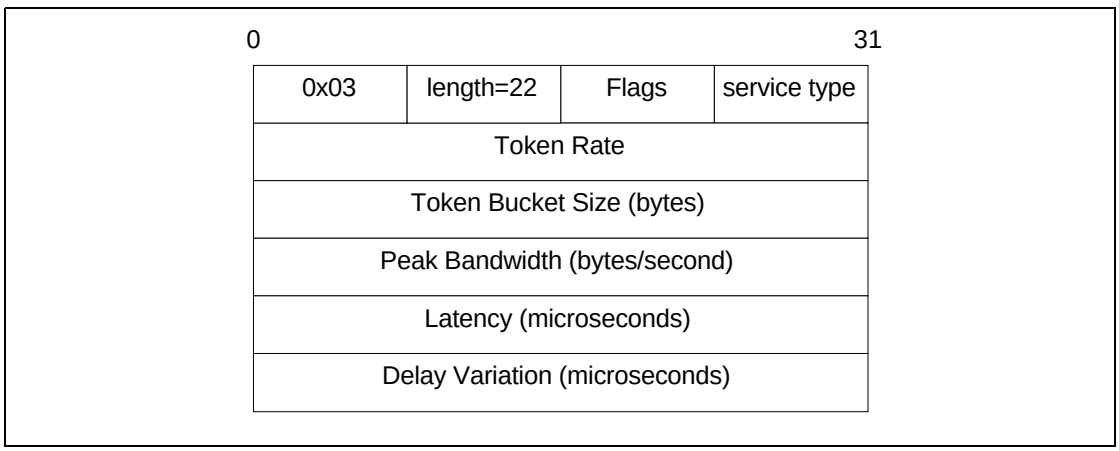


Figure 6.4: Quality of Service Flow Specification

- *Flags: 1 octet*

  Reserved for future use and must be set to 0.

- *Service Type: 1 octet*

  This field indicates the level of service required. Table 6.1 defines the different services available. If 'No traffic' is selected, the remainder of the fields may be ignored because there is no data being sent across the channel in the outgoing direction.

  If 'Best effort', the default value, is selected, the remaining fields should be treated as hints by the remote device. The remote device may choose to ignore the fields, try to satisfy the hint but provide no response (QoS option omitted in the Response message), or respond with the settings it will try to meet.

Value	Description
0x00	No traffic
0x01	Best effort (Default)
0x02	Guaranteed
Other	Reserved

Table 6.1: Service type definitions

- *Token Rate: 4 octets*

  The value of this field represents the rate at which traffic credits are granted in bytes per second. An application may send data at this rate continuously. Burst data may be sent up to the token bucket size (see below). Until that data burst has been drained, an application must limit itself to the token rate. The value 0x00000000 indicates no token rate is specified. This is the default value and implies indifference to token rate. The value 0xFFFFFFFF represents a wild card matching the maximum token rate available. The meaning of this value depends on the semantics associated with the service type. For best effort, the value is a hint that the application wants as much bandwidth as possible. For Guaranteed service the value represents the maximum bandwidth available at the time of the request.

- *Token Bucket Size: 4 octets*

  The value of this field represents the size of the token bucket in bytes. If the bucket is full, then applications must either wait or discard data. The value of 0x00000000 represents no token bucket is needed; this is the default value. The value 0xFFFFFFFF represents a wild card matching the maximum token bucket available. The meaning of this value depends on the semantics associated with the service type. For best effort, the value indicates the application wants a bucket as big as possible. For Guaranteed service the value represents the maximum buffer space available at the time of the request.

- *Peak Bandwidth: 4 octets*

  The value of this field, expressed in bytes per second, limits how fast packets may be sent back-to-back from applications. Some intermediate systems can take advantage of this information, resulting in more efficient resource allocation. The value of 0x00000000 states that the maximum bandwidth is unknown, which is the default value.

- *Latency: 4 octets*

  The value of this field represents the maximum acceptable delay between transmission of a bit by the sender and its initial transmission over the air, expressed in microseconds. The precise interpretation of this number depends on the level of guarantee specified in the Class of Service. The value 0xFFFFFFFF represents a do not care and is the default value.

- *Delay Variation: 4 octets*

  The value of this field is the difference, in microseconds, between the maximum and minimum possible delay that a packet will experience. This value is used by applications to determine the amount of buffer space needed at the receiving side in order to restore the original data transmission pattern. The value 0xFFFFFFFF represents a do not care and is the default value.

6.4 CONFIGURATION PROCESS

Negotiating the channel parameters involves three steps:

1. Informing the remote side of the non-default parameters that the local side will accept using a Configuration Request
2. Remote side responds, agreeing or disagreeing to these values, including the default ones, using a Configuration Response.The local and remote devices repeat steps (1) and (2) as needed.
3. Repeat steps (1) and (2) exactly once more for the reverse direction.

This process can be abstracted into the initial Request negotiation path and a Response negotiation path, followed by the reverse direction phase. Reconfiguration follows a similar two-phase process by requiring negotiation in both directions.

6.4.1 Request Path

The Request Path negotiates the incoming MTU, flush timeout, and outgoing flowspec. Table 6.2 defines the configuration options that may be placed in the Configuration Request message and their semantics.

Parameter	Description
MTU	Incoming MTU information
FlushTO	Outgoing flush timeout
OutFlow	Outgoing flow information.

Table 6.2: Parameters allowed in Request

6.4.2 Response Path

The Response Path negotiates the outgoing MTU (remote side's incoming MTU), the remote side's flush timeout, and incoming flowspec (remote side's outgoing flowspec). If a request-oriented parameter is not present in the Request message (reverts to default value), the remote side may negotiate for a non-default value by including the proposed value in a negative Response message.

Parameter	Description
MTU	Outgoing MTU information
FlushTO	Incoming flush timeout
InFlow	Incoming flow information

Table 6.3: Parameters allowed in Response

6.4.3 Configuration State Machine

The configuration state machine shown below depicts two paths. Before leaving the CONFIG state and moving into the OPEN state, both paths must reach closure. The request path requires the local device to receive a positive response to reach closure while the response path requires the local device to send a positive response to reach closure.

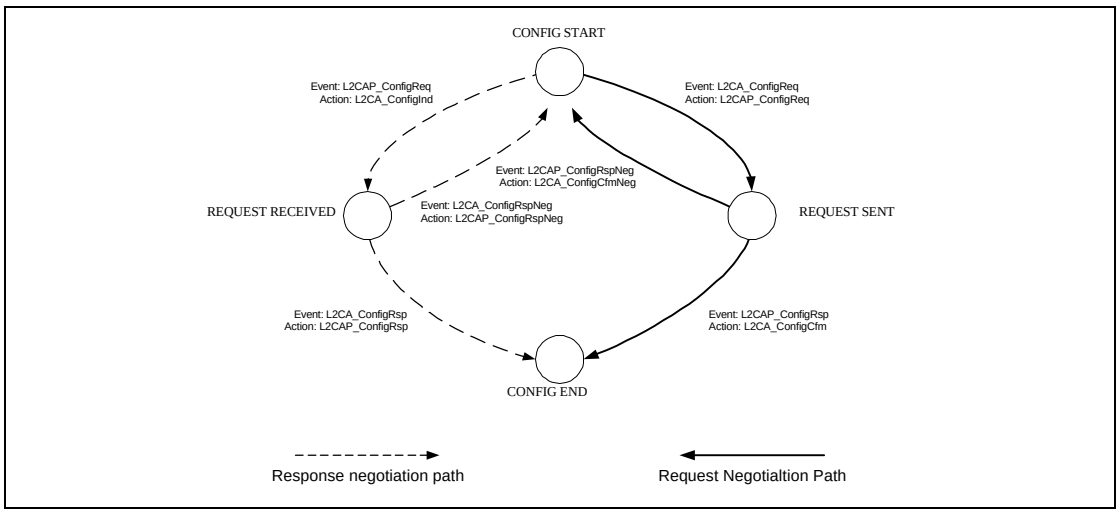


Figure 6.5: Configuration State Machine

"Appendix A: Configuration MSCs" on page 326 provides some configuration examples.

7 SERVICE PRIMITIVES

This section presents an abstract description of the services offered by L2CAP in terms of service primitives and parameters. The service interface is required for testing. The interface is described independently of any platform specific implementation. All data values use Little Endian byte ordering.

7.1 EVENT INDICATION

Service	Input Parameters	Output Parameters
EventIndication	Event, Callback	Result

Description:

The use of this primitive requests a callback when the selected indication Event occurs.

Input Parameters:

Event *Type: uint* *Size: 2 octets*

Value	Description
0x00	Reserved
0x01	L2CA_ConnectInd
0x02	L2CA_ConfigInd
0x03	L2CA_DisconnectInd
0x04	L2CA_QoSViolationInd
other	Reserved for future use

Callback *Type: function* *Size: N/A*

Event	Callback Function Input Parameters
L2CA_ConnectInd	BD_ADDR, CID, PSM, Identifier
L2CA_ConfigInd	CID, OutMTU, InFlow, InFlushTO
L2CA_DisconnectInd	CID
L2CA_QoSViolationInd	BD_ADDR

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Event successfully registered
0x0001	Event registration failed

7.1.1 L2CA_ConnectInd Callback

This callback function includes the parameters for the address of the remote device that issued the connection request, the local CID representing the channel being requested, the Identifier contained in the request, and the PSM value the request is targeting.

7.1.2 L2CA_ConfigInd Callback

This callback function includes the parameters indicating the local CID of the channel the request has been sent to, the outgoing MTU size (maximum packet that can be sent across the channel) and the flowspec describing the characteristics of the incoming data. All other channel parameters are set to their default values if not provided by the remote device.

7.1.3 L2CA_DisconnectInd Callback

This callback function includes the parameter indicating the local CID the request has been sent to.

7.1.4 L2CA_QoSViolationInd Callback

This callback function includes the parameter indicating the address of the remote Bluetooth device where the QoS contract has been violated.

7.2 CONNECT

Service	Input Parameters	Output Parameters
L2CA_ConnectReq	PSM, BD_ADDR	LCID, Result, Status

Description:

This primitive initiates the sending of an L2CAP_ConnectReq message and blocks until a corresponding L2CA_ConnectCfm(Neg) or L2CA_TimeOutInd eventis received.

The use of this primitive requests the creation of a channel representing a logical connection to a physical address. Input parameters are the target protocol (*PSM*) and remote device's 48-bit address (*BD_ADDR*). Output parameters are the local CID *(LCID)* allocated by the local L2CAP entity, and *Result* of the request. If the *Result* indicates success, the *LCID* value contains the identification of the local endpoint. Otherwise the *LCID* returned should be set to 0. If *Result* indicates a pending notification, the *Status* value may contain more information of what processing is delaying the establishment of the connection. Otherwise the *Status* value should be ignored.

Input Parameters:

PSM *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Target PSM provided for the connection

BD_ADDR *Type: unit* *Size: 6 octets*

Value	Description
0xXXXXXXXXXXXX	Unique Bluetooth address of target device

Output Parameters:

LCID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel if Result = 0x0000, otherwise set to 0.

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Connection successful and the CID identifies the local endpoint. Ignore Status parameter
0x0001	Connection pending. Check Status parameter for more information
0x0002	Connection refused because no service for the PSM has been registered
0x0003	Connection refused because the security architecture on the remote side has denied the request
0xEEEE	Connection timeout occurred. This is a result of a timer expiration indication being included in the connection confirm message

Status *Type: uint* *Size: 2 octets*

Value	Description
0x0000	No further information
0x0001	Authentication pending
0x0002	Authorization pending

7.3 CONNECT RESPONSE

Service	Input Parameters	Output Parameters
L2CA_ConnectRsp	BD_ADDR, Identifier, LCID, Response, Status	Result

Description:

This primitive represents the L2CAP_ConnectRsp.

The use of this primitive issues a response to a connection request event indication. Input parameters are the remote device's 48-bit address, Identifier sent in the request, local CID, the Response code, and the Status attached to the Response code. The output parameter is the Result of the service request.

This primitive must be called no more than once after receiving the callback indication. This primitive returns once the local L2CAP entity has validated the request. A successful return does indicate the response has been sent over the air interface.

Input Parameters:

BD_ADDR *Type: unit* *Size: 6 octets*

Value	Description
0xXXXXXXXXXXXX	Unique Bluetooth address of target device

Identifier *Type: uint* *Size: 1 octets*

Value	Description
0xXX.	This value must match the value received in the L2CA_ConnectInd event described in Section 7.1.1 on page 304

LCID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel

Response *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Connection successful
0x0001	Connection pending
0x0002	Connection refused – PSM not supported
0x0003	Connection refused – security block
0x0004	Connection refused – no resources available
0xXXXX	Other connection response code

Status *Type: uint* *Size: 2 octets*

Value	Description
0x0000	No further information available
0x0001	Authentication pending
0x0002	Authorization pending
0xXXXX	Other status code

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Response successfully sent
0x0001	Failure to match any outstanding connection request

7.4 CONFIGURE

Service	Input Parameters	Output Parameters
L2CA_ConfigReq	CID, InMTU, OutFlow, OutFlushTO, LinkTO	Result, InMTU, OutFlow, OutFlushTO

Description:

This primitive initiates the sending of an L2CAP_ConfigReq message and blocks until a corresponding L2CA_ConfigCfm(Neg) or L2CA_TimeOutInd eventis received.

The use of this primitive requests the initial configuration (or reconfiguration) of a channel to a new set of channel parameters. Input parameters are the local CID endpoint, new incoming receivable MTU (InMTU), new outgoing flow specification, and flush and link timeouts. Output parameters composing the L2CA_ConfigCfm(Neg) message are the Result, accepted incoming MTU(InMTU), the remote side's flow requests, and flush and link timeouts. Note that the output results are returned only after the local L2CAP entity transitions out of the CONFIG state (even if this transition is back to the CONFIG state).

Input Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Local CID

InMTU *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Maximum transmission unit this channel can accept

OutFlow *Type: Flow* *Size: x octets*

Value	Description
flowspec	Quality of service parameters dealing with the traffic characteristics of the outgoing data flow

OutFlushTO *Size 2 octets*

Value	Description
0xXXXX	Number of milliseconds to wait before an L2CAP packet that cannot be acknowledged at the physical layer is dropped
0x0000	Request to use the existing flush timeout value if one exists, otherwise the default value (0xFFFF) will be used
0x0001	Perform no retransmissions at the Baseband layer
0xFFFF	Perform retransmission at the Baseband layer until the link timeout terminates the channel

LinkTO *Size 2 octets*

Value	Description
0xXXXX	Number of milliseconds to wait before terminating an unresponsive link

Output Parameters:

Result *Size 2 octets*

Value	Description
0x0000	Configuration is successful. Parameters contain agreed upon values
0x0001	Failure – invalid CID
0x0002	Failure – unacceptable parameters
0x0003	Failure – signalling MTU exceeded
0x0004	Failure – unknown options
0xEEEE	Configuration timeout occurred. This is a result of a timer expiration indication being included in the configuration confirm

InMTU *Size 2 octets*

Value	Description
0xXXXX	Maximum transmission unit that the remote unit will send across this channel (maybe less or equal to the InMTU input parameter).

OutFlow *Size 2 octets*

Value	Description
FlowSpec	Quality of service parameters dealing with the traffic characteristics of the agreed-upon outgoing data flow if Result is successful. Otherwise this represents the requested Quality of Service

OutFlushTO *Size 2 octets*

Value	Description
0xXXXX	Number of milliseconds before an L2CAP packet that cannot be acknowledged at the physical layer is dropped. This value is informative of the actual value that will be used for outgoing packets. It may be less or equal to the OutFlushTO parameter given as input.

7.5 CONFIGURATION RESPONSE

Service	Input Parameters	Output Parameters
L2CA_ConfigRsp	CID, OutMTU, InFlow	Result

Description:

This primitive represents the L2CAP_ConfigRsp.

The use of this primitive issues a response to a configuration request event indication. Input parameters include the local CID of the endpoint being configured, outgoing transmit MTU (which may be equal or less to the OutMTU parameter in the L2CA_ConfigInd event) and the accepted flowspec for incoming traffic. The output parameter is the Result value.

Input Parameters:

LCID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Local channel identifier

OutMTU *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Maximum transmission unit this channel will send

InFlow *Type: Flow* *Size: x octets*

Value	Description
FlowSpec	Quality of service parameters dealing with the traffic characteristics of the incoming data flow

Output Parameters:

Result *Size 2 octets*

Value	Description
0x0000	Configuration is successful. Parameters contain agreed upon values
0x0001	Configuration failed – unacceptable parameters
0x0002	Configuration failed – rejected
0x0003	Configuration failed – invalid CID
0x0004	Configuration failed – unknown options
0xXXXX	Reserved

7.6 DISCONNECT

Service	Input Parameters	Output Parameters
L2CA_DisconnectReq	CID	Result

Description:

This primitive represents the L2CAP_DisconnectReq and the returned output parameters represent the corresponding L2CAP_DisconnectRsp or the RTX timer expiration.

The use of this primitive requests the disconnection of the channel. Input parameter is the *CID* representing the local channel endpoint. Output parameter is *Result*. *Result* is zero if a L2CAP_DisconnectRsp is received, otherwise a non-zero value is returned. Once disconnection has been requested, no process will be able to successfully read or write from the CID. Writes in progress should continue to be processed.

Input Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Disconnection successful. This is a result of the receipt of a disconnection response message
0xEEEE	Disconnection timeout occurred.

7.7 WRITE

Service	Input Parameters	Output Parameters
L2CA_DataWriteReq	CID, Length, OutBuffer	
L2CA_DataWriteCfm		Size, Result

Description:

The use of this primitive requests the transfer of data across the channel. If the length of the data exceeds the OutMTU then only the first OutMTU bytes are sent This command may be used for both connection-oriented and connection-less traffic.

Input Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel

Length *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Size, in bytes, of the buffer where data to be transmitted are stored

OutBuffer *Type: pointer* *Size: N/A*

Value	Description
N/A	Address of the input buffer used to store the message

Output Parameters:

Size *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	The number of bytes transferred

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Successful write
0x0001	Error – Flush timeout expired
0x0002	Error – Link termination (perhaps this should be left to the indication)

7.8 READ

Service	Input Parameters	Output Parameters
L2CA_DataRead	CID, Length, InBuffer	Result, N

Description:

The use of this primitive requests for the reception of data. This request returns when data is available or the link is terminated. The data returned represents a single L2CAP payload. If not enough data is available, the command will block until the data arrives or the link is terminated. If the payload is bigger than the buffer, only the portion of the payload that fits into the buffer will be returned, and the remainder of the payload will be discarded. This command may be used for both connection-oriented and connectionless traffic.

Input Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	CID

Length *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Size, in bytes, of the buffer where received data are to be stored

InBuffer *Type: pointer* *Size: N/A*

Value	Description
N/A	Address of the buffer used to store the message

Output parameters:

Result

Value	Description
0x0000	Success
0x0001	Unsuccessful

N *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Number of bytes transferred to InBuffer

7.9 GROUP CREATE

Service	Input Parameters	Output Parameters
L2CA_GroupCreate	PSM	CID

Description:

The use of this primitive requests the creation of a CID to represent a logical connection to multiple devices. Input parameter is the *PSM* value that the outgoing connectionless traffic is labelled with, and the filter used for incoming traffic. Output parameter is the *CID* representing the local endpoint. On creation, the group is empty but incoming traffic destined for the PSM value is readable.

Input Parameters:

PSM *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Protocol/service multiplexer value

Output Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel

7.10 GROUP CLOSE

Service	Input Parameters	Output Parameters
L2CA_GroupClose	CID	Result

Description:

The use of this primitive closes down a Group.

Input Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Successful closure of the channel
0x0001	Invalid CID

7.11 GROUP ADD MEMBER

Service	Input Parameters	Output Parameters
L2CA_GroupAddMember	CID, BD_ADDR	Result

Description:

The use of this primitive requests the addition of a member to a group. The input parameter includes the CID representing the group and the BD_ADDR of the group member to be added. The output parameter Result confirms the success or failure of the request.

Input Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel

BD_ADDR *Type: uint* *Size: 6 octets*

Value	Description
0xXXXXXXXXXXXX	Remote device address

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Success
0x0001	Failure to establish connection to remote device
Other	Reserved

7.12 GROUP REMOVE MEMBER

Service	Input Parameters	Output Parameters
L2CA_GroupRemoveMember	CID, BD_ADDR	Result

Description:

The use of this primitive requests the removal of a member from a group. The input parameters include the CID representing the group and BD_ADDR of the group member to be removed. The output parameter Result confirms the success or failure of the request.

Input Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel

BD_ADDR *Type: uint* *Size: 6 octets*

Value	Description
0xXXXXXXXXXXXX	Unique Bluetooth address device to be removed

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Success
0x0001	Failure – device not a member of the group
Other	Reserved

7.13 GET GROUP MEMBERSHIP

Service	Input Parameters	Output Parameters
L2CA_GroupMembership	CID	Result, N, BD_ADDR_Lst

Description:

The use of this primitive requests a report of the members of a group. The input parameter CID represents the group being queried. The output parameter Result confirms the success or failure of the operation. If the Result is successful, BD_ADDR_Lst is a list of the Bluetooth addresses of the N members of the group.

Input Parameters:

CID *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Channel ID representing local end-point of the communication channel

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Success
0x0001	Failure – group does not exist
Other	Reserved

N *Type: uint* *Size: 2 octets*

Value	Description
0x0000-0xFFFF	The number of devices in the group identified by the channel end-point CID. If Result indicates failure, N should be set to 0

BD_ADDR_List *Type: pointer* *Size: N/A*

Value	Description
0xXXXXXXXXXXXX	List of N unique Bluetooth addresses of the devices in the group identified by the channel end-point CID. If Result indicates failure, the all-zero address is the only address that should be returned

7.14 PING

Service	Input Parameters	Output Parameters
L2CA_Ping	BD_ADDR, ECHO_DATA, Length	Result, ECHO_DATA, Size

Description:

This primitive represents the initiation of an L2CA_EchoReq messageand the reception of the corresponding L2CAP_EchoRsp message.

Input Parameters:

BD_ADDR *Type: uint* *Size: 6 octets*

Value	Description
0xXXXXXXXXXXXX	Unique Bluetooth address of target device.

ECHO_DATA *Type: pointer* *Size: N/A*

Value	Description
N/A	The buffer containing the contents to be transmitted in the data payload of the Echo Request command.

Length *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Size, in bytes, of the data in the buffer.

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Response received.
0x0001	Timeout occurred.

ECHO_DATA *Type: pointer* *Size: N/A*

Value	Description
N/A	The buffer containing the contents received in the data payload of the Echo Response command.

Size *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Size, in bytes, of the data in the buffer.

7.15 GETINFO

Service	Input Parameters	Output Parameters
L2CA_GetInfo	BD_ADDR, InfoType	Result, InfoData, Size

Description:

This primitive represents the initiation of an L2CA_InfoReq messageand the reception of the corresponding L2CAP_InfoRsp message.

Input Parameters:

BD_ADDR *Type: unit* *Size: 6 octets*

Value	Description
0xXXXXXXXXXXXX	Unique Bluetooth address of target device

InfoType *Type: unit* *Size: 2 octets*

Value	Description
0x0001	Maximum connectionless MTU size

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Response received
0x0001	Not supported
0x0002	Informational PDU rejected, not supported by remote device
0x0003	Timeout occurred

InfoData *Type: pointer* *Size: N/A*

Value	Description
N/A	The buffer containing the contents received in the data payload of the Information Response command.

Size *Type: uint* *Size: 2 octets*

Value	Description
0xXXXX	Size, in bytes, of the data in the InfoData buffer.

7.16 DISABLE CONNECTIONLESS TRAFFIC

Service	Input Parameters	Output Parameters
L2CA_DisableCLT	PSM	Result

Description:

General request to disable the reception of connectionless packets. The input parameter is the *PSM* value indicating service that should be blocked. This command may be used to incrementally disable a set of PSM values. The use of the 'invalid' PSM 0x0000 blocks all connectionless traffic. The output parameter *Result* indicates the success or failure of the command. A limited device might support only general blocking rather than PSM-specific blocks and would fail to block a single non-zero PSM value.

Input Parameters:

PSM *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Block all connectionless traffic
0xXXXX	Protocol/Service Multiplexer field to be blocked

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Successful
0x0001	Failure – not supported

7.17 ENABLE CONNECTIONLESS TRAFFIC

Service	Input Parameters	Output Parameters
L2CA_EnableCLT	PSM	Result

Description:

General request to enable the reception of connectionless packets. The input parameter is the *PSM* value indicating the service that should be unblocked. This command may be used to incrementally enable a set of PSM values. The use of the 'invalid' PSM 0x0000 enables all connectionless traffic. The output parameter *Result* indicates the success or failure of the command. A limited device might support only general enabling rather than PSM-specific filters, and would fail to enable a single non-zero PSM value.

Input Parameters:

PSM *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Enable all connectionless traffic
0xXXXX	Protocol/Service Multiplexer field to enable

Output Parameters:

Result *Type: uint* *Size: 2 octets*

Value	Description
0x0000	Successful
0x0001	Failure – not supported

8 SUMMARY

The Logical Link Control and Adaptation Protocol (L2CAP) is one of two link level protocols running over the Baseband. L2CAP is responsible for higher level protocol multiplexing, MTU abstraction, group management, and conveying quality of service information to the link level.

Protocol multiplexing is supported by defining channels. Each channel is bound to a single protocol in a many-to-one fashion. Multiple channels can be bound to the same protocol, but a channel cannot be bound to multiple protocols. Each L2CAP packet received on a channel is directed to the appropriate higher level protocol.

L2CAP abstracts the variable-sized packets used by the Baseband Protocol (page 33). It supports large packet sizes up to 64 kilobytes using a low-overhead segmentation-and-reassembly mechanism.

Group management provides the abstraction of a group of units allowing more efficient mapping between groups and members of the Bluetooth piconet. Group communication is connectionless and unreliable. When composed of only a pair of units, groups provide connectionless channel alternative to L2CAP's connection-oriented channel.

L2CAP conveys QoS information across channels and provides some admission control to prevent additional channels from violating existing QoS contracts.

9 REFERENCES

[1] Internet Engineering Task Force, “A Proposed Flow Specification”, RFC 1363, September 1992.

10 LIST OF FIGURES

Figure 1.1: L2CAP within protocol layers 259
Figure 1.2: ACL Payload Header for single-slot packets....260
Figure 1.3: ACL Payload Header for multi-slot packets 260
Figure 1.4: L2CAP in Bluetooth Protocol Architecture 261
Figure 2.1: Channels between devices 264
Figure 2.2: L2CAP Architecture....265
Figure 2.3: L2CAP SAR Variables....265
Figure 2.4: L2CAP segmentation 266
Figure 2.5: Segmentation and Reassembly Services in a unit with an HCI 267
Figure 3.1: L2CAP Layer Interactions 268
Figure 3.2: MSC of Layer Interactions....269
Figure 3.3: State Machine Example 280
Figure 3.4: Message Sequence Chart of Basic Operation 281
Figure 4.1: L2CAP Packet (field sizes in bits) 282
Figure 4.2: Connectionless Packet....283
Figure 5.1: Signalling Command Packet Format....285
Figure 5.2: Command format 285
Figure 5.3: Command Reject Packet 287
Figure 5.4: Connection Request Packet....288
Figure 5.5: Connection Response Packet....289
Figure 5.6: Configuration Request Packet 291
Figure 5.7: Configuration Request Flags field format....291
Figure 5.8: Configuration Response Packet....293
Figure 5.9: Configuration Response Flags field format 293
Figure 5.10: Disconnection Request Packet 295
Figure 5.11: Disconnection Response Packet 296
Figure 5.12: Echo Request Packet 296
Figure 5.13: Echo Response Packet....297
Figure 5.14: Information Request Packet....297
Figure 5.15: Information Response Packet....298
Figure 6.1: Configuration option format....299
Figure 6.2: MTU Option Format 300
Figure 6.3: Flush Timeout 300
Figure 6.4: Quality of Service Flow Specification 301
Figure 6.5: Configuration State Machine....304

Figure I Basic MTU exchange 328
Figure II Dealing with Unknown Options 329
Figure III Unsuccessful Configuration Request 330

11 LIST OF TABLES

Table 1.1: Logical channel L_CH field contents 260
Table 2.1: CID Definitions 263
Table 2.2: Types of Channel Identifiers 264
Table 3.1: L2CAP Channel State Machine 277
Table 5.1: Signalling Command Codes 286
Table 5.2: Reason Code Descriptions 287
Table 5.4: Defined PSM Values 288
Table 5.3: Reason Data values 288
Table 5.5: Result values 290
Table 5.6: Status values 290
Table 5.7: Configuration Response Result codes 294
Table 5.8: InfoType definitions 297
Table 5.9: Information Response Result values 298
Table 5.10: Information Response Data fields 298
Table 6.1: Service type definitions 302
Table 6.2: Parameters allowed in Request 303
Table 6.3: Parameters allowed in Response 304

Table I: Result of Second Link Timeout Request 332
Table II: Result of Second Flush Timeout Request 332

TERMS AND ABBREVIATIONS

Baseband	Baseband Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
L_CH	Logical Channel
LC	Link Controller
LM	Link Manager
LMP	Link Manager Protocol
MTU	Maximum Transmission Unit
Reliable	Characteristic of an L2CAP channel that has an infinite flush timeout
RFC	Request For Comments
SAR	Segmentation and Reassembly

APPENDIX A: CONFIGURATION MSCs

The examples in this appendix describe a sample of the multiple possible configuration scenarios that might occur. Currently, these are provided as suggestions and may change in the next update of the Specification.

Figure I illustrates the basic configuration process. In this example, the devices exchange MTU information. All other values are assumed to be default.

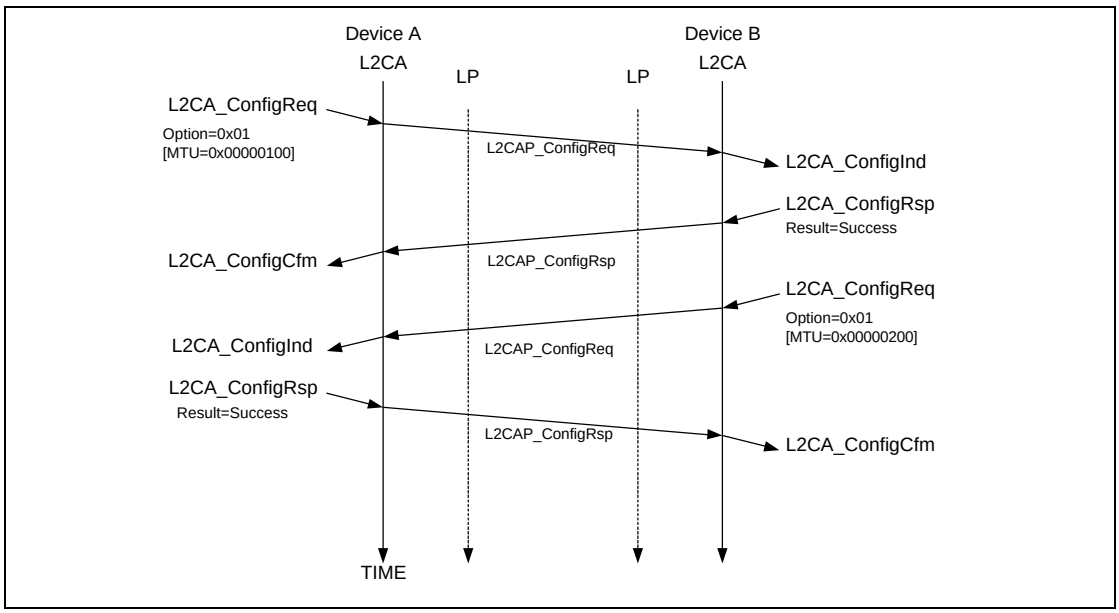


Figure I: Basic MTU exchange

Figure II on page 327 illustrates how two devices interoperate even though one device supports more options than the other does. Device A is an upgraded version. It uses a hypothetically defined option type 0x20 for link-level security. Device B rejects the command using the Configuration Response packet with result 'unknown parameter' informing Device A that option 0x20 is not understood. Device A then resends the request omitting option 0x20. Device B notices that it does not need to such a large MTU and accepts the request but includes in the response the MTU option informing Device A that Device B will not send an L2CAP packet with a payload larger than 0x80 octets over this channel. On receipt of the response, Device A could reduce the buffer allocated to hold incoming traffic.

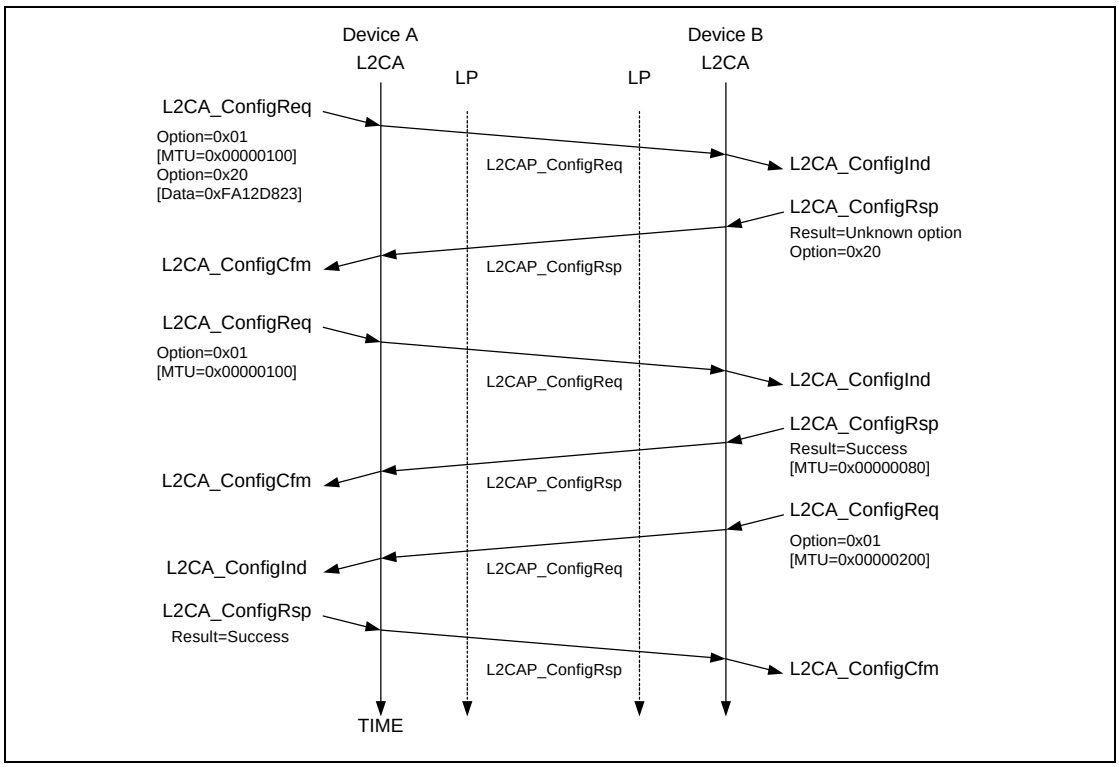


Figure II: Dealing with Unknown Options

Figure III on page 328 illustrates an unsuccessful configuration request. There are two problems described by this example. The first problem is that the configuration request is placed in an L2CAP packet that cannot be accepted by the remote device, due to its size. The remote device informs the sender of this problem using the Command Reject message. Device A then resends the configuration options using two smaller L2CAP_ConfigReq messages.

The second problem is an attempt to configure a channel with an invalid CID. For example device B may not have an open connection on that CID (0x01234567 in this example case).

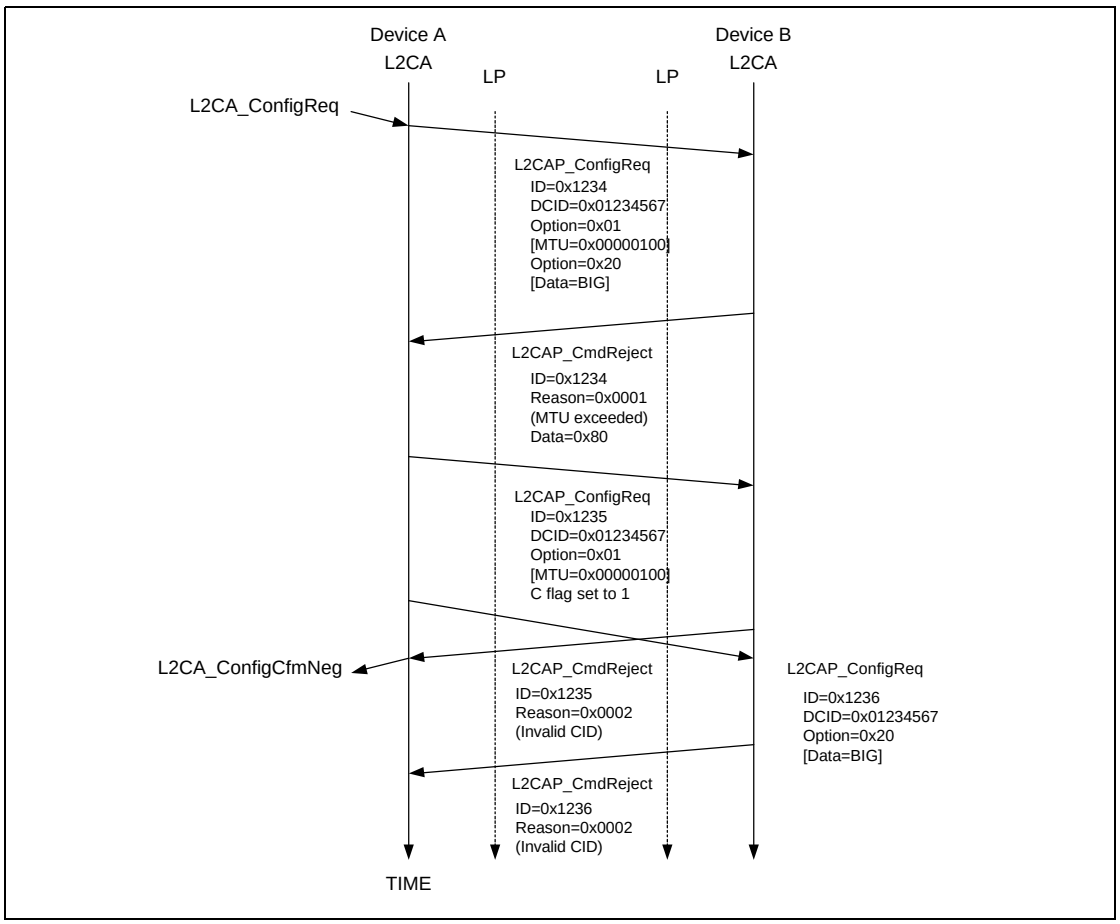


Figure III: Unsuccessful Configuration Request

APPENDIX B: IMPLEMENTATION GUIDELINES

This section contains some guidelines for implementations. These guidelines are not part of the compliance tests. At the moment they are simply suggestions on how to solve some difficult problems.

RTX TIMER

Implementations should not start this timer on an L2CAP Connection Request packet unless the physical link has been established. Otherwise the Baseband paging mechanism might increase the cost of the request beyond that of the minimal timeout value. If an implementation performs some form of security check it is recommended that the connection pending response be sent back prior to any consultation with a security manager that might perform Baseband authentication commands. If any security check requires user interaction, the link might timeout waiting for the user to enter a PIN.

QOS MAPPING TO LM AND L2CAP IMPLEMENTATIONS

Token Rate

The Link Manager (LM) should ensure data is removed from the transmission buffer at this rate. The LM should ensure the polling interval is fast enough to support this data rate. The polling interval should be adjusted if the packet type changes. If the buffer overflows, and the service type is Guaranteed, a QoS violation should be reported. If the service type is Best Effort, and a Token Rate was non-zero, a QoS violation should also be reported.

Given a Token Rate of 0xFFFFFFFF, and Service Type of Guaranteed, the LM should refuse any additional connections from remote devices and disable all periodic scans.

Token Bucket Size

L2CAP implementations should ensure that a buffer meeting the size request is allocated for the channel. If no buffer is available, and the service type is Guaranteed, the request should be rejected. If no appropriately sized buffer is available, and the service type is Best Effort, the largest available buffer should be allocated.

Peak Bandwidth

If the token bucket buffer overflows, a QoS violation should be raised.

Latency

The LM should ensure the polling interval is at least this value. If the polling interval necessary to support the token rate is less than this value, the smaller interval should be used. If this interval cannot be supported, a QoS violation should be raised.

Delay Variation

The LM may ignore this value because there is no clear mapping between L2CAP packet delays and the necessary polling interval without requiring the LM to comprehend the length field in L2CAP packets.

COLLISION TABLES

Current Value	Requested Value	Result
X	X	X
X	Y	If (X < Y) then X, else Y

Table I: Result of Second Link Timeout Request

Current Value	Requested Value	Result
N	0	N
N	N	N
N	M != N	Reject

Table II: Result of Second Flush Timeout Request

Part E

SERVICE DISCOVERY PROTOCOL (SDP)

This specification defines a protocol for locating services provided by or available through a Bluetooth device.

CONTENTS

1 Introduction 337
1.1 General Description 337
1.2 Motivation 337
1.3 Requirements 337
1.4 Non-requirements and Deferred Requirements 338
1.5 Conventions 339
1.5.1 Bit And Byte Ordering Conventions 339
2 Overview 340
2.1 SDP Client-Server Interaction 340
2.2 Service Record 342
2.3 Service Attribute 344
2.4 Attribute ID 345
2.5 Attribute Value 345
2.6 Service Class 346
2.6.1 A Printer Service Class Example 346
2.7 Searching for Services 347
2.7.1 UUID 347
2.7.2 Service Search Patterns 348
2.8 Browsing for Services 348
2.8.1 Example Service Browsing Hierarchy 349
3 Data Representation 351
3.1 Data Element 351
3.2 Data Element Type Descriptor 351
3.3 Data Element Size Descriptor 352
3.4 Data Element Examples 353
4 Protocol Description 354
4.1 Transfer Byte Order 354
4.2 Protocol Data Unit Format 354
4.3 Partial Responses and Continuation State 356
4.4 Error Handling 356
4.4.1 SDP_ErrorResponse PDU 357
4.5 ServiceSearch Transaction 358
4.5.1 SDP_ServiceSearchRequest PDU 358
4.5.2 SDP_ServiceSearchResponse PDU 359
4.6 ServiceAttribute Transaction 361
4.6.1 SDP_ServiceAttributeRequest PDU 361
4.6.2 SDP_ServiceAttributeResponse PDU 363

4.7 ServiceSearchAttribute Transaction 364
4.7.1 SDP_ServiceSearchAttributeRequest PDU 364
4.7.2 SDP_ServiceSearchAttributeResponse PDU 366

5 Service Attribute Definitions 368
5.1 Universal Attribute Definitions 368
5.1.1 ServiceRecordHandle Attribute 368
5.1.2 ServiceClassIDList Attribute 369
5.1.3 ServiceRecordState Attribute 369
5.1.4 ServiceID Attribute 369
5.1.5 ProtocolDescriptorList Attribute 370
5.1.6 BrowseGroupList Attribute 371
5.1.7 LanguageBaseAttributeIDList Attribute 371
5.1.8 ServiceInfoTimeToLive Attribute 372
5.1.9 ServiceAvailability Attribute 373
5.1.10 BluetoothProfileDescriptorList Attribute 373
5.1.11 DocumentationURL Attribute 374
5.1.12 ClientExecutableURL Attribute 374
5.1.13 IconURL Attribute 375
5.1.14 ServiceName Attribute 375
5.1.15 ServiceDescription Attribute 376
5.1.16 ProviderName Attribute 376
5.1.17 Reserved Universal Attribute IDs 376
5.2 ServiceDiscoveryServer Service Class Attribute Definitions 377
5.2.1 ServiceRecordHandle Attribute 377
5.2.2 ServiceClassIDList Attribute 377
5.2.3 VersionNumberList Attribute 377
5.2.4 ServiceDatabaseState Attribute 378
5.2.5 Reserved Attribute IDs 378
5.3 BrowseGroupDescriptor Service Class Attribute Definitions 379
5.3.1 ServiceClassIDList Attribute 379
5.3.2 GroupID Attribute 379
5.3.3 Reserved Attribute IDs 379

Appendix A– Background Information 380

Appendix B – Example SDP Transactions 381

1 INTRODUCTION

1.1 GENERAL DESCRIPTION

The service discovery protocol (SDP) provides a means for applications to discover which services are available and to determine the characteristics of those available services.

1.2 MOTIVATION

Service Discovery in the Bluetooth environment, where the set of services that are available changes dynamically based on the RF proximity of devices in motion, is qualitatively different from service discovery in traditional network-based environments. The service discovery protocol defined in this specification is intended to address the unique characteristics of the Bluetooth environment. See “Appendix A – Background Information,” on page378 , for further information on this topic.

1.3 REQUIREMENTS

The following capabilities have been identified as requirements for version 1.0 of the Service Discovery Protocol.

1. SDP shall provide the ability for clients to search for needed services based on specific attributes of those services.
2. SDP shall permit services to be discovered based on the class of service.
3. SDP shall enable browsing of services without a priori knowledge of the specific characteristics of those services.
4. SDP shall provide the means for the discovery of new services that become available when devices enter RF proximity with a client device as well as when a new service is made available on a device that is in RF proximity with the client device.
5. SDP shall provide a mechanism for determining when a service becomes unavailable when devices leave RF proximity with a client device as well as when a service is made unavailable on a device that is in RF proximity with the client device.
6. SDP shall provide for services, classes of services, and attributes of services to be uniquely identified.
7. SDP shall allow a client on one device to discover a service on another device without consulting a third device.
8. SDP should be suitable for use on devices of limited complexity.
9. SDP shall provide a mechanism to incrementally discover information about the services provided by a device. This is intended to minimize the quantity

of data that must be exchanged in order to determine that a particular service is not needed by a client.

10. SDP should support the caching of service discovery information by intermediary agents to improve the speed or efficiency of the discovery process.
11. SDP should be transport independent.
12. SDP shall function while using L2CAP as its transport protocol.
13. SDP shall permit the discovery and use of services that provide access to other service discovery protocols.
14. SDP shall support the creation and definition of new services without requiring registration with a central authority.

1.4 NON-REQUIREMENTS AND DEFERRED REQUIREMENTS

The Bluetooth SIG recognizes that the following capabilities are related to service discovery. These items are not addressed in SDP version 1.0. However, some may be addressed in future revisions of the specification.

1. SDP 1.0 does not provide access to services. It only provides access to information about services.
2. SDP 1.0 does not provide brokering of services.
3. SDP 1.0 does not provide for negotiation of service parameters.
4. SDP 1.0 does not provide for billing of service use.
5. SDP 1.0 does not provide the means for a client to control or change the operation of a service.
6. SDP 1.0 does not provide an event notification when services, or information about services, become unavailable.
7. SDP 1.0 does not provide an event notification when attributes of services are modified.
8. This specification does not define an application programming interface for SDP.
9. SDP 1.0 does not provide support for service agent functions such as service aggregation or service registration.

1.5 CONVENTIONS

1.5.1 Bit And Byte Ordering Conventions

When multiple bit fields are contained in a single byte and represented in a drawing in this specification, the more significant (high-order) bits are shown toward the left and less significant (low-order) bits toward the right.

Multiple-byte fields are drawn with the more significant bytes toward the left and the less significant bytes toward the right. Multiple-byte fields are transferred in network byte order. See Section 4.1 Transfer Byte Order on page 352.

2 OVERVIEW

2.1 SDP CLIENT-SERVER INTERACTION

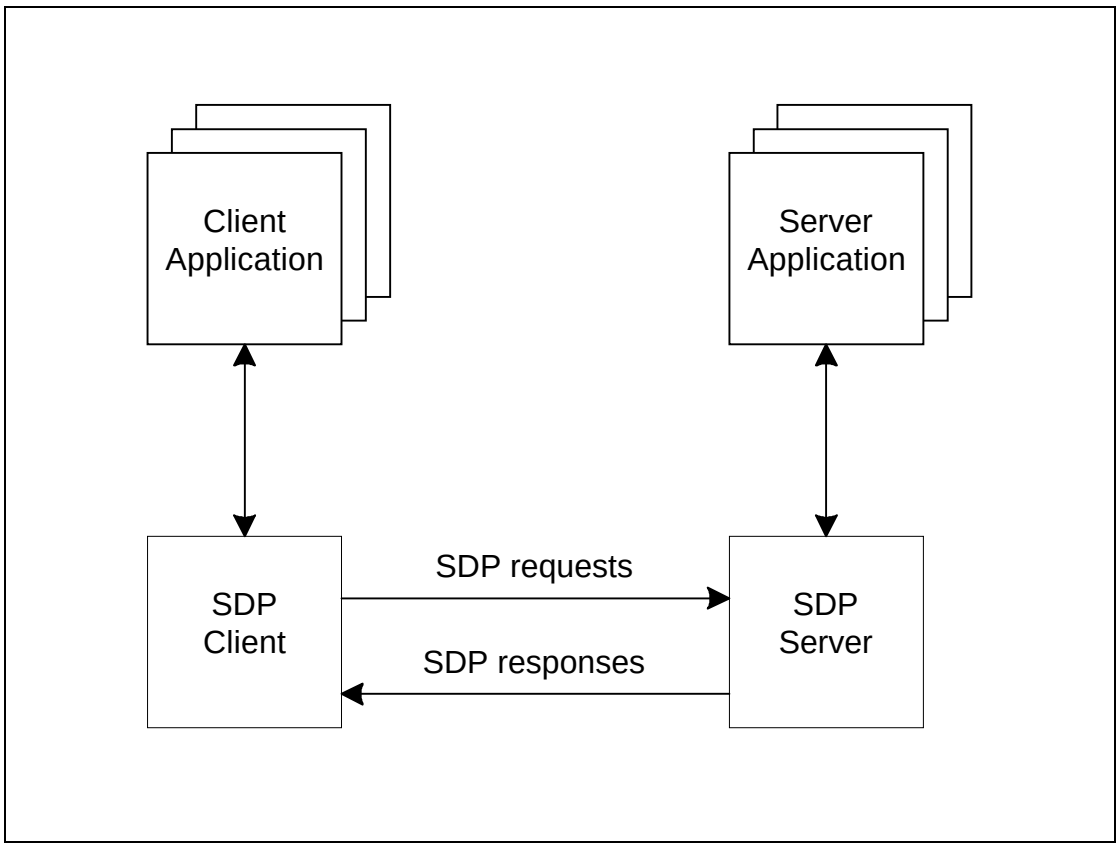


Figure 2.1:

The service discovery mechanism provides the means for client applications to discover the existence of services provided by server applications as well as the attributes of those services. The attributes of a service include the type or class of service offered and the mechanism or protocol information needed to utilize the service.

As far as the Service Discovery Protocol (SDP) is concerned, the configuration shown in Figure 1 may be simplified to that shown in Figure 2.

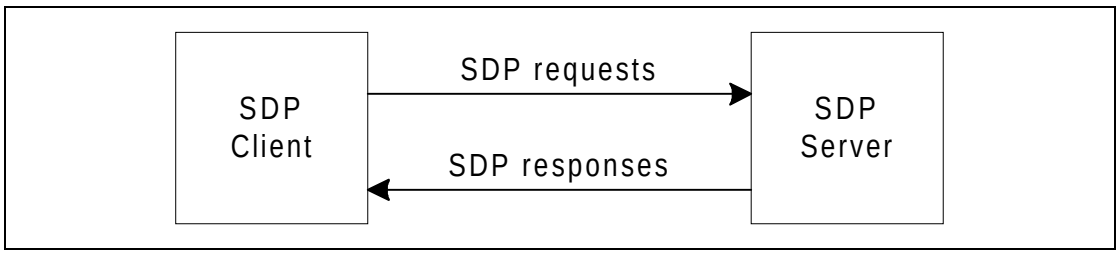


Figure 2.2:

SDP involves communication between an SDP server and an SDP client. The server maintains a list of service records that describe the characteristics of services associated with the server. Each service record contains information about a single service. A client may retrieve information from a service record maintained by the SDP server by issuing an SDP request.

If the client, or an application associated with the client, decides to use a service, it must open a separate connection to the service provider in order to utilize the service. SDP provides a mechanism for discovering services and their attributes (including associated service access protocols), but it does not provide a mechanism for utilizing those services (such as delivering the service access protocols).

There is a maximum of one SDP server per Bluetooth device. (If a Bluetooth device acts only as a client, it needs no SDP server.) A single Bluetooth device may function both as an SDP server and as an SDP client. If multiple applications on a device provide services, an SDP server may act on behalf of those service providers to handle requests for information about the services that they provide.

Similarly, multiple client applications may utilize an SDP client to query servers on behalf of the client applications.

The set of SDP servers that are available to an SDP client can change dynamically based on the RF proximity of the servers to the client. When a server becomes available, a potential client must be notified by a means other than SDP so that the client can use SDP to query the server about its services. Similarly, when a server leaves proximity or becomes unavailable for any reason, there is no explicit notification via the service discovery protocol. However, the client may use SDP to poll the server and may infer that the server is not available if it no longer responds to requests.

Additional information regarding application interaction with SDP is contained in the Bluetooth Service Discovery Profile document.

2.2 SERVICE RECORD

A service is any entity that can provide information, perform an action, or control a resource on behalf of another entity. A service may be implemented as software, hardware, or a combination of hardware and software.

All of the information about a service that is maintained by an SDP server is contained within a single service record. The service record consists entirely of a list of service attributes.

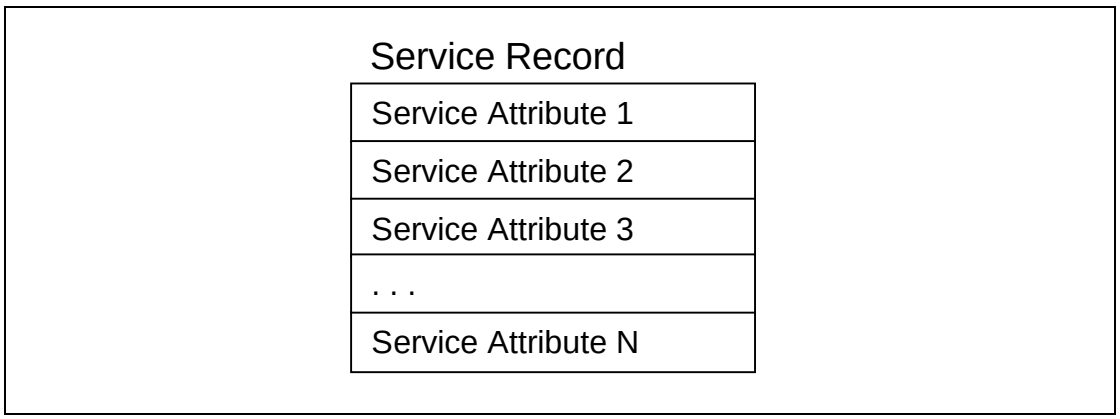


Figure 2.3: Service Record

A service record handle is a 32-bit number that uniquely identifies each service record within an SDP server. It is important to note that, in general, each handle is unique only within each SDP server. If SDP server S1 and SDP server S2 both contain identical service records (representing the same service), the service record handles used to reference these identical service records are completely independent. The handle used to reference the service on S1 will be meaningless if presented to S2.

The service discovery protocol does not provide a mechanism for notifying clients when service records are added to or removed from an SDP server. While an L2CAP (Logical Link Control and Adaptation Protocol) connection is established to a server, a service record handle acquired from the server will remain valid unless the service record it represents is removed. If a service is removed from the server, further requests to the server (during the L2CAP connection in which the service record handle was acquired) using the service's (now stale) record handle will result in an error response indicating an invalid service record handle. An SDP server must ensure that no service record handle values are re-used while an L2CAP connection remains established. Note that service record handles are known to remain valid across successive L2CAP connections while the ServiceDatabaseState attribute value remains unchanged. See the ServiceRecordState and ServiceDatabaseState attributes in Section 5 Service Attribute Definitions on page 366.

There is one service record handle whose meaning is consistent across all SDP servers. This service record handle has the value 0x00000000 and is a

handle to the service record that represents the SDP server itself. This service record contains attributes for the SDP server and the protocol it supports. For example, one of its attributes is the list of SDP protocol versions supported by the server. Service record handle values 0x00000001-0x0000FFFF are reserved.

2.3 SERVICE ATTRIBUTE

Each service attribute describes a single characteristic of a service. Some examples of service attributes are:

ServiceClassIDList	Identifies the type of service represented by a service record. In other words, the list of classes of which the service is an instance
ServiceID	Uniquely identifies a specific instance of a service
ProtocolDescriptorList	Specifies the protocol stack(s) that may be used to utilize a service
ProviderName	The textual name of the individual or organization that provides a service
IconURL	Specifies a URL that refers to an icon image that may be used to represent a service
ServiceName	A text string containing a human-readable name for the service
ServiceDescription	A text string describing the service

See Section 5.1 Universal Attribute Definitions on page 366, for attribute definitions that are common to all service records. Service providers can also define their own service attributes.

A service attribute consists of two components: an attribute ID and an attribute value.

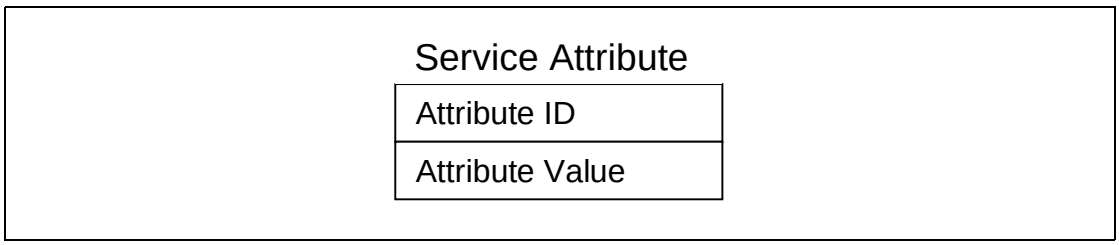


Figure 2.4: Service Attribute

2.4 ATTRIBUTE ID

An attribute ID is a 16-bit unsigned integer that distinguishes each service attribute from other service attributes within a service record. The attribute ID also identifies the semantics of the associated attribute value.

A service class definition specifies each of the attribute IDs for a service class and assigns a meaning to the attribute value associated with each attribute ID.

For example, assume that service class C specifies that the attribute value associated with attribute ID 12345 is a text string containing the date the service was created. Assume further that service A is an instance of service class C. If service A's service record contains a service attribute with an attribute ID of 12345, the attribute value must be a text string containing the date that service A was created. However, services that are not instances of service class C may assign a different meaning to attribute ID 12345.

All services belonging to a given service class assign the same meaning to each particular attribute ID. See Section 2.6 Service Class on page344 .

In the Service Discovery Protocol, an attribute ID is often represented as a data element. See Section 3 Data Representation on page 349.

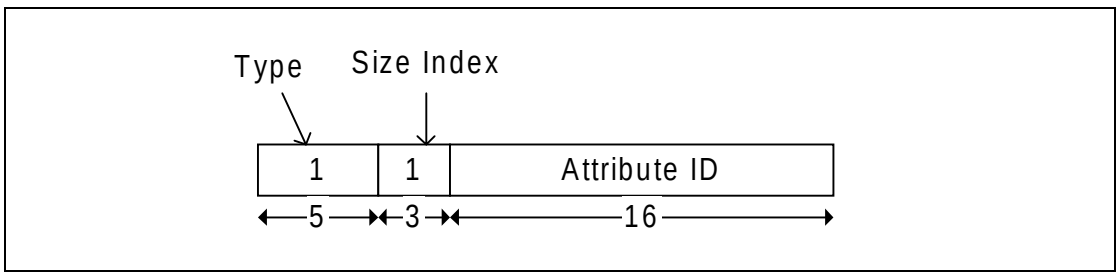


Figure 2.5:

2.5 ATTRIBUTE VALUE

The attribute value is a variable length field whose meaning is determined by the attribute ID associated with it and by the service class of the service record in which the attribute is contained. In the Service Discovery Protocol, an attribute value is represented as a data element. (See Section 3 Data Representation on page 349.) Generally, any type of data element is permitted as an attribute value, subject to the constraints specified in the service class definition that assigns an attribute ID to the attribute and assigns a meaning to the attribute value. See Section 5 Service Attribute Definitions on page 366, for attribute value examples.

2.6 SERVICE CLASS

Each service is an instance of a service class. The service class definition provides the definitions of all attributes contained in service records that represent instances of that class. Each attribute definition specifies the numeric value of the attribute ID, the intended use of the attribute value, and the format of the attribute value. A service record contains attributes that are specific to a service class as well as universal attributes that are common to all services.

Each service class is also assigned a unique identifier. This service class identifier is contained in the attribute value for the ServiceClassIDList attribute, and is represented as a UUID (see Section 2.7.1 UUID on page 345). Since the format and meanings of many attributes in a service record are dependent on the service class of the service record, the ServiceClassIDList attribute is very important. Its value should be examined or verified before any class-specific attributes are used. Since all of the attributes in a service record must conform to all of the service's classes, the service class identifiers contained in the ServiceClassIDList attribute are related. Typically, each service class is a subclass of another class whose identifier is contained in the list. A service subclass definition differs from its superclass in that the subclass contains additional attribute definitions that are specific to the subclass. The service class identifiers in the ServiceClassIDList attribute are listed in order from the most specific class to the most general class.

When a new service class is defined that is a subclass of an existing service class, the new service class retains all of the attributes defined in its superclass. Additional attributes will be defined that are specific to the new service class. In other words, the mechanism for adding new attributes to some of the instances of an existing service class is to create a new service class that is a subclass of the existing service class.

2.6.1 A Printer Service Class Example

A color postscript printer with duplex capability might conform to 4 ServiceClass definitions and have a ServiceClassIDList with UUIDs (See Section 2.7.1 UUID on page345 .) representing the following ServiceClasses:

DuplexColorPostscriptPrinterServiceClassID,
ColorPostscriptPrinterServiceClassID,
PostscriptPrinterServiceClassID,
PrinterServiceClassID

Note that this example is only illustrative. This may not be a practical printer class hierarchy.

2.7 SEARCHING FOR SERVICES

Once an SDP client has a service record handle, it may easily request the values of specific attributes, but how does a client initially acquire a service record handle for the desired service records? The Service Search transaction allows a client to retrieve the service record handles for particular service records based on the values of attributes contained within those service records.

The capability search for service records based on the values of arbitrary attributes is not provided. Rather, the capability is provided to search only for attributes whose values are Universally Unique Identifiers[1] (UUIDs). Important attributes of services that can be used to search for a service are represented as UUIDs.

2.7.1 UUID

A UUID is a universally unique identifier that is guaranteed to be unique across all space and all time. UUIDs can be independently created in a distributed fashion. No central registry of assigned UUIDs is required. A UUID is a 128-bit value.

To reduce the burden of storing and transferring 128-bit UUID values, a range of UUID values has been pre-allocated for assignment to often-used, registered purposes. The first UUID in this pre-allocated range is known as the Bluetooth Base UUID and has the value 00000000-0000-1000-8000-00805F9B34FB, from the Bluetooth Assigned Numbers document. UUID values in the pre-allocated range have aliases that are represented as 16-bit or 32-bit values. These aliases are often called 16-bit and 32-bit UUIDs, but it is important to note that each actually represents a 128-bit UUID value.

The full 128-bit value of a 16-bit or 32-bit UUID may be computed by a simple arithmetic operation.

128_bit_value = 16_bit_value * 2^{96} + Bluetooth_Base_UUID

128_bit_value = 32_bit_value * 2^{96} + Bluetooth_Base_UUID

A 16-bit UUID may be converted to 32-bit UUID format by zero-extending the 16-bit value to 32-bits. An equivalent method is to add the 16-bit UUID value to a zero-valued 32-bit UUID.

Note that two 16-bit UUIDs may be compared directly, as may two 32-bit UUIDs or two 128-bit UUIDs. If two UUIDs of differing sizes are to be compared, the shorter UUID must be converted to the longer UUID format before comparison.

1. The format of UUIDs is defined by the International Organization for Standardization in ISO/IEC 11578:1996. “Information technology – Open Systems Interconnection – Remote Procedure Call (RPC)”

2.7.2 Service Search Patterns

A service search pattern is a list of UUIDs used to locate matching service records. A service search pattern is said to match a service record if each and every UUID in the service search pattern is contained within any of the service record's attribute values. The UUIDs need not be contained within any specific attributes or in any particular order within the service record. The service search pattern matches if the UUIDs it contains constitute a subset of the UUIDs in the service record's attribute values. The only time a service search pattern does not match a service record is if the service search pattern contains at least one UUID that is not contained within the service record's attribute values. Note also that a valid service search pattern must contain at least one UUID.

2.8 BROWSING FOR SERVICES

Normally, a client searches for services based on some desired characteristic(s) (represented by a UUID) of the services. However, there are times when it is desirable to discover which types of services are described by an SDP server's service records without any a priori information about the services. This process of looking for any offered services is termed browsing. In SDP, the mechanism for browsing for services is based on an attribute shared by all service classes. This attribute is called the BrowseGroupList attribute. The value of this attribute contains a list of UUIDs. Each UUID represents a browse group with which a service may be associated for the purpose of browsing.

When a client desires to browse an SDP server's services, it creates a service search pattern containing the UUID that represents the root browse group. All services that may be browsed at the top level are made members of the root browse group by having the root browse group's UUID as a value within the BrowseGroupList attribute.

Normally, if an SDP server has relatively few services, all of its services will be placed in the root browse group. However, the services offered by an SDP server may be organized in a browse group hierarchy, by defining additional browse groups below the root browse group. Each of these additional browse groups is described by a service record with a service class of BrowseGroupDescriptor.

A browse group descriptor service record defines a new browse group by means of its Group ID attribute. In order for a service contained in one of these newly defined browse groups to be browseable, the browse group descriptor service record that defines the new browse group must in turn be browseable. The hierarchy of browseable services that is provided by the use of browse group descriptor service records allows the services contained in an SDP server to be incrementally browsed and is particularly useful when the SDP server contains many service records.

2.8.1 Example Service Browsing Hierarchy

Here is a fictitious service browsing hierarchy that may illuminate the manner in which browse group descriptors are used. Browse group descriptor service records are identified with (G); other service records with (S).

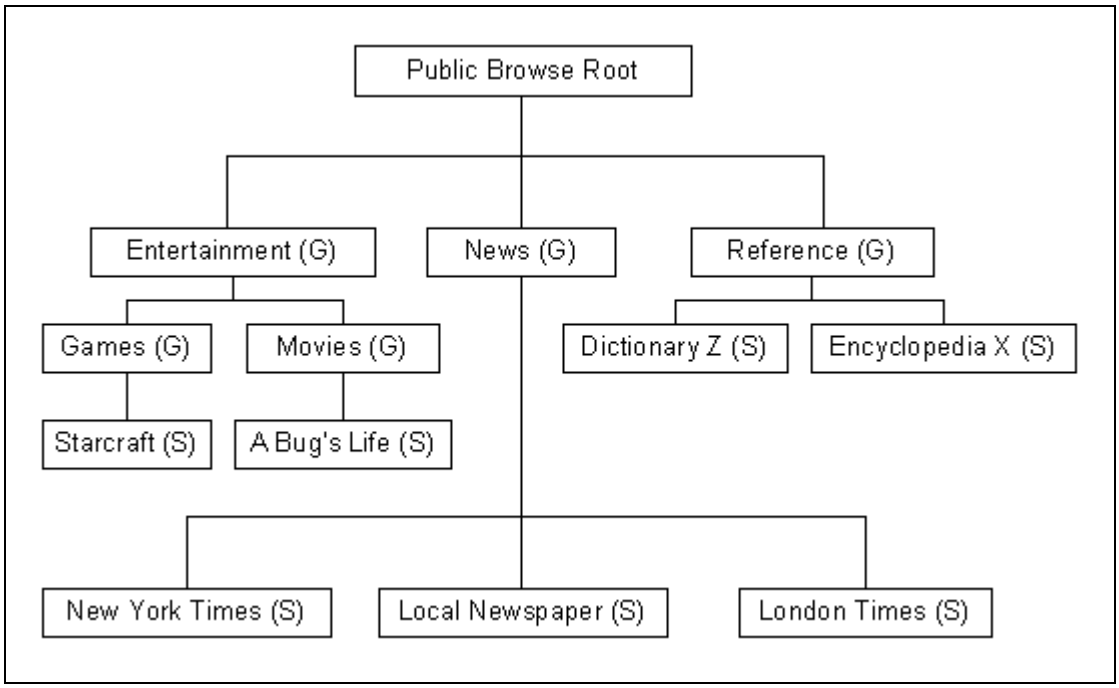


Figure 2.6:

This table shows the services records and service attributes necessary to implement the browse hierarchy.

Service Name	Service Class	Attribute Name	Attribute Value
Entertainment	BrowseGroupDescriptor	BrowseGroupList	PublicBrowseRoot
		GroupID	EntertainmentID
News	BrowsegroupDescriptor	BrowseGroupList	PublicBrowseRoot
		GroupID	NewsID
Reference	BrowseGroupDescriptor	BrowseGroupList	PublicBrowseRoot
		GroupID	ReferenceID
Games	BrowseGroupDescriptor	BrowseGroupList	EntertainmentID
		GroupID	GamesID
Movies	BrowseGroupDescriptor	BrowseGroupList	EntertainmentID
		GroupID	MoviesID
Starcraft	Video Game Class ID	BrowseGroupList	GamesID

Table 2.1:

A Bug's Life	Movie Class ID	BrowseGroupList	MovieID
Dictionary Z	Dictionary Class ID	BrowseGroupList	ReferenceID
Encyclopedia X	Encyclopedia Class ID	BrowseGroupList	ReferenceID
New York Times	Newspaper ID	BrowseGroupList	NewspaperID
London Times	Newspaper ID	BrowseGroupList	NewspaperID
Local Newspaper	Newspaper ID	BrowseGroupList	NewspaperID

Table 2.1:

3 DATA REPRESENTATION

Attribute values can contain information of various types with arbitrary complexity; thus enabling an attribute list to be generally useful across a wide variety of service classes and environments.

SDP defines a simple mechanism to describe the data contained within an attribute value. The primitive construct used is the data element.

3.1 DATA ELEMENT

A data element is a typed data representation. It consists of two fields: a header field and a data field. The header field, in turn, is composed of two parts: a type descriptor and a size descriptor. The data is a sequence of bytes whose length is specified in the size descriptor (described in Section 3.3 Data Element Size Descriptor on page350) and whose meaning is (partially) specified by the type descriptor.

3.2 DATA ELEMENT TYPE DESCRIPTOR

A data element type is represented as a 5-bit type descriptor. The type descriptor is contained in the most significant (high-order) 5 bits of the first byte of the data element header. The following types have been defined.

Type Descriptor Value	Valid Size Descriptor Values	Type Description
0	0	Nil, the null type
1	0, 1, 2, 3, 4	Unsigned Integer
2	0, 1, 2, 3, 4	Signed twos-complement integer
3	1, 2, 4	UUID, a universally unique identifier
4	5, 6, 7	Text string
5	0	Boolean
6	5, 6, 7	Data element sequence, a data element whose data field is a sequence of data elements
7	5, 6, 7	Data element alternative, data element whose data field is a sequence of data elements from which one data element is to be selected.
8	5, 6, 7	URL, a uniform resource locator
9-31		Reserved

Table 3.1:

3.3 DATA ELEMENT SIZE DESCRIPTOR

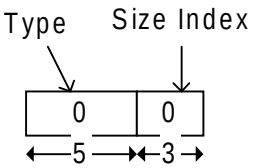
The data element size descriptor is represented as a 3-bit size index followed by 0, 8, 16, or 32 bits. The size index is contained in the least significant (low-order) 3 bits of the first byte of the data element header. The size index is encoded as follows.

Size Index	Additional bits	Data Size
0	0	1 byte. Exception: if the data element type is nil, the data size is 0 bytes.
1	0	2 bytes
2	0	4 bytes
3	0	8 bytes
4	0	16 bytes
5	8	The data size is contained in the additional 8 bits, which are interpreted as an unsigned integer.
6	16	The data size is contained in the additional 16 bits, which are interpreted as an unsigned integer.
7	32	The data size is contained in the additional 32 bits, which are interpreted as an unsigned integer.

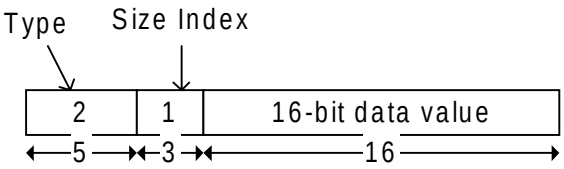
Table 3.2:

3.4 DATA ELEMENT EXAMPLES

Nil is represented as:



A 16-bit signed integer is represented as:



he 3 character ASCII string "Hat" is represented as:

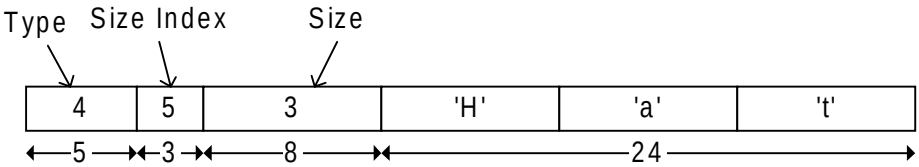


Figure 3.1:

4 PROTOCOL DESCRIPTION

SDP is a simple protocol with minimal requirements on the underlying transport. It can function over a reliable packet transport (or even unreliable, if the client implements timeouts and repeats requests as necessary).

SDP uses a request/response model where each transaction consists of one request protocol data unit (PDU) and one response PDU. In the case where SDP is used with the Bluetooth L2CAP transport protocol, only one SDP request PDU per connection to a given SDP server may be outstanding at a given instant. In other words, a client must receive a response to each request before issuing another request on the same L2CAP connection. Limiting SDP to sending one unacknowledged request PDU provides a simple form of flow control.

The protocol examples found in Appendix B – Example SDP Transactions, may be helpful in understanding the protocol transactions.

4.1 TRANSFER BYTE ORDER

The service discovery protocol transfers multiple-byte fields in standard network byte order (Big Endian), with more significant (high-order) bytes being transferred before less-significant (low-order) bytes.

4.2 PROTOCOL DATA UNIT FORMAT

Every SDP PDU consists of a PDU header followed by PDU-specific parameters. The header contains three fields: a PDU ID, a Transaction ID, and a ParameterLength. Each of these header fields is described here. Parameters may include a continuation state parameter, described below; PDU-specific parameters for each PDU type are described later in separate PDU descriptions.

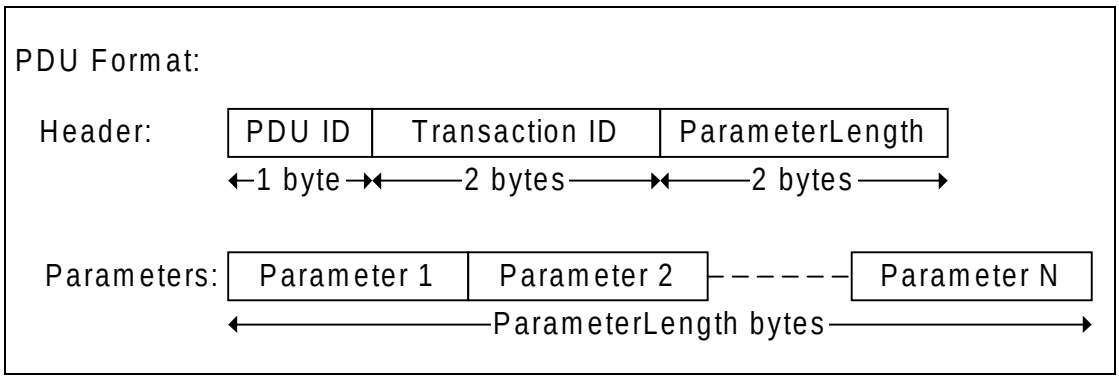


Figure 4.1:

PDU ID: *Size: 1 Byte*

Value	Parameter Description
N	The PDU ID field identifies the type of PDU. I.e. its meaning and the specific parameters.
0x00	Reserved
0x01	SDP_ErrorResponse
0x02	SDP_ServiceSearchRequest
0x03	SDP_ServiceSearchResponse
0x04	SDP_ServiceAttributeRequest
0x05	SDP_ServiceAttributeResponse
0x06	SDP_ServiceSearchAttributeRequest
0x07	SDP_ServiceSearchAttributeResponse
0x07-0xFF	Reserved

TransactionID: *Size: 2 Bytes*

Value	Parameter Description
N	The TransactionID field uniquely identifies request PDUs and is used to match response PDUs to request PDUs. The SDP client can choose any value for a request's TransactionID provided that it is different from all outstanding requests. The TransactionID value in response PDUs is required to be the same as the request that is being responded to. Range: 0x0000 – 0xFFFF

ParameterLength: *Size: 2 Bytes*

Value	Parameter Description
N	The ParameterLength field specifies the length (in bytes) of all parameters contained in the PDU. Range: 0x0000 – 0xFFFF

4.3 PARTIAL RESPONSES AND CONTINUATION STATE

Some SDP requests may require responses that are larger than can fit in a single response PDU. In this case, the SDP server will generate a partial response along with a continuation state parameter. The continuation state parameter can be supplied by the client in a subsequent request to retrieve the next portion of the complete response. The continuation state parameter is a variable length field whose first byte contains the number of additional bytes of continuation information in the field. The format of the continuation information is not standardized among SDP servers. Each continuation state parameter is meaningful only to the SDP server that generated it.

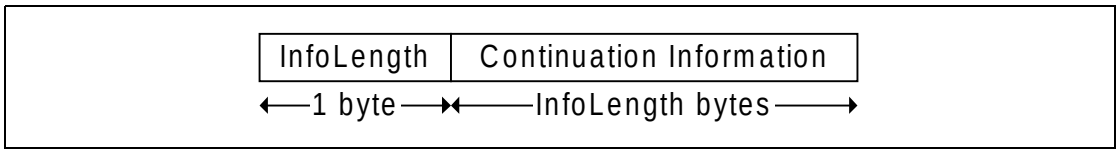


Figure 4.2: Continuation State Format

After a client receives a partial response and the accompanying continuation state parameter, it can re-issue the original request (with a new transaction ID) and include the continuation state in the new request indicating to the server that the remainder of the original response is desired. The maximum allowable value of the InfoLength field is 16 (0x10).

Note that an SDP server can split a response at any arbitrary boundary when it generates a partial response. The SDP server may select the boundary based on the contents of the reply, but is not required to do so.

4.4 ERROR HANDLING

Each transaction consists of a request and a response PDU. Generally, each type of request PDU has a corresponding type of response PDU. However, if the server determines that a request is improperly formatted or for any reason the server cannot respond with the appropriate PDU type, it will respond with an SDP_ErrorResponse PDU.

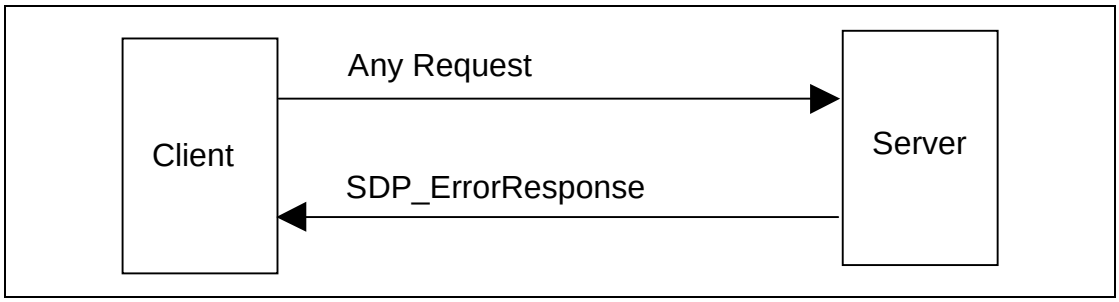


Figure 4.3:

4.4.1 SDP_ErrorResponse PDU

PDU Type	PDU ID	Parameters
SDP_ErrorResponse	0x01	ErrorCode, ErrorInfo

Description:

The SDP server generates this PDU type in response to an improperly formatted request PDU or when the SDP server, for whatever reason, cannot generate an appropriate response PDU.

PDU Parameters:

ErrorCode: *Size: 2 Bytes*

Value	Parameter Description
N	The ErrorCode identifies the reason that an SDP_ErrorResponse PDU was generated.
0x0000	Reserved
0x0001	Invalid/unsupported SDP version
0x0002	Invalid Service Record Handle
0x0003	Invalid request syntax
0x0004	Invalid PDU Size
0x0005	Invalid Continuation State
0x0006	Insufficient Resources to satisfy Request
0x0007-0xFFFF	Reserved

ErrorInfo: *Size: N Bytes*

Value	Parameter Description
Error-specific	ErrorInfo is an ErrorCode-specific parameter. Its interpretation depends on the ErrorCode parameter. The currently defined ErrorCode values do not specify the format of an ErrorInfo field.

4.5 SERVICESEARCH TRANSACTION

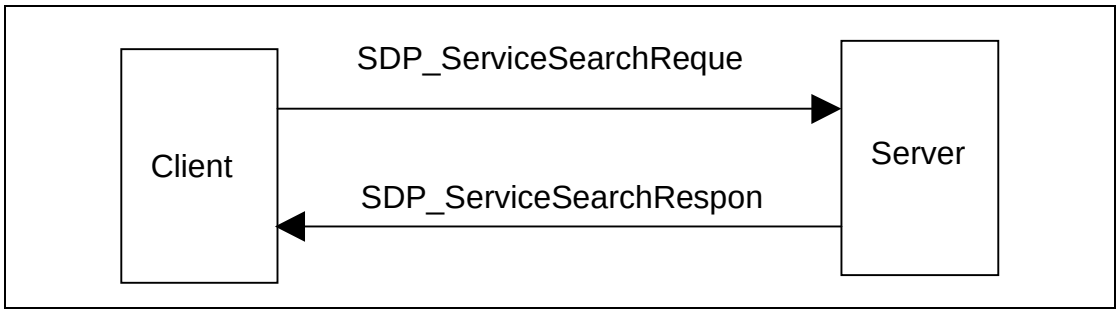


Figure 4.4:

4.5.1 SDP_ServiceSearchRequest PDU

PDU Type	PDU ID	Parameters
SDP_ServiceSearchRequest	0x02	ServiceSearchPattern, MaximumServiceRecordCount, ContinuationState

Description:

The SDP client generates an SDP_ServiceSearchRequest to locate service records that match the service search pattern given as the first parameter of the PDU. Upon receipt of this request, the SDP server will examine its service record data base and return an SDP_ServiceSearchResponse containing the service record handles of service records that match the given service search pattern.

Note that no mechanism is provided to request information for all service records. However, see Section 2.8 Browsing for Services on page346 for a description of a mechanism that permits browsing for non-specific services without a priori knowledge of the services.

PDU Parameters:

ServiceSearchPattern: *Size: Varies*

Value	Parameter Description
Data Element Sequence	The ServiceSearchPattern is a data element sequence where each element in the sequence is a UUID. The sequence must contain at least one UUID. The maximum number of UUIDs in the sequence is 12*. The list of UUIDs constitutes a service search pattern.

*. The value of 12 has been selected as a compromise between the scope of a service search and the size of a search request PDU. It is not expected that more than 12 UUIDs will be useful in a service search pattern.

MaximumServiceRecordCount: *Size: 2 Bytes*

Value	Parameter Description
N	MaximumServiceRecordCount is a 16-bit count specifying the maximum number of service record handles to be returned in the response(s) to this request. The SDP server should not return more handles than this value specifies. If more than N service records match the request, the SDP server determines which matching service record handles to return in the response(s). Range: 0x0001-0xFFFF

ContinuationState: *Size: 1 to 17 Bytes*

Value	**Parameter Description**
Continuation State	ContinuationState consists of an 8-bit count, N, of the number of bytes of continuation state information, followed by the N bytes of continuation state information that were returned in a previous response from the server. N is required to be less than or equal to 16. If no continuation state is to be provided in the request, N is set to 0.

4.5.2 SDP_ServiceSearchResponse PDU

PDU Type	**PDU ID**	**Parameters**
SDP_ServiceSearchResponse	0x03	TotalServiceRecordCount, CurrentServiceRecordCount, ServiceRecordHandleList, ContinuationState

Description:

The SDP server generates an SDP_ServiceSearchResponse upon receipt of a valid SDP_ServiceSearchRequest. The response contains a list of service record handles for service records that match the service search pattern given in the request. Note that if a partial response is generated, it must contain an integral number of complete service record handles; a service record handle value may not be split across multiple PDUs.

PDU Parameters:

TotalServiceRecordCount: *Size: 2 Bytes*

Value	Parameter Description
N	The TotalServiceRecordCount is an integer containing the number of service records that match the requested service search pattern. If no service records match the requested service search pattern, this parameter is set to 0. N should never be larger than the MaximumServiceRecordCount value specified in the SDP_ServiceSearchRequest. When multiple partial responses are used, each partial response contains the same value for TotalServiceRecordCount. Range: 0x0000-0xFFFF

CurrentServiceRecordCount: *Size: 2 Bytes*

Value	Parameter Description
N	The CurrentServiceRecordCount is an integer indicating the number of service record handles that are contained in the next parameter. If no service records match the requested service search pattern, this parameter is set to 0. N should never be larger than the TotalServiceRecordCount value specified in the current response. Range: 0x0000-0xFFFF

ServiceRecordHandleList: *Size: (CurrentServiceRecordCount*4) Bytes*

Value	Parameter Description
List of 32-bit handles	The ServiceRecordHandleList contains a list of service record handles. The number of handles in the list is given in the CurrentServiceRecordCount parameter. Each of the handles in the list refers to a service record that matches the requested service search pattern. Note that this list of service record handles does not have the format of a data element. It contains no header fields, only the 32-bit service record handles.

ContinuationState: *Size: 1 to 17 Bytes*

Value	Parameter Description
Continuation State	ContinuationState consists of an 8-bit count, N, of the number of bytes of continuation state information, followed by the N bytes of continuation information. If the current response is complete, this parameter consists of a single byte with the value 0. If a partial response is contained in the PDU, the ContinuationState parameter may be supplied in a subsequent request to retrieve the remainder of the response.

4.6 SERVICEATTRIBUTE TRANSACTION

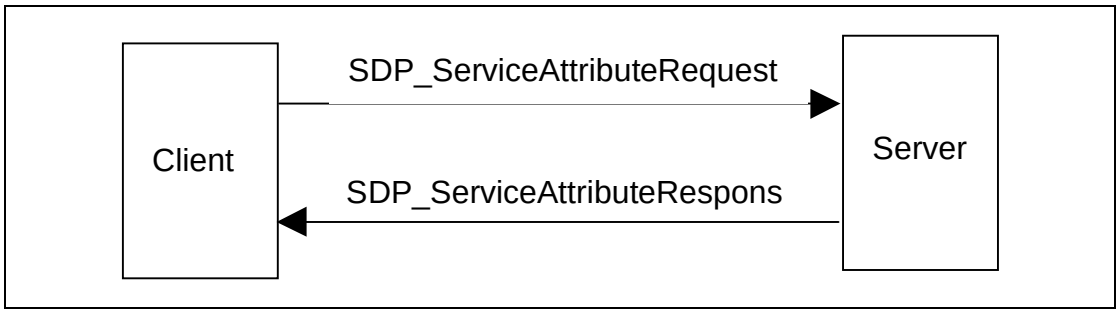


Figure 4.5:

4.6.1 SDP_ServiceAttributeRequest PDU

PDU Type	PDU ID	Parameters
SDP_ServiceAttributeRequest	0x04	ServiceRecordHandle, MaximumAttributeByteCount, AttributeIDList, ContinuationState

Description:

The SDP client generates an SDP_ServiceAttributeRequest to retrieve specified attribute values from a specific service record. The service record handle of the desired service record and a list of desired attribute IDs to be retrieved from that service record are supplied as parameters.

Command Parameters:

ServiceRecordHandle: *Size: 4 Bytes*

Value	Parameter Description
32-bit handle	The ServiceRecordHandle parameter specifies the service record from which attribute values are to be retrieved. The handle is obtained via a previous SDP_ServiceSearch transaction.

MaximumAttributeByteCount: *Size: 2 Bytes*

Value	Parameter Description
N	MaximumAttributeByteCount specifies the maximum number of bytes of attribute data to be returned in the response to this request. The SDP server should not return more than N bytes of attribute data in the response PDU. If the requested attributes require more than N bytes, the SDP server determines how to segment the list. In this case the client may request each successive segment by issuing a request containing the continuation state that was returned in the previous response PDU. Note that in the case where multiple response PDUs are needed to return the attribute data, MaximumAttributeByteCount specifies the maximum size of the portion of the attribute data contained in each response PDU. Range: 0x0007-0xFFFF

AttributeIDList: *Size: Varies*

Value	Parameter Description
Data Element Sequence	The AttributeIDList is a data element sequence where each element in the list is either an attribute ID or a range of attribute IDs. Each attribute ID is encoded as a 16-bit unsigned integer data element. Each attribute ID range is encoded as a 32-bit unsigned integer data element, where the high order 16 bits are interpreted as the beginning attribute ID of the range and the low order 16 bits are interpreted as the ending attribute ID of the range. The attribute IDs contained in the AttributeIDList must be listed in ascending order without duplication of any attribute ID values. Note that all attributes may be requested by specifying a range of 0x0000-0xFFFF.

ContinuationState: *Size: 1 to 17 Bytes*

Value	Parameter Description
Continuation State	ContinuationState consists of an 8-bit count, N, of the number of bytes of continuation state information, followed by the N bytes of continuation state information that were returned in a previous response from the server. N is required to be less than or equal to 16. If no continuation state is to be provided in the request, N is set to 0.

4.6.2 SDP_ServiceAttributeResponse PDU

PDU Type	PDU ID	Parameters
SDP_ServiceAttributeResponse	0x05	AttributeListByteCount, AttributeList, ContinuationState

Description:

The SDP server generates an SDP_ServiceAttributeResponse upon receipt of a valid SDP_ServiceAttributeRequest. The response contains a list of attributes (both attribute ID and attribute value) from the requested service record.

PDU Parameters:

AttributeListByteCount: *Size: 2 Bytes*

Value	Parameter Description
N	The AttributeListByteCount contains a count of the number of bytes in the AttributeList parameter. N must never be larger than the MaximumAttributeByteCount value specified in the SDP_ServiceAttributeRequest. Range: 0x0002-0xFFFF

AttributeList: *Size: AttributeListByteCount*

Value	Parameter Description
Data Element Sequence	The AttributeList is a data element sequence containing attribute IDs and attribute values. The first element in the sequence contains the attribute ID of the first attribute to be returned. The second element in the sequence contains the corresponding attribute value. Successive pairs of elements in the list contain additional attribute ID and value pairs. Only attributes that have non-null values within the service record and whose attribute IDs were specified in the SDP_ServiceAttributeRequest are contained in the AttributeList. Neither an attribute ID nor an attribute value is placed in the AttributeList for attributes in the service record that have no value. The attributes are listed in ascending order of attribute ID value.

ContinuationState: *Size: 1 to 17 Bytes*

Value	Parameter Description
Continuation State	ContinuationState consists of an 8-bit count, N, of the number of bytes of continuation state information, followed by the N bytes of continuation information. If the current response is complete, this parameter consists of a single byte with the value 0. If a partial response is given, the ContinuationState parameter may be supplied in a subsequent request to retrieve the remainder of the response.

4.7 SERVICESEARCHATTRIBUTE TRANSACTION

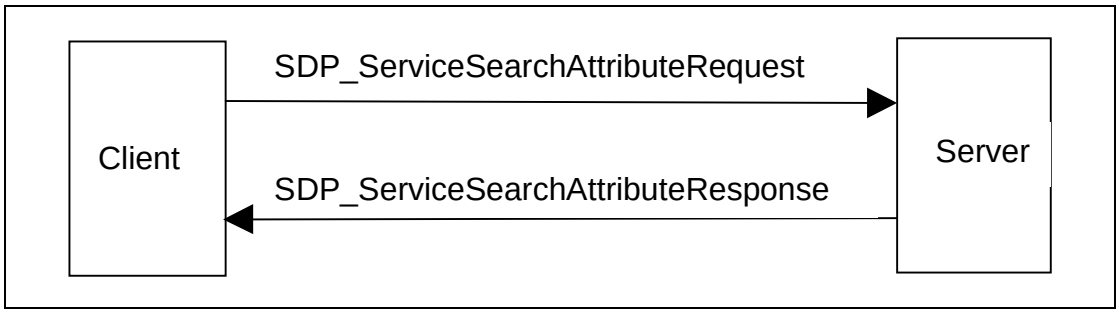


Figure 4.6:

4.7.1 SDP_ServiceSearchAttributeRequest PDU

PDU Type	PDU ID	Parameters
SDP_ServiceSearchAttributeRequest	0x06	ServiceSearchPattern, MaximumAttributeByteCount, AttributeIDList, ContinuationState

Description:

The SDP_ServiceSearchAttributeRequest transaction combines the capabilities of the SDP_ServiceSearchRequest and the SDP_ServiceAttributeRequest into a single request. As parameters, it contains both a service search pattern and a list of attributes to be retrieved from service records that match the service search pattern. The SDP_ServiceSearchAttributeRequest and its response are more complex and may require more bytes than separate SDP_ServiceSearch and SDP_ServiceAttribute transactions. However, using SDP_ServiceSearchAttributeRequest may reduce the total number of SDP transactions, particularly when retrieving multiple service records.

Note that the service record handle for each service record is contained in the ServiceRecordHandle attribute of that service and may be requested along with other attributes.

PDU Parameters:

ServiceSearchPattern: *Size: Varies*

Value	Parameter Description
Data Element Sequence	The ServiceSearchPattern is a data element sequence where each element in the sequence is a UUID. The sequence must contain at least one UUID. The maximum number of UUIDs in the sequence is 12*. The list of UUIDs constitutes a service search pattern.

*. The value of 12 has been selected as a compromise between the scope of a service search and the size of a search request PDU. It is not expected that more than 12 UUIDs will be useful in a service search pattern.

MaximumAttributeByteCount: *Size: 2 Bytes*

Value	Parameter Description
N	MaximumAttributeByteCount specifies the maximum number of bytes of attribute data to be returned in the response to this request. The SDP server should not return more than N bytes of attribute data in the response PDU. If the requested attributes require more than N bytes, the SDP server determines how to segment the list. In this case the client may request each successive segment by issuing a request containing the continuation state that was returned in the previous response PDU. Note that in the case where multiple response PDUs are needed to return the attribute data, MaximumAttributeByteCount specifies the maximum size of the portion of the attribute data contained in each response PDU. Range: 0x0007-0xFFFF

AttributeIDList: *Size: Varies*

Value	Parameter Description
Data Element Sequence	The AttributeIDList is a data element sequence where each element in the list is either an attribute ID or a range of attribute IDs. Each attribute ID is encoded as a 16-bit unsigned integer data element. Each attribute ID range is encoded as a 32-bit unsigned integer data element, where the high order 16 bits are interpreted as the beginning attribute ID of the range and the low order 16 bits are interpreted as the ending attribute ID of the range. The attribute IDs contained in the AttributeIDList must be listed in ascending order without duplication of any attribute ID values. Note that all attributes may be requested by specifying a range of 0x0000-0xFFFF.

ContinuationState: *Size: 1 to 17 Bytes*

Value	Parameter Description
Continuation State	ContinuationState consists of an 8-bit count, N, of the number of bytes of continuation state information, followed by the N bytes of continuation state information that were returned in a previous response from the server. N is required to be less than or equal to 16. If no continuation state is to be provided in the request, N is set to 0.

4.7.2 SDP_ServiceSearchAttributeResponse PDU

PDU Type	PDU ID	Parameters
SDP_ServiceSearchAttributeResponse	0x07	AttributeListsByteCount, AttributeLists, ContinuationState

Description:

The SDP server generates an SDP_ServiceSearchAttributeResponse upon receipt of a valid SDP_ServiceSearchAttributeRequest. The response contains a list of attributes (both attribute ID and attribute value) from the service records that match the requested service search pattern.

PDU Parameters:

AttributeListsByteCount: *Size: 2 Bytes*

Value	Parameter Description
N	The AttributeListsByteCount contains a count of the number of bytes in the AttributeLists parameter. N must never be larger than the MaximumAttributeByteCount value specified in the SDP_ServiceSearchAttributeRequest. Range: 0x0002-0xFFFF

AttributeLists: *Size: Varies*

Value	Parameter Description
Data Element Sequence	The AttributeLists is a data element sequence where each element in turn is a data element sequence representing an attribute list. Each attribute list contains attribute IDs and attribute values from one service record. The first element in each attribute list contains the attribute ID of the first attribute to be returned for that service record. The second element in each attribute list contains the corresponding attribute value. Successive pairs of elements in each attribute list contain additional attribute ID and value pairs. Only attributes that have non-null values within the service record and whose attribute IDs were specified in the SDP_ServiceSearchAttributeRequest are contained in the AttributeLists. Neither an attribute ID nor attribute value is placed in AttributeLists for attributes in the service record that have no value. Within each attribute list, the attributes are listed in ascending order of attribute ID value.

ContinuationState: *Size: 1 to 17 Bytes*

Value	Parameter Description
Continuation State	ContinuationState consists of an 8-bit count, N, of the number of bytes of continuation state information, followed by the N bytes of continuation information. If the current response is complete, this parameter consists of a single byte with the value 0. If a partial response is given, the ContinuationState parameter may be supplied in a subsequent request to retrieve the remainder of the response.

5 SERVICE ATTRIBUTE DEFINITIONS

The service classes and attributes contained in this document are necessarily a partial list of the service classes and attributes supported by SDP. Only service classes that directly support the SDP server are included in this document. Additional service classes will be defined in other documents and possibly in future revisions of this document. Also, it is expected that additional attributes will be discovered that are applicable to a broad set of services; these may be added to the list of Universal attributes in future revisions of this document.

5.1 UNIVERSAL ATTRIBUTE DEFINITIONS

Universal attributes are those service attributes whose definitions are common to all service records. Note that this does not mean that every service record must contain values for all of these service attributes. However, if a service record has a service attribute with an attribute ID allocated to a universal attribute, the attribute value must conform to the universal attribute's definition.

Only two attributes are required to exist in every service record instance. They are the ServiceRecordHandle (attribute ID 0x0000) and the ServiceClassIDList (attribute ID 0x0001). All other service attributes are optional within a service record.

5.1.1 ServiceRecordHandle Attribute

Attribute Name	Attribute ID	Attribute Value Type
ServiceRecordHandle	0x0000	32-bit unsigned integer

Description:

A service record handle is a 32-bit number that uniquely identifies each service record within an SDP server. It is important to note that, in general, each handle is unique only within each SDP server. If SDP server S1 and SDP server S2 both contain identical service records (representing the same service), the service record handles used to reference these identical service records are completely independent. The handle used to reference the service on S1 will, in general, be meaningless if presented to S2.

5.1.2 ServiceClassIDList Attribute

Attribute Name	Attribute ID	Attribute Value Type
ServiceClassIDList	0x0001	Data Element Sequence

Description:

The ServiceClassIDList attribute consists of a data element sequence in which each data element is a UUID representing the service classes that a given service record conforms to. The UUIDs are listed in order from the most specific class to the most general class. The ServiceClassIDList must contain at least one service class UUID.

5.1.3 ServiceRecordState Attribute

Attribute Name	Attribute ID	Attribute Value Type
ServiceRecordState	0x0002	32-bit unsigned integer

Description:

The ServiceRecordState is a 32-bit integer that is used to facilitate caching of ServiceAttributes. If this attribute is contained in a service record, its value is guaranteed to change when any other attribute value is added to, deleted from or changed within the service record. This permits a client to check the value of this single attribute. If its value has not changed since it was last checked, the client knows that no other attribute values within the service record have changed.

5.1.4 ServiceID Attribute

Attribute Name	Attribute ID	Attribute Value Type
ServiceID	0x0003	UUID

Description:

The ServiceID is a UUID that universally and uniquely identifies the service instance described by the service record. This service attribute is particularly useful if the same service is described by service records in more than one SDP server.

5.1.5 ProtocolDescriptorList Attribute

Attribute Name	Attribute ID	Attribute Value Type
ProtocolDescriptorList	0x0004	Data Element Sequence or Data Element Alternative

Description:

The ProtocolDescriptorList attribute describes one or more protocol stacks that may be used to gain access to the service described by the service record.

If the ProtocolDescriptorList describes a single stack, it takes the form of a data element sequence in which each element of the sequence is a protocol descriptor. Each protocol descriptor is, in turn, a data element sequence whose first element is a UUID identifying the protocol and whose successive elements are protocol-specific parameters. Potential protocol-specific parameters are a protocol version number and a connection-port number. The protocol descriptors are listed in order from the lowest layer protocol to the highest layer protocol used to gain access to the service.

If it is possible for more than one kind of protocol stack to be used to gain access to the service, the ProtocolDescriptorList takes the form of a data element alternative where each member is a data element sequence as described in the previous paragraph.

Protocol Descriptors

A protocol descriptor identifies a communications protocol and provides protocol-specific parameters. A protocol descriptor is represented as a data element sequence. The first data element in the sequence must be the UUID that identifies the protocol. Additional data elements optionally provide protocol-specific information, such as the L2CAP protocol/service multiplexer (PSM) and the RFCOMM server channel number (CN) shown below.

ProtocolDescriptorList Examples

These examples are intended to be illustrative. The parameter formats for each protocol are not defined within this specification.

In the first two examples, it is assumed that a single RFCOMM instance exists on top of the L2CAP layer. In this case, the L2CAP protocol specific information (PSM) points to the single instance of RFCOMM. In the last example, two different and independent RFCOMM instances are available on top of the L2CAP layer. In this case, the L2CAP protocol specific information (PSM) points to a distinct identifier that distinguishes each of the RFCOMM instances. According to the L2CAP specification, this identifier takes values in the range 0x1000-0xFFFF.

IrDA-like printer

((L2CAP, PSM=RFCOMM), (RFCOMM, CN=1), (PostscriptStream))

IP Network Printing

((L2CAP, PSM=RFCOMM), (RFCOMM, CN=2), (PPP), (IP), (TCP), (IPP))

Synchronization Protocol Descriptor Example

((L2CAP, PSM=0x1001), (RFCOMM, CN=1), (Obex), (vCal))

((L2CAP, PSM=0x1002), (RFCOMM, CN=1), (Obex), (otherSynchronisationApplication))

5.1.6 BrowseGroupList Attribute

Attribute Name	Attribute ID	Attribute Value Type
BrowseGroupList	0x0005	Data Element Sequence

Description:

The BrowseGroupList attribute consists of a data element sequence in which each element is a UUID that represents a browse group to which the service record belongs. The top-level browse group ID, called PublicBrowseRoot and representing the root of the browsing hierarchy, has the value 00001002-0000-1000-8000-00805F9B34FB (UUID16: 0x1002) from the Bluetooth Assigned Numbers document.

5.1.7 LanguageBaseAttributeIDList Attribute

Attribute Name	Attribute ID	Attribute Value Type
LanguageBaseAttributeIDList	0x0006	Data Element Sequence

Description:

In order to support human-readable attributes for multiple natural languages in a single service record, a base attribute ID is assigned for each of the natural languages used in a service record. The human-readable universal attributes are then defined with an attribute ID offset from each of these base values, rather than with an absolute attribute ID.

The LanguageBaseAttributeIDList attribute is a list in which each member contains a language identifier, a character encoding identifier, and a base attribute

ID for each of the natural languages used in the service record. The LanguageBaseAttributeIDList attribute consists of a data element sequence in which each element is a 16-bit unsigned integer. The elements are grouped as triplets (threes).

The first element of each triplet contains an identifier representing the natural language. The language is encoded according to ISO 639:1988 (E/F): “Code for the representation of names of languages”.

The second element of each triplet contains an identifier that specifies a character encoding used for the language. Values for character encoding can be found in IANA's database[2], and have the values that are referred to as MIBEnum values. The recommended character encoding is UTF-8.

The third element of each triplet contains an attribute ID that serves as the base attribute ID for the natural language in the service record. Different service records within a server may use different base attribute ID values for the same language.

To facilitate the retrieval of human-readable universal attributes in a principal language, the base attribute ID value for the primary language supported by a service record must be 0x0100. Also, if a LanguageBaseAttributeIDList attribute is contained in a service record, the base attribute ID value contained in its first element must be 0x0100.

5.1.8 ServiceInfoTimeToLive Attribute

Attribute Name	Attribute ID	Attribute Value Type
ServiceInfoTimeToLive	0x0007	32-bit unsigned integer

Description:

The ServiceTimeToLive attribute is a 32-bit integer that contains the number of seconds for which the information in a service record is expected to remain valid and unchanged. This time interval is measured from the time that the attribute value is retrieved from the SDP server. This value does not imply a guarantee that the service record will remain available or unchanged. It is simply a hint that a client may use to determine a suitable polling interval to revalidate the service record contents.

2. See http://www.isi.edu/in-notes/iana/assignments/character-sets

5.1.9 ServiceAvailability Attribute

Attribute Name	Attribute ID	Attribute Value Type
ServiceAvailability	0x0008	8-bit unsigned integer

Description:

The ServiceAvailability attribute is an 8-bit unsigned integer that represents the relative ability of the service to accept additional clients. A value of 0xFF indicates that the service is not currently in use and is thus fully available, while a value of 0x00 means that the service is not accepting new clients. For services that support multiple simultaneous clients, intermediate values indicate the relative availability of the service on a linear scale.

For example, a service that can accept up to 3 clients should provide ServiceAvailability values of 0xFF, 0xAA, 0x55, and 0x00 when 0, 1, 2, and 3 clients, respectively, are utilizing the service. The value 0xAA is approximately (2/3) * 0xFF and represents 2/3 availability, while the value 0x55 is approximately (1/3)*0xFF and represents 1/3 availability. Note that the availability value may be approximated as

(1 - (current_number_of_clients / maximum_number_of_clients)) * 0xFF

When the maximum number of clients is large, this formula must be modified to ensure that ServiceAvailability values of 0x00 and 0xFF are reserved for their defined meanings of unavailability and full availability, respectively.

Note that the maximum number of clients a service can support may vary according to the resources utilized by the service's current clients.

A non-zero value for ServiceAvailability does not guarantee that the service will be available for use. It should be treated as a hint or an approximation of availability status.

5.1.10 BluetoothProfileDescriptorList Attribute

Attribute Name	Attribute ID	Attribute Value Type
BluetoothProfileDescriptorList	0x0009	Data Element Sequence

Description:

The BluetoothProfileDescriptorList attribute consists of a data element sequence in which each element is a profile descriptor that contains information about a Bluetooth profile to which the service represented by this service record conforms. Each profile descriptor is a data element sequence whose

first element is the UUID assigned to the profile and whose second element is a 16-bit profile version number.

Each version of a profile is assigned a 16-bit unsigned integer profile version number, which consists of two 8-bit fields. The higher-order 8 bits contain the major version number field and the lower-order 8 bits contain the minor version number field. The initial version of each profile has a major version of 1 and a minor version of 0. When upward compatible changes are made to the profile, the minor version number will be incremented. If incompatible changes are made to the profile, the major version number will be incremented.

5.1.11 DocumentationURL Attribute

Attribute Name	Attribute ID	Attribute Value Type
DocumentationURL	0x000A	URL

Description:

This attribute is a URL which points to documentation on the service described by a service record.

5.1.12 ClientExecutableURL Attribute

Attribute Name	Attribute ID	Attribute Value Type
ClientExecutableURL	0x000B	URL

Description:

This attribute contains a URL that refers to the location of an application that may be used to utilize the service described by the service record. Since different operating environments require different executable formats, a mechanism has been defined to allow this single attribute to be used to locate an executable that is appropriate for the client device's operating environment. In the attribute value URL, the first byte with the value 0x2A (ASCII character '*') is to be replaced by the client application with a string representing the desired operating environment before the URL is to be used.

The list of standardized strings representing operating environments is contained in the Bluetooth Assigned Numbers document.

For example, assume that the value of the ClientExecutableURL attribute is http://my.fake/public/*/client.exe. On a device capable of executing SH3 WindowsCE files, this URL would be changed to http://my.fake/public/sh3-microsoft-wince/client.exe. On a device capable of executing Windows 98 binaries, this URL would be changed to http://my.fake/public/i86-microsoft-win98/client.exe.

5.1.13 IconURL Attribute

Attribute Name	Attribute ID	Attribute Value Type
IconURL	0x000C	URL

Description:

This attribute contains a URL that refers to the location of an icon that may be used to represent the service described by the service record. Since different hardware devices require different icon formats, a mechanism has been defined to allow this single attribute to be used to locate an icon that is appropriate for the client device. In the attribute value URL, the first byte with the value 0x2A (ASCII character '*') is to be replaced by the client application with a string representing the desired icon format before the URL is to be used.

The list of standardized strings representing icon formats is contained in the Bluetooth Assigned Numbers document.

For example, assume that the value of the IconURL attribute is http://my.fake/public/icons/*. On a device that prefers 24 x 24 icons with 256 colors, this URL would be changed to http://my.fake/public/icons/24x24x8.png. On a device that prefers 10 x 10 monochrome icons, this URL would be changed to http://my.fake/public/icons/10x10x1.png.

5.1.14 ServiceName Attribute

Attribute Name	Attribute ID Offset	Attribute Value Type
ServiceName	0x0000	String

Description:

The ServiceName attribute is a string containing the name of the service represented by a service record. It should be brief and suitable for display with an Icon representing the service. The offset 0x0000 must be added to the attribute ID base (contained in the LanguageBaseAttributeIDList attribute) in order to compute the attribute ID for this attribute.

5.1.15 ServiceDescription Attribute

Attribute Name	Attribute ID Offset	Attribute Value Type
ServiceDescription	0x0001	String

Description:

This attribute is a string containing a brief description of the service. It should be less than 200 characters in length. The offset 0x0001 must be added to the attribute ID base (contained in the LanguageBaseAttributeIDList attribute) in order to compute the attribute ID for this attribute.

5.1.16 ProviderName Attribute

Attribute Name	Attribute ID Offset	Attribute Value Type
ProviderName	0x0002	String

Description:

This attribute is a string containing the name of the person or organization providing the service. The offset 0x0002 must be added to the attribute ID base (contained in the LanguageBaseAttributeIDList attribute) in order to compute the attribute ID for this attribute.

5.1.17 Reserved Universal Attribute IDs

Attribute IDs in the range of 0x000D-0x01FF are reserved.

5.2 SERVICEDISCOVERYSERVER SERVICE CLASS ATTRIBUTE DEFINITIONS

This service class describes service records that contain attributes of service discovery server itself. The attributes listed in this section are only valid if the ServiceClassIDList attribute contains the ServiceDiscoveryServerServiceClassID. Note that all of the universal attributes may be included in service records of the ServiceDiscoveryServer class.

5.2.1 ServiceRecordHandle Attribute

Described in the universal attribute definition for ServiceRecordHandle.

Value

A 32-bit integer with the value 0x000000000.

5.2.2 ServiceClassIDList Attribute

Described in the universal attribute definition for ServiceClassIDList.

Value

A UUID representing the ServiceDiscoveryServerServiceClassID.

5.2.3 VersionNumberList Attribute

Attribute Name	Attribute ID	Attribute Value Type
VersionNumberList	0x0200	Data Element Sequence

Description:

The VersionNumberList is a data element sequence in which each element of the sequence is a version number supported by the SDP server.

A version number is a 16-bit unsigned integer consisting of two fields. The higher-order 8 bits contain the major version number field and the low-order 8 bits contain the minor version number field. The initial version of SDP has a major version of 1 and a minor version of 0. When upward compatible changes are made to the protocol, the minor version number will be incremented. If incompatible changes are made to SDP, the major version number will be incremented. This guarantees that if a client and a server support a common major version number, they can communicate if each uses only features of the specification with a minor version number that is supported by both client and server.

5.2.4 ServiceDatabaseState Attribute

Attribute Name	Attribute ID	Attribute Value Type
ServiceDatabaseState	0x0201	32-bit unsigned integer

Description:

The ServiceDatabaseState is a 32-bit integer that is used to facilitate caching of service records. If this attribute exists, its value is guaranteed to change when any of the other service records are added to or deleted from the server's database. If this value has not changed since the last time a client queried its value, the client knows that a) none of the other service records maintained by the SDP server have been added or deleted; and b) any service record handles acquired from the server are still valid. A client should query this attribute's value when a connection to the server is established, prior to using any service record handles acquired during a previous connection.

Note that the ServiceDatabaseState attribute does not change when existing service records are modified, including the addition, removal, or modification of service attributes. A service record's ServiceRecordState attribute indicates when that service record is modified.

5.2.5 Reserved Attribute IDs

Attribute IDs in the range of 0x0202-0x02FF are reserved.

5.3 BROWSEGROUPDESCRIPTOR SERVICE CLASS ATTRIBUTE DEFINITIONS

This service class describes the ServiceRecord provided for each BrowseGroupDescriptor service offered on a Bluetooth device. The attributes listed in this section are only valid if the ServiceClassIDList attribute contains the BrowseGroupDescriptorServiceClassID. Note that all of the universal attributes may be included in service records of the BrowseGroupDescriptor class.

5.3.1 ServiceClassIDList Attribute

Described in the universal attribute definition for ServiceClassIDList.

Value

A UUID representing the BrowseGroupDescriptorServiceClassID.

5.3.2 GroupID Attribute

Attribute Name	Attribute ID	Attribute Value Type
GroupID	0x0200	UUID

Description:

This attribute contains a UUID that can be used to locate services that are members of the browse group that this service record describes.

5.3.3 Reserved Attribute IDs

Attribute IDs in the range of 0x0201-0x02FF are reserved.

APPENDIX A – BACKGROUND INFORMATION

A.1. Service Discovery

As computing continues to move to a network-centric model, finding and making use of services that may be available in the network becomes increasingly important. Services can include common ones such as printing, paging, FAXing, and so on, as well as various kinds of information access such as teleconferencing, network bridges and access points, eCommerce facilities, and so on — most any kind of service that a server or service provider might offer. In addition to the need for a standard way of discovering available services, there are other considerations: getting access to the services (finding and obtaining the protocols, access methods, “drivers” and other code necessary to utilize the service), controlling access to the services, advertising the services, choosing among competing services, billing for services, and so on. This problem is widely recognized; many companies, standards bodies and consortia are addressing it at various levels in various ways. Service Location Protocol (SLP), Jini™, and Salutation™, to name just a few, all address some aspect of service discovery.

A.2. Bluetooth Service Discovery

Bluetooth Service Discovery Protocol (SDP) addresses service discovery specifically for the Bluetooth environment. It is optimized for the highly dynamic nature of Bluetooth communications. SDP focuses primarily on discovering services available from or through Bluetooth devices. SDP does not define methods for accessing services; once services are discovered with SDP, they can be accessed in various ways, depending upon the service. This might include the use of other service discovery and access mechanisms such as those mentioned above; SDP provides a means for other protocols to be used along with SDP in those environments where this can be beneficial. While SDP can coexist with other service discovery protocols, it does not require them. In Bluetooth environments, services can be discovered using SDP and can be accessed using other protocols defined by Bluetooth.

APPENDIX B – EXAMPLE SDP TRANSACTIONS

The following are simple examples of typical SDP transactions. These are meant to be illustrative of SDP flows. The examples do not consider:

- Caching (in a caching system, the SDP client would make use of the ServiceRecordState and ServiceDatabaseState attributes);
- Service availability (if this is of interest, the SDP client should use the ServiceAvailability and/or ServiceTimeToLive attributes);
- SDP versions (the VersionNumberList attribute could be used to determine compatible SDP versions);
- SDP Error Responses (an SDP error response is possible for any SDP request that is in error); and
- Communication connection (the examples assume that an L2CAP connection is established).

The examples are meant to be illustrative of the protocol. The format used is `ObjectName[ObjectSizeInBytes] {SubObjectDefinitions},` but this is not meant to illustrate an interface. The `ObjectSizeInBytes` is the size of the object in decimal. The `SubObjectDefinitions` (inside of `{}` characters) are components of the immediately enclosing object. Hexadecimal values shown as lower-case letters, such as for transaction IDs and service handles, are variables (the particular value is not important for the illustration, but each such symbol always represents the same value). Comments are included in this manner: `/* comment text */`.

Numeric values preceded by “0x” are hexadecimal, while those preceded by “0b” are binary. All other numeric values are decimal.

B.1. SDP Example 1 – ServiceSearchRequest

The first example is that of an SDP client searching for a generic printing service. The client does not specify a particular type of printing service. In the example, the SDP server has two available printing services. The transaction illustrates:

1. SDP client to SDP server: SDP_ServiceSearchRequest, specifying the PrinterServiceClassID (represented as a DataElement with a 32-bit UUID value of `ppp...ppp`) as the only element of the ServiceSearchPattern. The PrinterServiceClassID is assumed to be a 32-bit UUID and the data element type for it is illustrated. The TransactionID is illustrated as `tttt`.
2. SDP server to SDP client: SDP_ServiceSearchResponse, returning handles to two printing services, represented as `qqqqqqqq` for the first printing service and `rrrrrrrr` for the second printing service. The Transaction ID is the same value as supplied by the SDP client in the corresponding request (ττττ).

```
/* Sent from SDP Client to SDP server */
SDP_ServiceSearchRequest[15] {
  PDUID[1] {
    0x02
  }
  TransactionID[2] {
    0xtttt
  }
  ParameterLength[2] {
    0x000A
  }
  ServiceSearchPattern[7] {
    DataElementSequence[7] {
      0b00110 0b101 0x05
      UUID[5] {
        /* PrinterServiceClassID */
        0b00011 0b010 0xpppppppp
      }
    }
  }
  MaximumServiceRecordCount[2] {
    0x0003
  }
  ContinuationState[1] {
    /* no continuation state */
    0x00
  }
}


/* Sent from SDP server to SDP client */
SDP_ServiceSearchResponse[18] {
  PDUID[1] {
    0x03
  }
  TransactionID[2] {
    0xtttt
  }
  ParameterLength[2] {
    0x000D
  }
  TotalServiceRecordCount[2] {
    0x0002
  }
  CurrentServiceRecordCount[2] {
    0x0002
  }
  ServiceRecordHandleList[8] {
    /* print service 1 handle */
    0xqqqqqqqq
    /* print service 2 handle */
    0xrrrrrrrr
  }
  ContinuationState[1] {
    /* no continuation state */
```

```
    0x00
  }
}
```

B.2. SDP Example 2 – ServiceAttributeTransaction

The second example continues the first example. In Example 1, the SDP client obtained handles to two printing services. In Example 2, the client uses one of those service handles to obtain the ProtocolDescriptorList attribute for that printing service. The transaction illustrates:

1. SDP client to SDP server: SDP_ServiceAttributeRequest, presenting the previously obtained service handle (the one denoted as qqqqqqqq) and specifying the ProtocolDescriptorList attribute ID (AttributeID 0x0004) as the only attribute requested (other attributes could be retrieved in the same transaction if desired). The TransactionID is illustrated as uuuu to distinguish it from the TransactionID of Example 1.
2. SDP server to SDP client: SDP_ServiceAttributeResponse, returning the ProtocolDescriptorList for the specified printing service. This protocol stack is assumed to be ((L2CAP), (RFCOMM, 2), (PostscriptStream)). The ProtocolDescriptorList is a data element sequence in which each element is, in turn, a data element sequence whose first element is a UUID representing the protocol, and whose subsequent elements are protocol-specific parameters. In this example, one such parameter is included for the RFCOMM protocol, an 8-bit value indicating RFCOMM server channel 2. The Transaction ID is the same value as supplied by the SDP client in the corresponding request (uuuu). The Attributes returned are illustrated as a data element sequence where the protocol descriptors are 32-bit UUIDs and the RFCOMM server channel is a data element with an 8-bit value of 2.

```
 /* Sent from SDP Client to SDP server */
SDP_ServiceAttributeRequest[17] {
  PDUID[1] {
    0x04
  }
  TransactionID[2] {
    0xuuuu
  }
  ParameterLength[2] {
    0x000C
  }
  ServiceRecordHandle[4] {
    0xqqqqqqqq
  }
  MaximumAttributeByteCount[2] {
    0x0080
  }
  AttributeIDList[5] {
    DataElementSequence[5] {
```

```
      0b00110 0b101 0x03
      AttributeID[3] {
        0b00001 0b001 0x0004
      }
    }
  }
  ContinuationState[1] {
    /* no continuation state */
    0x00
  }
}

/* Sent from SDP server to SDP client */
SDP_ServiceAttributeResponse[38] {
  PDUID[1] {
    0x05
  }
  TransactionID[2] {
    0xuuuu
  }
  ParameterLength[2] {
    0x0021
  }
  AttributeListByteCount[2] {
    0x001E
  }
  AttributeList[30] {
    DataElementSequence[30] {
      0b00110 0b101 0x1C
      Attribute[28] {
        AttributeID[3] {
          0b00001 0b001 0x0004
        }
        AttributeValue[25] {
          /* ProtocolDescriptorList */
          DataElementSequence[25] {
            0b00110 0b101 0x17
            /* L2CAP protocol descriptor */
            DataElementSequence[7] {
              0b00110 0b101 0x05
              UUID[5] {
                /* L2CAP Protocol UUID */
                0b00011 0b010 <32-bit L2CAP UUID>
              }
            }
            /* RFCOMM protocol descriptor */
            DataElementSequence[9] {
              0b00110 0b101 0x07
              UUID[5] {
                /* RFCOMM Protocol UUID */
                0b00011 0b010 <32-bit RFCOMM UUID>
              }
              /* parameter for server 2 */
              Uint8[2] {
                0b00001 0b000 0x02
              }
            }
```

```
            /* PostscriptStream protocol descriptor */
            DataElementSequence[7] {
              0b00110 0b101 0x05
              UUID[5] {
                /* PostscriptStream Protocol UUID */
                0b00011 0b010 <32-bit PostscriptStream UUID>
              }
            }
          }
        }
      }
    }
  }
  ContinuationState[1] {
    /* no continuation state */
    0x00
  }
}
```

B.3. SDP Example 3 – ServiceSearchAttributeTransaction

The third example is a form of service browsing, although it is not generic browsing in that it does not make use of SDP browse groups. Instead, an SDP client is searching for available Synchronization services that can be presented to the user for selection. The SDP client does not specify a particular type of synchronization service. In the example, the SDP server has three available synchronization services: an address book synchronization service and a calendar synchronization service (both from the same provider), and a second calendar synchronization service from a different provider. The SDP client is retrieving the same attributes for each of these services; namely, the data formats supported for the synchronization service (vCard, vCal, ICal, etc.) and those attributes that are relevant for presenting information to the user about the services. Also assume that the maximum size of a response is 400 bytes. Since the result is larger than this, the SDP client will repeat the request supplying a continuation state parameter to retrieve the remainder of the response. The transaction illustrates:

1. SDP client to SDP server: SDP_ServiceSearchAttributeRequest, specifying the generic SynchronisationServiceClassID (represented as a data element whose 32-bit UUID value is sss...sss) as the only element of the ServiceSearchPattern. The SynchronisationServiceClassID is assumed to be a 32-bit UUID. The requested attributes are the ServiceRecordHandle (attribute ID 0x0000), ServiceClassIDList (attribute ID 0x0001), IconURL (attribute ID 0x000C), ServiceName (attribute ID 0x0100), ServiceDescription (attribute ID 0x0101), and ProviderName (attributeID 0x0102) attributes; as well as the service-specific SupportedDataStores (AttributeID 0x0301). Since the first two attribute IDs (0x0000 and 0x0001) and three other attribute IDs(0x0100, 0x0101, and 0x0102 are consecutive, they are specified as attribute ranges. The TransactionID is illustrated as vvvv to distinguish it from the TransactionIDs of the other Examples.

   Note that values in the service record's primary language are requested for the text attributes (ServiceName, ServiceDescription and ProviderName) so that absolute attribute IDs may be used, rather than adding offsets to a base obtained from the LanguageBaseAttributeIDList attribute.

2. SDP server to SDP client: SDP_ServiceSearchAttributeResponse, returning the specified attributes for each of the three synchronization services. In the example, each ServiceClassIDList is assumed to contain a single element, the generic SynchronisationServiceClassID (a 32-bit UUID represented as sss...sss). Each of the other attributes contain illustrative data in the example (the strings have illustrative text; the icon URLs are illustrative, for each of the respective three synchronization services; and the SupportedDataStore attribute is represented as an unsigned 8-bit integer where 0x01 = vCard2.1, 0x02 = vCard3.0, 0x03 = vCal1.0 and 0x04 = iCal). Note that one of the service records (the third for which data is returned) has no ServiceDescription attribute. The attributes are returned as a data element sequence, where each element is in turn a data element sequence repre-

senting a list of attributes. Within each attribute list, the ServiceClassIDList is a data element sequence while the remaining attributes are single data elements. The Transaction ID is the same value as supplied by the SDP client in the corresponding request (0xvvvv). Since the entire result cannot be returned in a single response, a non-null continuation state is returned in this first response.

Note that the total length of the initial data element sequence (487 in the example) is indicated in the first response, even though only a portion of this data element sequence (368 bytes in the example, as indicated in the AttributeLists byte count) is returned in the first response. The remainder of this data element sequence is returned in the second response (without an additional data element header).

3. SDP client to SDP server: SDP_ServiceSearchAttributeRequest, with the same parameters as in step 1, except that the continuation state received from the server in step 2 is included as a request parameter. The TransactionID is changed to 0xwww to distinguish it from previous request.
4. SDP server to SDP client: SDP_ServiceSearchAttributeResponse, with the remainder of the result computed in step 2 above. Since all of the remaining result fits in this second response, a null continuation state is included.

```
/* Part 1 -- Sent from SDP Client to SDP server */
SdpSDP_ServiceSearchAttributeRequest[33] {
  PDUID[1] {
    0x06
  }
  TransactionID[2] {
    0xvvvv
  }
  ParameterLength[2] {
    0x001B
  }
  ServiceSearchPattern[7] {
    DataElementSequence[7] {
      0b00110 0b101 0x05
      UUID[5] {
        /* SynchronisationServiceClassID */
        0b00011 0b010 0xssssssss
      }
    }
  }
  MaximumAttributeByteCount[2] {
    0x0190
  }
AttributeIDList[18] {
    DataElementSequence[18] {
      0b00110 0b101 0x10
      AttributeIDRange[5] {
        0b00001 0b010 0x00000001
      }
      AttributeID[3] {
        0b00001 0b001 0x000C
```

```
      }
      AttributeIDRange[5] {
        0b00001 0b010 0x01000102
      }
      AttributeID[3] {
        0b00001 0b001 0x0301
      }    }
  }
  ContinuationState[1] {
    /* no continuation state */
    0x00
  }
}

/* Part 2 -- Sent from SDP server to SDP client */
SdpSDP_ServiceSearchAttributeResponse[384] {
  PDUID[1] {
    0x07
  }
  TransactionID[2] {
    0xvvvv
  }
  ParameterLength[2] {
    0x017B
  }
  AttributeListByteCount[2] {
    0x0170
  }
  AttributeLists[368] {
    DataElementSequence[487] {
      0b00110 0b110 0x01E4
      DataElementSequence[178] {
        0b00110 0b101 0xB0
        Attribute[8] {
          AttributeID[3] {
            0b00001 0b001 0x0000
          }
          AttributeValue[5] {
            /* service record handle */
            0b00001 0b010 0xhhhhhhhh
          }
        }
        Attribute[10] {
          AttributeID[3] {
            0b00001 0b001 0x0001
          }
AttributeValue[7] {
            DataElementSequence[7] {
              0b00110 0b101 0x05
              UUID[5] {
                /* SynchronisationServiceClassID */
                0b00011 0b010 0xssssssss
              }
            }
          }
        }
        Attribute[35] {
```

```
        AttributeID[3] {
          0b00001 0b001 0x000C
        }
        AttributeValue[32] {
          /* IconURL; '*' replaced by client application */
          0b01000 0b101 0x1E
          "http://Synchronisation/icons/*"
        }
      }
      Attribute[22] {
        AttributeID[3] {
          0b00001 0b001 0x0100
        }
        AttributeValue[19] {
          /* service name */
          0b00100 0b101 0x11
          "Address Book Sync"
        }
      }
      Attribute[59] {
        AttributeID[3] {
          0b00001 0b001 0x0101
        }
        AttributeValue[56] {
          /* service description */
          0b00100 0b101 0x36
          "Synchronisation Service for"
          " vCard Address Book Entries"
        }
      }
      Attribute[37] {
        AttributeID[3] {
          0b00001 0b001 0x0102
        }
        AttributeValue[34] {
          /* service provider */
          0b00100 0b101 0x20
          "Synchronisation Specialists Inc."
        }
      }
      Attribute[5] {
        AttributeID[3] {
          0b00001 0b001 0x0301
        }
        AttributeValue[2] {
          /* Supported Data Store 'phonebook' */
          0b00001 0b000 0x01
        }
      }
    }
    DataElementSequence[175] {
      0b00110 0b101 0xAD
      Attribute[8] {
        AttributeID[3] {
          0b00001 0b001 0x0000
        }
        AttributeValue[5] {
```

```
      /* service record handle */
      0b00001 0b010 0xmmmmmmmm
    }
  }
  Attribute[10] {
    AttributeID[3] {
      0b00001 0b001 0x0001
    }
              AttributeValue[7] {
      DataElementSequence[7] {
        0b00110 0b101 0x05
        UUID[5] {
          /* SynchronisationServiceClassID */
          0b00011 0b010 0xssssssss
        }
      }
    }
  }
  Attribute[35] {
    AttributeID[3] {
      0b00001 0b001 0x000C
    }
    AttributeValue[32] {
      /* IconURL; '*' replaced by client application */
      0b01000 0b101 0x1E
      "http://Synchronisation/icons/*"
    }
  }
  Attribute[21] {
    AttributeID[3] {
      0b00001 0b001 0x0100
    }
    AttributeValue[18] {
      /* service name */
      0b00100 0b101 0x10
      "Appointment Sync"
    }
  }
  Attribute[57] {
    AttributeID[3] {
      0b00001 0b001 0x0101
    }
    AttributeValue[54] {
      /* service description */
      0b00100 0b101 0x34
      "Synchronisation Service for"
      " vCal Appointment Entries"
    }
  }
  Attribute[37] {
    AttributeID[3] {
      0b00001 0b001 0x0102
    }
    AttributeValue[34] {
      /* service provider */
      0b00100 0b101 0x20
      "Synchronisation Specialists Inc."
```

```
          }
        }
        Attribute[5] {
          AttributeID[3] {
            0b00001 0b001 0x0301
          }
          AttributeValue[2] {
            /* Supported Data Store 'calendar' */
            0b00001 0b000 0x03
          }
        }
      }
    /* } Data element sequence of attribute lists */
    /* is not completed in this PDU. */
  }
  ContinuationState[9] {
    /* 8 bytes of continuation state */
    0x08 0xzzzzzzzzzzzzzzzz
  }
}

/* Part 3 -- Sent from SDP Client to SDP server */
SdpSDP_ServiceSearchAttributeRequest[41] {
  PDUID[1] {
    0x06
  }
  TransactionID[2] {
    0xwwww
  }
  ParameterLength[2] {
    0x0024
  }
  ServiceSearchPattern[7] {
    DataElementSequence[7] {
      0b00110 0b101 0x05
      UUID[5] {
        /* SynchronisationServiceClassID */
        0b00011 0b010 0xssssssss
      }
    }
  }
  MaximumAttributeByteCount[2] {
    0x0180
  }
    AttributeIDList[18] {
    DataElementSequence[18] {
      0b00110 0b101 0x10
      AttributeIDRange[5] {
        0b00001 0b010 0x00000001
      }
      AttributeID[3] {
        0b00001 0b001 0x000C
      }
      AttributeIDRange[5] {
        0b00001 0b010 0x01000102
      }
      AttributeID[3] {
```

```
        0b00001 0b001 0x0301
      }
    }
  }
  ContinuationState[9] {
    /* same 8 bytes of continuation state */
    /* received in part 2 */
    0x08 0xzzzzzzzzzzzzzzz
  }
}

Part 4 -- Sent from SDP server to SDP client

SdpSDP_ServiceSearchAttributeResponse[115] {
  PDUID[1] {
    0x07
  }
  TransactionID[2] {
    0xwwww
  }
  ParameterLength[2] {
    0x006E
  }
  AttributeListByteCount[2] {
    0x006B
  }
  AttributeLists[107] {
    /* Continuing the data element sequence of */
    /* attribute lists begun in Part 2. */
      DataElementSequence[107] {
        0b00110 0b101 0x69
        Attribute[8] {
          AttributeID[3] {
            0b00001 0b001 0x0000
          }
          AttributeValue[5] {
            /* service record handle */
            0b00001 0b010 0xffffffff
          }
        }
        Attribute[10] {
          AttributeID[3] {
            0b00001 0b001 0x0001
          }
          AttributeValue[7] {
            DataElementSequence[7] {
              0b00110 0b101 0x05
              UUID[5] {
                /* SynchronisationServiceClassID */
                0b00011 0b010 0xssssssss
              }
            }
          }
        }
        Attribute[35] {
          AttributeID[3] {
            0b00001 0b001 0x000C
```

```
          }
          AttributeValue[32] {
            /* IconURL; '*' replaced by client application */
            0b01000 0b101 0x1E
            "http://DevManufacturer/icons/*"
          }
        }
        Attribute[18] {
          AttributeID[3] {
            0b00001 0b001 0x0100
          }
          AttributeValue[15] {
            /* service name */
            0b00100 0b101 0x0D
            "Calendar Sync"
          }
        }
        Attribute[29] {
          AttributeID[3] {
            0b00001 0b001 0x0102
          }
          AttributeValue[26] {
            /* service provider */
            0b00100 0b101 0x18
            "Device Manufacturer Inc."
          }
        }
        Attribute[5] {
          AttributeID[3] {
            0b00001 0b001 0x0301
          }
          AttributeValue[2] {
            /* Supported Data Store 'calendar' */
            0b00001 0b000 0x03
          }
        }
      }
    /* This completes the data element sequence */
    /* of attribute lists begun in Part 2.
  }
  ContinuationState[1] {
    /* no continuation state */
    0x00
  }
}
```

Part F:1

RFCOMM with TS 07.10

Serial Port Emulation

This document specifies the RFCOMM protocol by specifying a subset of the ETSI TS 07.10 standard, along with some Bluetooth-specific adaptations

CONTENTS

1 Introduction 399
1.1 Overview 399
1.2 Device Types 399
1.3 Byte Ordering 400
2 RFCOMM Service Overview 401
2.1 RS-232 Control Signals 401
2.2 Null Modem Emulation 402
2.3 Multiple Emulated Serial Ports 403
2.3.1 Multiple Emulated Serial Ports between two Devices 403
2.3.2 Multiple Emulated Serial Ports and Multiple Bluetooth Devices 403
3 Service Interface Description 405
3.1 Service Definition Model 405
4 TS 07.10 Subset Supported by RFCOMM 406
4.1 Options and Modes 406
4.2 Frame Types 406
4.3 Commands 406
4.4 Convergence Layers 407
5 TS 07.10 Adaptations for RFCOMM 408
5.1 Media Adaptation 408
5.1.1 FCS calculation 408
5.1.2 P/F-Bit(Erratum 1053) 408
5.1.3 CR bit(Erratum 1510) 409
5.2 TS 07.10 Multiplexer Start-up and Closedown Procedure 410
5.2.1 Start-up procedure 410
5.2.2 Close-down procedure 410
5.2.3 Link loss handling 411
5.3 System Parameters 412
5.4 DLCI allocation with RFCOMM server channels 413
5.5 Multiplexer Control Commands 414
5.5.1 Remote Port Negotiation Command (RPN) 414
5.5.2 Remote Line Status Command (RLS) 415
5.5.3 DLC parameter negotiation (PN) 415
6 Flow Control 417
6.1 L2CAP Flow Control in Overview 417
6.2 Wired Serial Port Flow Control 417
6.3 (Erratum 1549)GSM TS 07.10 Flow Control 417
6.4 Port Emulation Entity Serial Flow Control 419

6.5 Credit based flow control(Erratum 1053) 420
6.5.1 Initial DLC Negotiation(Erratum1053) 420
6.5.2 DLC Operation(Erratum1053) 420
6.5.3 Other flow control aspects(Erratum1053) 421

7 Interaction with Other Entities 422
7.1 Port Emulation and Port Proxy Entities 422
7.1.1 Port Emulation Entity 422
7.1.2 Port Proxy Entity 422
7.2 Service Registration and Discovery 422
7.3 Lower Layer Dependencies 424
7.3.1 Reliability 424
7.3.2 Low power modes 424

8 References 425

9 Terms and Abbreviations 426

1 INTRODUCTION

The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10. This document does not contain a complete specification. Instead, references are made to the relevant parts of the TS 07.10 standard. Only a subset of the TS 07.10 standard is used, and some adaptations of the protocol are specified in this document. Furthermore, an RFCOMM - specific extension is added, in the form of a mandatory credit based flow control scheme.

1.1 OVERVIEW

RFCOMM is a simple transport protocol, with additional provisions for emulating the 9 circuits of RS-232 (EIATIA-232-E) serial ports.

The RFCOMM protocol supports up to 60 simultaneous connections between two Bluetooth devices. The number of connections that can be used simultaneously in a Bluetooth device is implementation-specific.

1.2 DEVICE TYPES

For the purposes of RFCOMM, a complete communication path involves two applications running on different devices (the communication endpoints) with a communication segment between them. Figure 1.1 shows the complete communication path. (In this context, the term *application* may mean other things than end-user application; e.g. higher layer protocols or other services acting on behalf of end-user applications.)

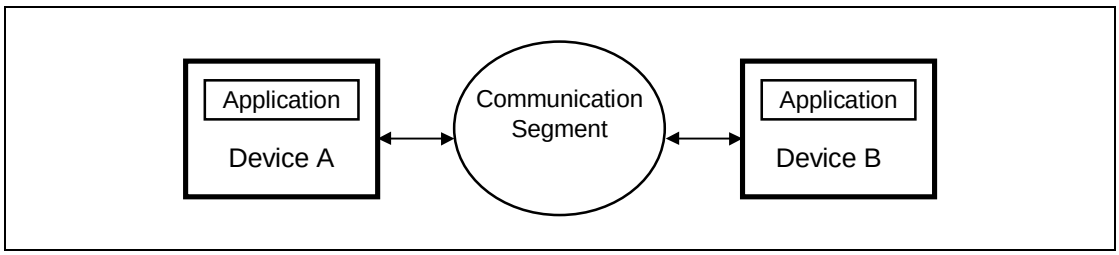


Figure 1.1: RFCOMM Communication Segment

RFCOMM is intended to cover applications that make use of the serial ports of the devices in which they reside. In the simple configuration, the communication segment is a Bluetooth link from one device to another (direct connect), see Figure 1.2. Where the communication segment is another network, Bluetooth wireless technology is used for the path between the device and a network connection device like a modem. RFCOMM is only concerned with the connection between the devices in the direct connect case, or between the device and a modem in the network case. RFCOMM can support other configurations, such as modules that communicate via Bluetooth wireless technology on one side and provide a wired interface on the other side, as shown in Figure

1.3. These devices are not really modems but offer a similar service. They are therefore not explicitly discussed here.

Basically two device types exist that RFCOMM must accommodate. Type 1 devices are communication end points such as computers and printers. Type 2 devices are those that are part of the communication segment; e.g. modems. Though RFCOMM does not make a distinction between these two device types in the protocol, accommodating both types of devices impacts the RFCOMM protocol.

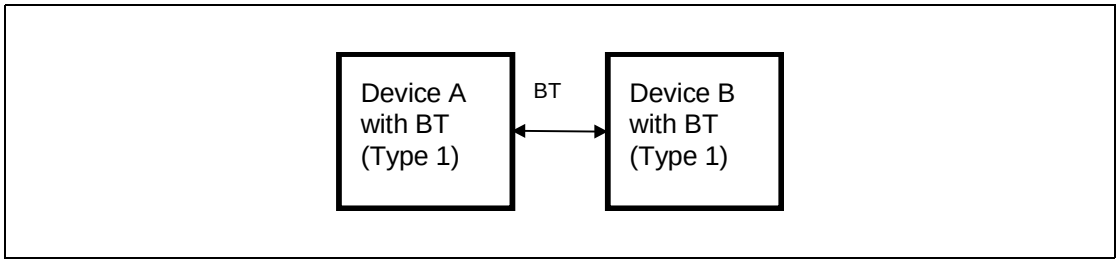


Figure 1.2: RFCOMM Direct Connect

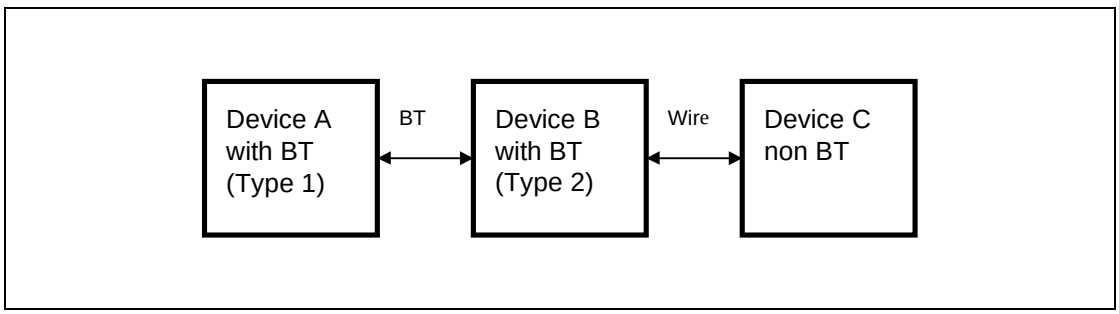


Figure 1.3: RFCOMM used with legacy COMM device

The information transferred between two RFCOMM entities has been defined to support both type 1 and type 2 devices. Some information is only needed by type 2 devices while other information is intended to be used by both. In the protocol, no distinction is made between type 1 and type 2. It is therefore up to the RFCOMM implementers to determine if the information passed in the RFCOMM protocol is of use to the implementation. Since the device is not aware of the type of the other device in the communication path, each must pass on all available information specified by the protocol.

1.3 BYTE ORDERING

This document uses the same byte ordering as the TS 07.10 specification; i.e. all binary numbers are in Least Significant Bit to Most Significant Bit order, reading from left to right.

2 RFCOMM SERVICE OVERVIEW

RFCOMM emulates RS-232 (EIATIA-232-E) serial ports. The emulation includes transfer of the state of the non-data circuits. RFCOMM has a built-in scheme for null modem emulation.

In the event that a baud rate is set for a particular port through the RFCOMM service interface, that will not affect the actual data throughput in RFCOMM; i.e. RFCOMM does not incur artificial rate limitation or pacing. However, if either device is a type 2 device (relays data onto other media), or if data pacing is done above the RFCOMM service interface in either or both ends, actual throughput will, on an average, reflect the baud rate setting.

RFCOMM supports emulation of multiple serial ports between two devices and also emulation of serial ports between multiple devices, see Section 2.3 on page 401.

2.1 RS-232 CONTROL SIGNALS

RFCOMM emulates the 9 circuits of an RS-232 interface. The circuits are listed below.

Pin	Circuit Name
102	Signal Common
103	Transmit Data (TD)
104	Received Data (RD)
105	Request to Send (RTS)
106	Clear to Send (CTS)
107	Data Set Ready (DSR)
108	Data Terminal Ready (DTR)
109	Data Carrier Detect (CD)
125	Ring Indicator (RI)

Table 2.1: Emulated RS-232 circuits in RFCOMM

2.2 NULL MODEM EMULATION

RFCOMM is based on TS 07.10. When it comes to transfer of the states of the non-data circuits, TS 07.10 does not distinguish between DTE and DCE devices. The RS-232 control signals are sent as a number of DTE/DCE independent signals, see Table 2.2.

TS 07.10 Signals	Corresponding RS-232 Control Signals
RTC	DSR, DTR
RTR	RTS, CTS
IC	RI
DV	DCD

Table 2.2: TS 07.10 Serial Port Control Signals

The way in which TS 07.10 transfers the RS-232 control signals creates an implicit null modem when two devices of the same kind are connected together. Figure 2.1 shows the null modem that is created when two DTE are connected via RFCOMM. No single null-modem cable wiring scheme works in all cases; however the null modem scheme provided in RFCOMM should work in most cases.

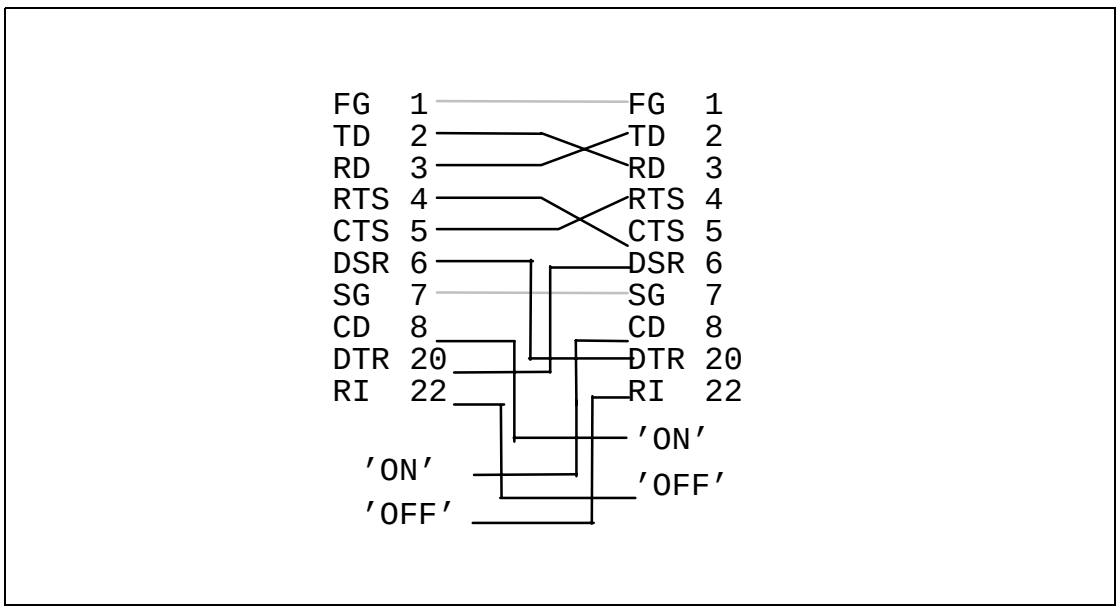


Figure 2.1: RFCOMM DTE–DTE Null Modem Emulation

2.3 MULTIPLE EMULATED SERIAL PORTS

2.3.1 Multiple Emulated Serial Ports between two Devices

Two Bluetooth devices using RFCOMM in their communication may open multiple emulated serial ports. RFCOMM supports up to 60 open emulated ports; however the number of ports that can be used in a device is implementation-specific.

A Data Link Connection Identifier (DLCI) [1] identifies an ongoing connection between a client and a server application. The DLCI is represented by 6 bits, but its usable value range is 2…61; in TS 07.10, DLCI 0 is the dedicated control channel, DLCI 1 is unusable due to the concept of Server Channels, and DLCI 62-63 is reserved. The DLCI is unique within one RFCOMM session between two devices. (This is explained further in Section 2.3.2) To account for the fact that both client and server applications may reside on both sides of an RFCOMM session, with clients on either side making connections independent of each other, the DLCI value space is divided between the two communicating devices using the concept of RFCOMM server channels. This is further described in Section 5.4.

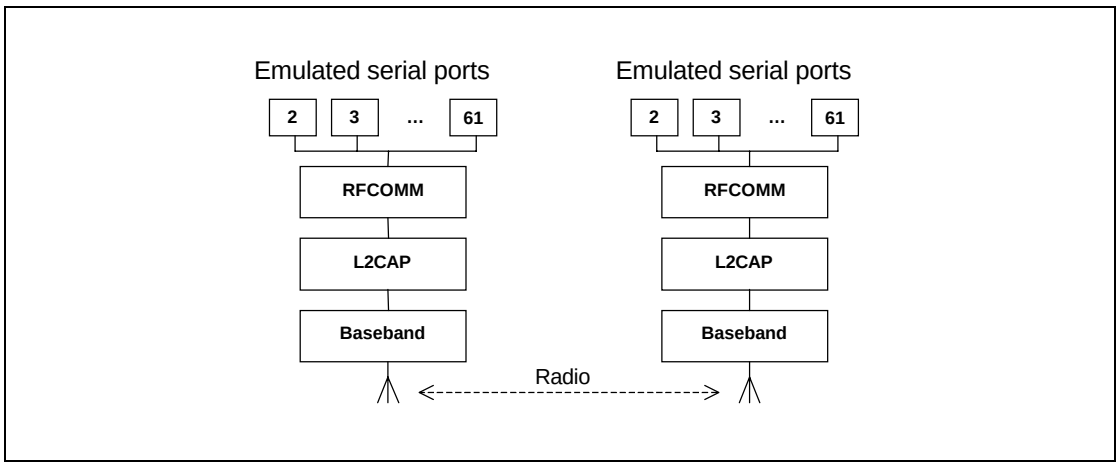


Figure 2.2: Multiple Emulated Serial Ports.

2.3.2 Multiple Emulated Serial Ports and Multiple Bluetooth Devices

If a Bluetooth device supports multiple emulated serial ports and the connections are allowed to have endpoints in different Bluetooth devices, then the RFCOMM entity must be able to run multiple TS 07.10 multiplexer sessions, see Figure 2.3. Note that each multiplexer session is using its own L2CAP channel ID (CID). The ability to run multiple sessions of the TS 07.10 multiplexer is optional for RFCOMM.

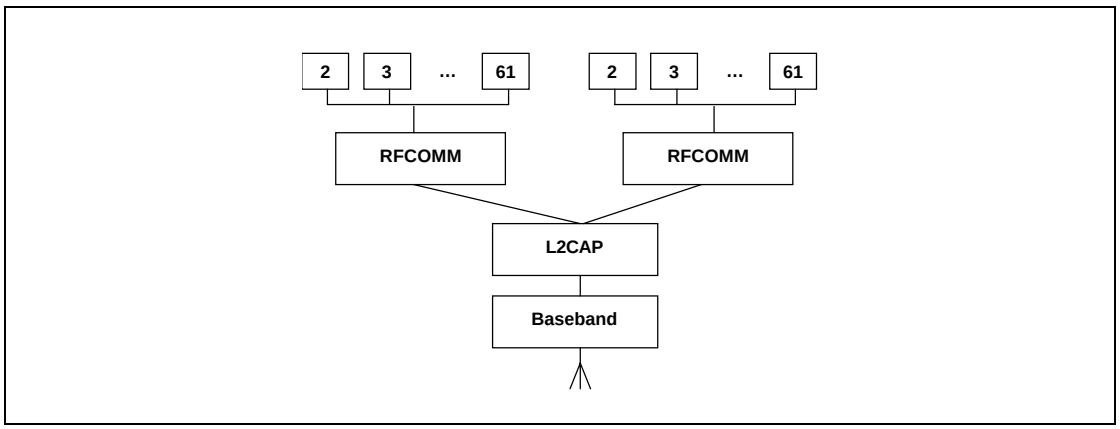


Figure 2.3: Emulating serial ports coming from two Bluetooth devices.

3 SERVICE INTERFACE DESCRIPTION

RFCOMM is intended to define a protocol that can be used to emulate serial ports. In most systems, RFCOMM will be part of a port driver which includes a serial port emulation entity.

3.1 SERVICE DEFINITION MODEL

The figure below shows a model of how RFCOMM fits into a typical system. This figure represents the RFCOMM reference model.

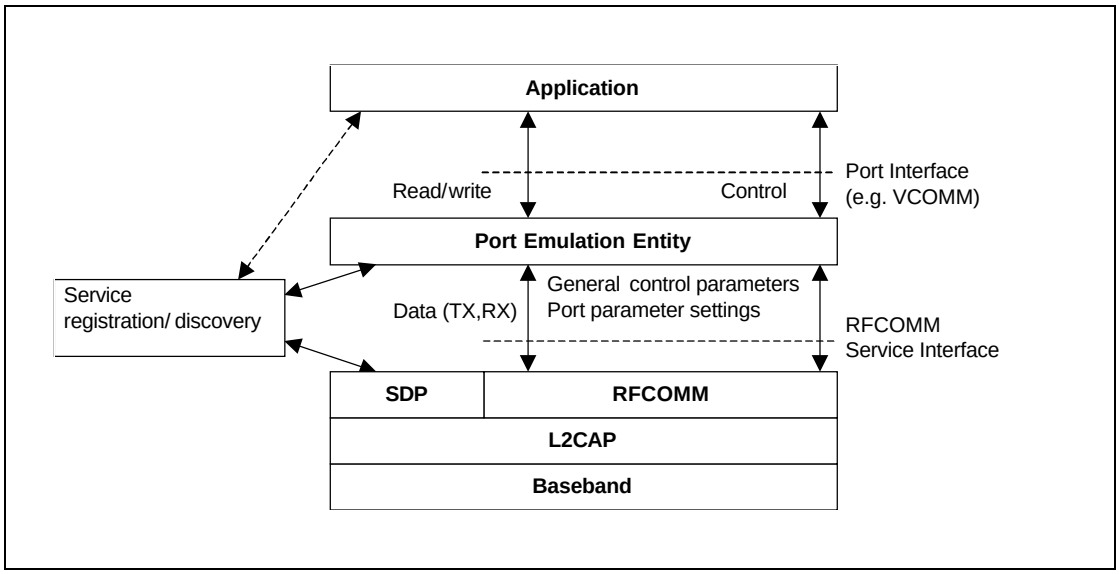


Figure 3.1: RFCOMM reference model

The elements of the RFCOMM reference model are described below.

Element	Description
Application	Applications that utilize a serial port communication interface
Port Emulation Entity	The port emulation entity maps a system-specific communication interface (API) to the RFCOMM services. The port emulation entity plus RFCOMM make up a port driver
RFCOMM	Provides a transparent data stream and control channel over an L2CAP channel. Multiplexes multiple emulated serial ports
Service Registration/ Discovery	Server applications register here on local device, and it provides services for client applications to discover how to reach server applications on other devices
L2CAP	Protocol multiplexing, SAR
Baseband	Baseband protocols defined by Bluetooth Specification

4 TS 07.10 SUBSET SUPPORTED BY RFCOMM

4.1 OPTIONS AND MODES

RFCOMM uses the basic option of TS 07.10.

4.2 FRAME TYPES

Table 4.1 shows the TS 7.10 frame types that are supported in RFCOMM.

Frame Types
Set Asynchronous Balanced Mode (SABM) command
Unnumbered Acknowledgement (UA) response
Disconnected Mode (DM) response
Disconnect (DISC) command
Unnumbered information with header check (UIH) command and response

Table 4.1: Supported frame types in RFCOMM

The 'Unnumbered Information (UI) command and response' are not supported by RFCOMM. Since the error recovery mode option of the TS 07.10 protocol is not used in RFCOMM none of the associated frame types are supported.

4.3 COMMANDS

TS 07.10 defines a multiplexer that has a dedicated control channel, DLCI 0. The control channel is used to convey information between two multiplexers. The following commands in TS 07.10 are supported by RFCOMM:

Supported Control Channel Commands
Test Command (Test)
Flow Control On Command (Fcon)
Flow Control Off Command (Fcoff)
Modem Status Command (MSC)
Remote Port Negotiation Command (RPN)
Remote Line Status (RLS)
DLC parameter negotiation (PN)
Non Supported Command Response (NSC)

Whenever a non-supported command type is received a 'Non-Supported Command Response (NSC)' should be sent.

4.4 CONVERGENCE LAYERS

RFCOMM only supports the type 1 convergence layer in TS 07.10.

The Modem Status Command (MSC) shall be used to convey the RS-232 control signals and the break signal for all emulated serial ports.

5 TS 07.10 ADAPTATIONS FOR RFCOMM

5.1 MEDIA ADAPTATION

The opening flag and the closing flags in the 07.10 basic option frame are not used in RFCOMM, instead it is only the fields contained between the flags that are exchanged between the L2CAP layer and RFCOMM layer, see Figure 5.1.

There is always exactly one RFCOMM frame contained in each L2CAP frame.

Flag	Address	Control	Length Indicator	Information	FCS	Flag
0111 1101	1 octet	1 octet	1 or 2 octets	Unspecified length but integral number of octets	1 octet	0111 1101

Figure 5.1: Frame Structure for Basic option. Note that the opening and closing flags from the 07.10 Basic option are excluded in RFCOMM.

5.1.1 FCS calculation

In 07.10, the frame check sequence (FCS) is calculated on different sets of fields for different frame types. These are the fields that the FCS are calculated on:

For SABM, DISC, UA, DM frames: on Address, Control and length field.

For UIH frames: on Address and Control field.

(This is stated here for clarification, and to set the standard for RFCOMM; the fields included in FCS calculation have actually changed in version 7.0.0 of TS 07.10, but RFCOMM will not change the FCS calculation scheme from the one above.)

5.1.2 P/F-Bit

In the control field (see Figure 5.1 above), there is one bit denoted as the P/F-bit. The general function of this bit is explained in 07.10, section 5.4.4. And the value to use for the P/F-bit in IUH frames is further clarified in 07.10, section 5.4.3.1. These rules apply without modification on an RFCOMM session where the credit based flow control scheme is not in use. See Section 6.5.

However, when credit based flow control is in use, the meaning of the P/F-bit is redefined for UIH frames. This also involves a redefinition of the frame structure, compared to Figure 5.1 above. See Section 6.5.2 for further details.

5.1.3 CR bit

In GSM 07.10, there are two different C/R-bits, one in the frame level (in the address field of the frame header), and one in the message level (in the command type field of the commands sent on the multiplexer control channel). The C/R bit in the frame level is set independently of the C/R bit in the message level.

In the frame level the C/R bit in the frame header is set as follows:

- For SABM, UA, DM and DISC frames C/R bit is set according to Table 1 in GSM 07.10, section 5.2.1.2
- For UIH frames, the C/R bit is always set according to section 5.4.3.1 in GSM 07.10. This applies independently of what is contained wthin the UIH frames, either data or control messages.

In the message level, the C/R bit in the command type field is set as stated in section 5.4.6.2 in GSM 07.10. Control messages are sent in UIH frames, where the C/R bit in the address field of the frame header is always set according to section 5.4.3.1 in GSM 07.10, independently of whether the control message is a comand or a response.

5.2 TS 07.10 MULTIPLEXER START-UP AND CLOSEDOWN PROCEDURE

The start-up and closedown procedures as specified in section 5.7 in TS 07.10 are not supported. This means that the AT-command AT+CMUX is not supported by RFCOMM, neither is the multiplexer close down (CLD) command.

At any time, there must be at most one RFCOMM session between any pair of devices. When establishing a new DLC, the initiating entity must check if there already exists an RFCOMM session with the remote device, and if so, establish the new DLC on that. A session is identified by the Bluetooth BD_ADDR of the two endpoints[1].

5.2.1 Start-up procedure

The device opening up the first emulated serial port connection between two devices is responsible for first establishing the multiplexer control channel. This involves the following steps:

- Establish an L2CAP channel to the peer RFCOMM entity, using L2CAP service primitives, see L2CAP “Service Primitives” on page303 .
- Start the RFCOMM multiplexer by sending SABM command on DLCI 0, and await UA response from peer entity. (Further optional negotiation steps are possible.)

After these steps, DLCs for user data traffic can be established.

Implementation note: There is a special case that may occur if two RFCOMM entities try to establish a session at the same time on an already existing baseband connection. This will be experienced by an RFCOMM entity as receiving a L2CAP connect indication after it has itself issued a L2CAP connect request. In this situation, the RFCOMM entity should respond negatively to the received connect indication (since there may only be one session between two RFCOMM entities). How the situation is resolved is up to the implementation (e.g. it may retry after a random time, or leave it up to the user to retry manually).

5.2.2 Close-down procedure

The device closing the last connection (DLC) on a particular session is responsible for closing the multiplexer by closing the corresponding L2CAP channel.

Closing the multiplexer by first sending a DISC command frame on DLCI 0 is optional, but it is mandatory to respond correctly to a DISC (with UA response).

1. This implies that, when responding to an L2CAP connection indication, the RFCOMM entity should save and associate the new RFCOMM session with the remote BD_ADDR. This is, at least, necessary if subsequent establishment of a DLC in the opposite direction is possible (which may depend on device capabilities).

5.2.3 Link loss handling

If an L2CAP link loss notification is received, the local RFCOMM entity is responsible for sending a connection loss notification to the port emulation/ proxy entity for each active DLC. Then all resources associated with the RFCOMM session should be freed.

The appropriate action to take in the port emulation/proxy entity depends on the API on top. For example, for an emulated serial port (vCOMM), it would be suitable to drop CD, DSR and CTS signals (assuming device is a DTE).

5.3 SYSTEM PARAMETERS

Table 5.1 contains all the applicable system parameters for the RFCOMM implementation of the TS 07.10 multiplexer.

System Parameter	Value
Maximum Frame Size *(N1)*	Default: 127 (negotiable range 23 – 32767)
Acknowledgement Timer *(T1)*	10-60 seconds. Recommended value: 20 seconds. See also note below.
Response Timer for Multiplexer Control Channel *(T2)*	10-60 seconds. Recommended value: 20 seconds. See also note below.

Table 5.1: System parameter values

Note: The timer T1 is the timeout for *frames* sent with the P/F-bit set to one (this applies only to SABM and DISC frames in RFCOMM). T2 is the timeout for *commands* sent in UIH frames on DLCI 0. The exact timeout values are implementation dependent, and may be chosen within the ranges indicated above. However, when sending SABM frame to start a new DLC (with DLCI <> 0), and if there is no knowledge in the local RFCOMM entity that LMP authentication has taken place on the link to the peer entity, T1 must be set in the interval 60 - 300 seconds. (Again, with exact value being implementation dependent).

Since RFCOMM relies on lower layers to provide reliable transmission, the default action performed on timeouts is to close down the multiplexer session.

On the responding side, if authentication procedures are triggered from RFCOMM, this must only be done when receiving a SABM frame, not when receiving configuration commands preparing an unopened DLC.

5.4 DLCI ALLOCATION WITH RFCOMM SERVER CHANNELS

To account for the fact that both client and server applications may reside on both sides of an RFCOMM session, with clients on either side making connections independent of each other, the DLCI value space is divided between the two communicating devices using the concept of RFCOMM server channels and a direction bit.

The RFCOMM server channel number is a subset of the bits in the DLCI part of the address field in the TS 07.10 frame.

Bit No.	1	2	3	4	5	6	7	8
TS 07.10	EA	C/R	DLCI					
RFCOMM	EA	C/R	D	Server Channel				

Table 5.2: The format of the Address Field

Server applications registering with an RFCOMM service interface are assigned a Server Channel number in the range 1…30. [0 and 31 should not be used since the corresponding DLCIs are reserved in TS 07.10] It is this value that should be registered in the Service Discovery Database, see Section 7.2.

For an RFCOMM session, the initiating device is given the direction bit D=1 (and conversely, D=0 in the other device). When establishing a new data link connection on an existing RFCOMM session, the direction bit is used in conjunction with the Server Channel to determine the DLCI to use to connect to a specific application. This DLCI is thereafter used for all packets in both directions between the endpoints.

In effect, this partitions the DLCI value space such that server applications on the non-initiating device are reachable on DLCIs 2,4,6,…,60; and server applications on the initiating device are reachable on DLCIs 3,5,7,…,61. (Note that for a device that supports multiple simultaneous RFCOMM sessions to two or more devices, the direction bit might not be the same on all sessions.)

An RFCOMM entity making a new DLC on an existing session forms the DLCI by combining the Server Channel for the application on the other device, and the inverse of its own direction bit for the session.

DLCI 1 and 62-63 are reserved and never used in RFCOMM.

5.5 MULTIPLEXER CONTROL COMMANDS

Note that in TS 07.10, some Multiplexer Control commands pertaining to specific DLCIs may be exchanged on the control channel (DLCI 0) *before* the corresponding DLC has been established. (This refers the PN and RPN commands.) All such states associated with an individual DLC must be reset to their default values upon receiving a DISC command frame, or when closing the DLC from the local side. This is to ensure that all DLC (re-)establishments on the same session will have predictable results, irrespective of the session history.

If a Multiplexer Control command is received relating to a DLCI that is not open, the responding implementation may replace the "proper" response on the Multiplexer Control channel with a DM frame, sent on the referenced DLCI to indicate that the DLCI is not open, and that the responder would not grant a request to open it later either. (That is, a subsequent SABM sent by initiator would be declined with DM again.)

In GSM TS 07.10, it is stated in section 5.4.6.1 that it is allowed it is allowed to include multiple multiplexer control messages in one frame (as long as the maximum frame size is not exceeded). This feature is disallowed in RFCOMM. (But it is still allowed for an RFCOMM entity to issue multiple multiplexer control messages, each in its own frame, without waiting for responses in between).

5.5.1 Remote Port Negotiation Command (RPN)

The RPN command can be used before a new DLC is opened and should be used whenever the port settings change.

The RPN command is specified as optional in TS 07.10, but it is mandatory to recognize and respond to it in RFCOMM. (Although the handling of individual settings are implementation-dependent.)

5.5.2 Remote Line Status Command (RLS)

This command is used for indication of remote port line status.

The RLS command is specified as optional in TS 07.10, but it is mandatory to recognize and respond to it in RFCOMM. (Although the handling of individual settings are implementation-dependent.)

5.5.3 DLC parameter negotiation (PN)

The PN command is specified as optional in TS 07.10, but it is mandatory to use for RFCOMM implementations conforming to the Bluetooth specification version 1.1 and later. This command *must* be used at least before creation of the first DLC on an RFCOMM session, and the initiator has to try to turn on the use of credit based flow control as described below, and in Section 6.5. TS 07.10 does not explicitly disallow use at any time, but after the DLC is established, the responder of a PN request *may* refuse to change any parameters (by simply including its current parameter set in the response).

There are some parameters in the PN command which convey information not applicable to RFCOMM. These fields must therefore be set to predetermined values by the sender, and they must be ignored by the receiver. This concern the following fields (see table 3 in ref. [1]):

- I1-I4 must be set to 0. (Meaning: use UIH frames.)
- T1-T8 must be set to 0. (Meaning: acknowledgment timer *T1*, which is not negotiable in RFCOMM.)
- NA1-NA8 must be set to 0. (Meaning: number of retransmissions *N2*; always 0 for RFCOMM)

The CL1 -CL4 field is completely redefined. (In TS07.10 this defines the convergence layer to use, which is not applicable to RFCOMM. In RFCOMM, in Bluetooth versions up to 1.0B, this field was forced to 0).

In the PN request sent prior to a DLC establishment, this field *must* contain the value 15 (0xF), indicating support of credit based flow control in the sender. See Table 5.3 below. If the PN response contains any other value than 14 (0xE) in this field, it is inferred that the peer RFCOMM entity is not supporting the credit based flow control feature. (This is only possible if the peer RFCOMM implementation is only conforming to Bluetooth version 1.0B.) If a PN request is sent on an already open DLC, then this field must contain the value zero; it is not possible to “set initial credits” more than once per DLC activation.

A responding implementation must set this field in the PN response to 14 (0xE), if (and only if) the value in the PN request was 15.

Bluetooth version	CL1 - CL4 in PN request	CL1 - CL4 in PN response
<= 1.0B	0x0	0x0
>= 1.1	0xF	0xE*

Table 5.3: CL field values for different RFCOMM versions

*. Or 0x0 if the request was sent from a 1.0B device with no CFC support.

The K1 - K3 field is completely redefined. (In TS07.10 this is the window size for error recovery mode, which is not applicable to RFCOMM. In RFCOMM, in Bluetooth versions up to 1.0B, this field was forced to 0).

In the PN request/response, this field is now interpreted as the initial amount of credits issued to the peer. Thus, this field may take any value in the range from 0 - 7, both in the request and in the response.

This interpretation depends on the contents of the CL1 - CL4 field defined above, i.e. when credit based flow control is not indicated, K1 - K3 must be forced to 0.

If a command is received with invalid (or for some reason unacceptable) values in any field, a DLC parameter negotiation response must be issued with values that are acceptable to the responding device. Or, the responder may send a DM frame on the DLC indicated in the PN command.A device receiving a PN command must send a response. The response may be a PN response, or a DM frame.
For a PN command with N1 value of N1c (c for command), a PN response must have an N1 value N1r (r for response) where N1r <= N1c.
If the receiver is not willing to establish a connection for any reason, it may send a DM frame on the DLCI indicated in the PN command.

A device receiving a PN response may either accept N1r and use this value as the maximum frame data size, or chose not to establish the connection. If it choses not to establish a connection, it must send a DISC or DM frame to indicate this.

If this connection is subsequently established, neither side may send a frame with more than N1r bytes of data,

In the case that no PN frames have been exchanged before the DLC establishment, then both implementations should use the default value described in RFCOMM spec Table 5.2:

6 FLOW CONTROL

Wired ports commonly use flow control such as RTS/CTS to control communications. On the other hand, the flow control between RFCOMM and the lower layer L2CAP depends on the service interface supported by the implementation. In addition RFCOMM has its own flow control mechanisms. This section describes the different flow control mechanisms.

6.1 L2CAP FLOW CONTROL IN OVERVIEW

L2CAP relies on the flow control mechanism provided by the Link Manager layer in the baseband. The flow control mechanism between the L2CAP and RFCOMM layers is implementation-specific.

6.2 WIRED SERIAL PORT FLOW CONTROL

Wired Serial ports falls into two camps – software flow control using characters such as XON/XOFF, and flow control using RTS/CTS or DTR/DSR circuits. These methods may be used by both sides of a wired link, or may be used only in one direction.

6.3 GSM TS 07.10 FLOW CONTROL

The GSM TS 07.10 protocol provides two flow control mechanisms:

1. The GSM TS 07.10 protocol contains flow control commands that operate on the aggregate data flow between two RFCOMM entities; i.e. all DLCIs are affected. The control channel commands, FCon and FCoff, are defined in section 5.4.6.3 in ref [1].
2. The Modem Status command as defined in section 5.4.6.3 in ref [1] is the flow control mechanism that operates on individual DLCI.

Note that these flow control mechanisms only relate to the flow of user payload data in UIH frames on DLCIs other than the multiplexer control channel (DLCI 0). Also note that it is mandatory to support these GSM TS 07.10-styles of flow control, in order to maintain backward compability with Bluetooth version 1.0B.

When MSC commands are used[2], it is only the FC bit that affects the flow on the RFCOMM protocol level. The RTR bit (along with the other V.24 signals in the MSC command) must only be treated transparently as “information” by the RFCOMM entity.

See also figure 3.1. The V.24 signals carry information between the port emulation entities on behalf of applications, and may also be interpreted as “flow con-

2. In any case MSC commands and responses must be exchanged before the data transfer may start, as stated in the ETSI standard TS 07.10, Section 5.4.6.3.7.

trol” information as described in the section on Port Emulation Entity Serial Flow Control 6.4, if negotiation has been done with the RPN command.

6.4 PORT EMULATION ENTITY SERIAL FLOW CONTROL

On Type 1 devices some port drivers (Port Emulation Entities plus RFCOMM) will need to provide flow control services as specified by the API they are emulating. An application may request a particular flow control mechanism like XON/XOFF or RTS/CTS and expect the port driver to handle the flow control. On type 2 devices the port driver may need to perform flow control on the non-RFCOMM part of the communication path; i.e. the physical RS-232 port. This flow control is specified via the control parameters sent by the peer RFCOMM entity (usually a type 1 device). The description of flow control in this section is for port drivers on type 1 devices.

Since RFCOMM already has its own flow control mechanism, the port driver does not need to perform flow control using the methods requested by the application. In the ideal case, the application sets a flow control mechanism and assumes that the COMM system will handle the details. The port driver could then simply ignore the request and rely on RFCOMM's flow control. The application is able to send and receive data, and does not know or care that the port driver did not perform flow control using the mechanism requested. However, in the real world some problems arise.

- The RFCOMM-based port driver is running on top of a packet-based protocol where data may be buffered somewhere in the communication path. Thus, the port driver cannot perform flow control with the same precision as in the wired case.
- The application may decide to apply the flow control mechanism itself in addition to requesting flow control from the port driver.

These problems suggest that the port driver must do some additional work to perform flow control emulation properly. Here are the basic rules for flow control emulation.

- The port driver will not solely rely on the mechanism requested by the application but use a combination of flow control mechanisms.
- The port driver must be aware of the flow control mechanisms requested by the application and behave like the wired case when it sees changes on the non-data circuits (hardware flow control) or flow control characters in the incoming data (software flow control). For example, if XOFF and XON characters would have been stripped in the wired case they must be stripped by the RFCOMM based port driver.
- If the application sets a flow control mechanism via the port driver interface and then proceeds to invoke the mechanism on its own, the port driver must behave in a manner similar to that of the wired case (e.g. If XOFF and XON characters would have been passed through to the wire in the wired case the port driver must also pass these characters).

These basic rules are applied to emulate each of the wired flow control schemes. Note that multiple types of flow control can be set at the same time. Section 5.4.8 in ref [1] defines each flow control mechanism.

6.5 CREDIT BASED FLOW CONTROL

This is a mandatory feature that did not exist in RFCOMM in Bluetooth specifications 1.0B and earlier. Therefore, its use is subject to negotiation before the first DLC establishment (see Section 5.5.3 and Section 6.5.1). Implementations conforming to this specification must support it, and must try to use it when connecting to other devices.

The credit based flow control feature provides flow control on a per - DLC basis. When used, both devices involved in a RFCOMM session will know, for each DLC, how many RFCOMM frames the other device is able to accept before it buffers fill up for that DLC. A sending entity may send as many frames on a DLC as it has credits; if the credit count reaches zero, the sender must stop and wait for further credits from the peer. It is always allowed to send frames containing no user data (length field = 0) when credit based flow control is in use. This mechanism operates independently for each DLC, and for each direction. It does not apply to DLCI 0 or to non-UIH frames.

6.5.1 Initial DLC Negotiation

The use of credit based flow control is a session characteristic. Thus, it has to be negotiated with the PN multiplexor control command (see Section 5.5.3) before the first DLC is established.

After the first successful negotiation and DLC establishment, all DLCs will be flow controlled with this scheme. PN negotiation at subsequent DLC establishments is optional, but recommended, since it also establishes initial credit count values on both sides for both sides.

6.5.2 DLC Operation

When credit based flow control is being used, the meaning of the P/F bit in the control field of the RFCOMM header is redefined for UIH frames.

When the P/F-bit is zero in a UIH-frame, the frame is structured according to Figure 5.1.

When the P/F-bit is one in a UIH-frame, the frame is structure according to Figure 6.1 below. In this case, a credit field is inserted between the length indicator and the payload information field. The value of the credit octet (0 - 255)signifies a number of frames, for which the sender now has buffer space available to receive on the DLC. (Each frame may be sized up to agreed maximum frame size). Credits are additive, meaning that received credits are added to whatever remaining credits that may be left from before. In this case, the length indicator field (as always) indicates the number of information octets in the following information field; however, the maximum number of allowable information octets is decreased by one to compensate for the credit field. (This is to keep the maximum L2CAP payload size constant). This means, that for UIH-frames with the P/F-bit = 0, the maximum size of the information field is the

negotiated one (= the N1 parameter), whereas for UIH-frames with the P/F-bit = 1, the actual maximum size is one less (N1 - 1).

Flag	Address	Control	Length Indicator	Credits	Information	FCS	Flag
0111 1101	1 octet	1 octet	1 or 2 octets	1 octet	Unspecified length but integral number of octets	1 octet	0111 1101

Figure 6.1: Frame Structure for Basic option, UIH frames with P/F-bit = 1 and credit based flow control used. Note that the opening and closing flags from 07.10 Basic option are excluded in RFCOMM.

6.5.3 Other flow control aspects

When credit based flow control is being used on a session, the following applies:

- The FCon and FCoff multiplexer control commands must not be used.
- The FC-bit in the MSC-command has no meaning; it should be set to zero in MSC-commands, and it should be ignored by a receiver.

7 INTERACTION WITH OTHER ENTITIES

7.1 PORT EMULATION AND PORT PROXY ENTITIES

This section defines how the RFCOMM protocol should be used to emulate serial ports. Figure 7.1 shows the two device types that the RFCOMM protocol supports.

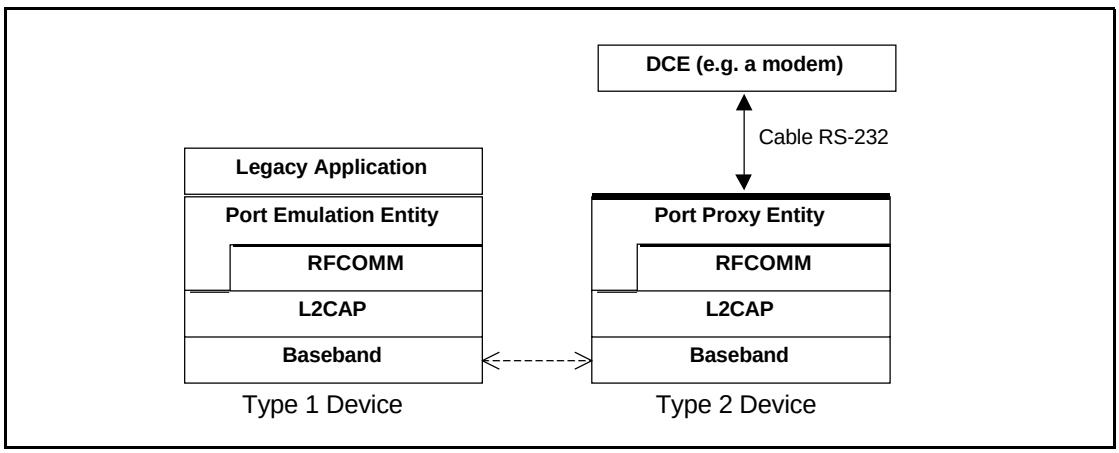


Figure 7.1: The RFCOMM communication model

Type 1 devices are communication endpoints such as computers and printers. Type 2 devices are part of a communication segment; e.g. modems.

7.1.1 Port Emulation Entity

The port emulation entity maps a system specific communication interface (API) to the RFCOMM services.

7.1.2 Port Proxy Entity

The port proxy entity relays data from RFCOMM to an external RS-232 interface linked to a DCE. The communications parameters of the RS-232 interface are set according to received RPN commands, see Section 5.5.1.

7.2 SERVICE REGISTRATION AND DISCOVERY

Registration of individual applications or services, along with the information needed to reach those (i.e. the RFCOMM Server Channel) is the responsibility of each application respectively (or possibly a Bluetooth configuration application acting on behalf of legacy applications not directly aware of Bluetooth).

Below is a template/example for developing service records for a given service or profile using RFCOMM. It illustrates the inclusion of the ServiceClassList with a single service class, a ProtocolDescriptor List with two protocols

(although there may be more protocols on top of RFCOMM). The example shows the use of one other universal attribute (ServiceName). For each service running on top of RFCOMM, appropriate SDP-defined universal attributes and/or service-specific attributes will apply. For additional information on Service Records, see the SDP Specification, Section 2.2 on page 340.

The attributes that a client application needs (at a minimum) to connect to a service on top of RFCOMM are the ServiceClassIDList and the ProtocolDescriptorList (corresponding to the shaded rows in the table below).

Item	Definition	Type/Size	Value	Attribute ID
ServiceClassIDList			Note1	0x0001
ServiceClass0	Note5	UUID/32-bit	Note1	
ProtocolDescriptorList				0x0004
Protocol0	L2CAP	UUID/32-bit	L2CAP /Note1	
Protocol1	RFCOMM	UUID/32-bit	RFCOMM /Note1	
ProtocolSpecificParameter0	Server Channel	Uint8	N = server channel #	
[other protocols]		UUID/32-bit	Note1	
[other protocol-specific parameters]	Note3	Note3	Note3	
ServiceName	Displayable text name	DataElement/ String	'Example service'	Note2
[other universal attributes as appropriate for this service]	Note4	Note4	Note4	Note4
[service-specific attributes]	Note3	Note3	Note3	Note3

Notes:

1. Defined in "Bluetooth Assigned Numbers" (http://www.bluetooth.org/assigned-numbers.htm).
2. For national language support for all 'displayable' text string attributes, an offset has to be added to the LanguageBaseAttributeIDList value for the selected language (see the SDP Specification, Section 5.1.14 on page 373 for details).
3. To be defined (where necessary) for the specific service.
4. For a specific service some of the SDP-defined universal attributes may apply. See the SDP Specification, Section 5.1 on page 366.
5. This indicates the class of service. It can be a single entry or a list of service classes ranging from generic to most specific.

7.3 LOWER LAYER DEPENDENCIES

7.3.1 Reliability

RFCOMM uses the services of L2CAP to establish L2CAP channels to RFCOMM entities on other devices. An L2CAP channel is used for the RFCOMM/TS 07.10 multiplexer session. On such a channel, the TS 07.10 frames listed in Section 4.2 are sent, with the adaptation defined in Section 5.1.

Some frame types (SABM and DISC) as well as UIH frames with multiplexer control commands sent on DLCI 0 always require a response from the remote entity, so they are acknowledged on the RFCOMM level (but not retransmitted in the absence of acknowledgment, see Section 5.3). Data frames do not require any response in the RFCOMM protocol, and are thus unacknowledged.

Therefore, RFCOMM must require L2CAP to provide channels with maximum reliability, to ensure that all frames are delivered in order, and without duplicates. Should an L2CAP channel fail to provide this, RFCOMM expects a link loss notification, which should be handled by RFCOMM as described in Section 5.2.3.

7.3.2 Low power modes

If all L2CAP channels towards a certain device are idle for a certain amount of time, a decision may be made to put that device in a low power mode (i.e. use hold, sniff or park, see 'Baseband Specification' Section 10.10.3 on page 124). This will be done without any interference from RFCOMM. RFCOMM can state its latency requirements to L2CAP.This information may be used by lower layers to decide which low power mode(s) to use.

The RFCOMM protocol as such does not suffer from latency delays incurred by low power modes, and consequentially, this specification does not state any maximum latency requirement on RFCOMM's behalf. Latency sensitivity inherently depends on application requirements, which suggests that an RFCOMM service interface implementation could include a way for applications to state latency requirements, to be aggregated and conveyed to L2CAP by the RFCOMM implementation. (That is if such procedures make sense for a particular platform.)

8 REFERENCES

[1] TS 07.10, ver 6.3.0, ETSI

[2] Bluetooth L2CAP Specification

[3] Bluetooth SDP Specification

[4] Bluetooth Assigned Numbers
(http://www.bluetooth.org/assigned-numbers.htm)

9 TERMS AND ABBREVIATIONS

The following terms are used throughout the document.

DTE	Data Terminal Equipment – in serial communications, DTE refers to a device at the endpoint of the communications path; typically a computer or terminal
DCE	Data Circuit-Terminating Equipment – in serial communications, DCE refers to a device between the communication endpoints whose sole task is to facilitate the communications process; typically a modem
RFCOMM initiator	The device initiating the RFCOMM session; i.e.setting up RFCOMM channel on L2CAP and starting RFCOMM multiplexing with the SABM command frame on DLCI 0 (zero)
RFCOMM Client	An RFCOMM client is an application that requests a connection to another application (RFCOMM server)
RFCOMM Server	An RFCOMM server is an application that awaits a connection from an RFCOMM client on another device. What happens after such a connection is established is not within the scope of this definition
RFCOMM Server Channel	This is a subfield of the TS 07.10 DLCI number. This abstraction is used to allow both server and client applications to reside on both sides of an RFCOMM session

Part F:2

IrDA INTEROPERABILITY

The IrOBEX protocol is utilized by the Bluetooth technology. In Bluetooth, OBEX offers the same features for applications as within the IrDA protocol hierarchy, enabling the applications to work over the Bluetooth protocol stack as well as the IrDA stack.

CONTENTS

1 Introduction 430
1.1 OBEX and Bluetooth Architecture 431
1.2 Bluetooth OBEX-Related Specifications 431
1.3 Other IrOBEX Implementations 432
2 OBEX Object and Protocol 433
2.1 Object 433
2.2 Session Protocol 433
2.2.1 Connect Operation 434
2.2.2 Disconnect Operation 435
2.2.3 Put Operation 435
2.2.4 Get Operation 436
2.2.5 Other Operations 436
3 OBEX over RFCOMM 437
3.1 OBEX Server Start-up on RFCOMM 437
3.2 Receiving OBEX Packets from Serial Port 437
3.3 Connection Establishment 438
3.4 Disconnection 438
3.5 Pushing and Pulling OBEX Packets over RFCOMM 438
4 OBEX over TCP/IP 439
4.1 OBEX Server Start-up on TCP/IP 439
4.2 Connection Establishment 439
4.3 Disconnection 440
4.4 Pushing and Pulling OBEX Packets over TCP 440
5 Bluetooth Application Profiles using OBEX 441
5.1 Synchronization 441
5.2 File Transfer 441
5.3 Object Push 442
6 References 443
7 List of Acronyms and Abbreviations 444

1 INTRODUCTION

The goal of this document is to enable the development of application programs that function well over both short-range RF and IR media. Each media type has its advantages and disadvantages but the goal is for applications to work over both. Rather than fragment the application domain, this document defines the intersection point where Bluetooth and IrDA applications may converge. That intersection point is IrOBEX [1].

IrOBEX is a session protocol defined by IrDA. This protocol is now also utilized by the Bluetooth technology, making it possible for applications to use either the Bluetooth radio technology or the IrDA IR technology. However, even though both IrDA and Bluetooth are designed for short-range wireless communications, they have some fundamental differences relating to the lower-layer protocols. IrOBEX will therefore be mapped over the lower layer protocols which are adopted by Bluetooth.

This document defines how IrOBEX (OBEX for short) is mapped over RFCOMM [2] and TCP/IP [3]. Originally, OBEX (Object Exchange Protocol) was developed to exchange data objects over an infrared link and was placed within the IrDA protocol hierarchy. However, it can appear above other transport layers, now RFCOMM and TCP/IP. At this moment, it is worth mentioning that the OBEX over TCP/IP implementation is an optional feature for Bluetooth devices supporting the OBEX protocol.

The IrOBEX specification [1] provides a model for representing objects and a session protocol, which structures the dialogue between two devices. The IrOBEX protocol follows a client/server **request-response** paradigm for the conversation format.

Bluetooth uses only the connection-oriented OBEX even though IrDA has specified the connectionless OBEX also. The reasons for the connection-oriented approach are:

- OBEX is mapped over the connection-oriented protocols in the Bluetooth architecture.
- Most of application profiles using OBEX and Bluetooth needs a connection-oriented OBEX to provide the functionality described for the features included in these profiles.
- The connectionless OBEX with the connection-oriented one would raise the interoperability problems, which are not desirable.

1.1 OBEX AND BLUETOOTH ARCHITECTURE

Figure 1.1 depicts part of the hierarchy of the Bluetooth architecture and shows the placement of the OBEX protocol and the application profiles using it (See also Section 5 on page 439). The protocols can also communicate with the service discovery DB even though the figure does not show it.

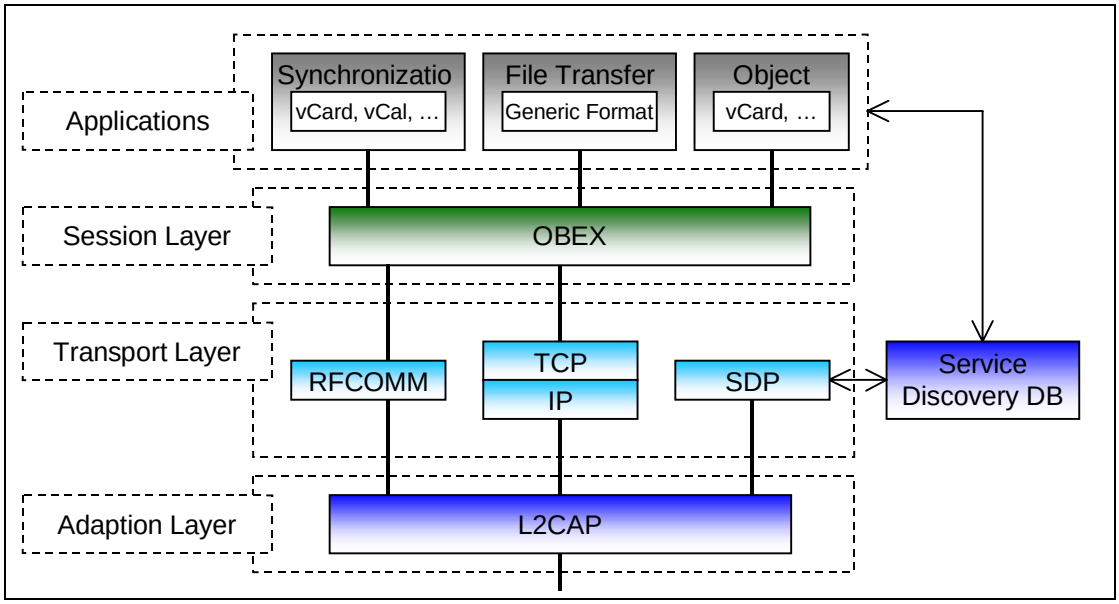


Figure 1.1: Part of Bluetooth Protocol Hierarchy

In the Bluetooth system, the purpose of the OBEX protocol is to enable the exchange of data objects. The typical example could be an object push of business cards to someone else. A more complex example is synchronizing calendars on multiple devices using OBEX. Also, the File Transfer applications can be implemented using OBEX. For the Object Push and Synchronization applications, content formats can be the vCard [4], vCalendar [5], vMessage [6], and vNotes [6] formats. The vCard, vCalendar, vMessage, and vNotes describe the formats for the electronic business card, the electronic calendaring and scheduling, the electronic message and mails, and the electronic notes, respectively.

1.2 BLUETOOTH OBEX-RELATED SPECIFICATIONS

Bluetooth Specification includes five separate specifications related to OBEX and applications using it:

1. Bluetooth IrDA Interoperability Specification (This specification)

- Defines how the applications can function over both Bluetooth and IrDA
- Specifies how OBEX is mapped over RFCOMM and TCP
- Defines the application profiles using OBEX over Bluetooth

2. Bluetooth Generic Object Exchange Profile Specification [7]

- Generic interoperability specification for the application profiles using OBEX
- Defines the interoperability requirements of the lower protocol layers (e.g. Baseband and LMP) for the application profiles

3. Bluetooth Synchronization Profile Specification [8]

- Application Profile for the Synchronization applications
- Defines the interoperability requirements for the applications within the Synchronization application profile
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

4. Bluetooth File Transfer Profile Specification [9]

- Application Profile for the File Transfer applications
- Defines the interoperability requirements for the applications within the File Transfer application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

5. Bluetooth Object Push Profile Specification [10]

- Application Profile for the Object Push applications
- Defines the interoperability requirements for the applications within the Object Push application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

1.3 OTHER IROBEX IMPLEMENTATIONS

Over IR, OBEX has also been implemented over IrCOMM and TinyTP. The Bluetooth technology does not define these protocols as transport protocols for OBEX, but they can be supported by independent software vendors if desired.

2 OBEX OBJECT AND PROTOCOL

This section is dedicated to the model of OBEX objects and the OBEX session protocol. The section is intended to be read with the IrOBEX specification[1].

2.1 OBJECT

The OBEX object model (Section 2 in [1]) describes how OBEX objects are presented. The OBEX protocol can transfer an object by using the **Put**- and **Get**-operations (See Section 2.2.3 and 2.2.4). One object can be exchanged in one or more **Put**-requests or **Get**-responses.

The model handles both information about the object (e.g. type) and object itself. It is composed of headers, which consist of a header ID and value (See Section 2.1 in [1]). The header ID describes what the header contains and how it is formatted, and the header value consists of one or more bytes in the format and meaning specified by Header ID. The specified headers are **Count**, **Name**, **Type**, **Length**, **Time**, **Description**, **Target**, **HTTP**, **Body**, **End of Body**, **Who**, **Connection ID**, **Application Parameters**, **Authenticate Challenge**, **Authenticate Response**, **Object Class**, and User-Defined Headers. These are explained in detail by Section 2.2 in the IrOBEX specification.

2.2 SESSION PROTOCOL

The OBEX operations are formed by **response-request** pairs. Requests are issued by the client and responses by the server. After sending a request, the client waits for a response from the server before issuing a new request. Each request packet consists of a one-byte opcode (See Section 3.3 in [1]), a two-byte length indicator, and required or optional data depending on the operation. Each response packet consists of a one-byte response code (See Section 3.2.1 in [1]), a two-byte length indicator, and required or optional data depending on the operation.

In the following subsections, the OBEX operations are explained in general.

2.2.1 Connect Operation

An OBEX session is started, when an application asks the first time to transmit an OBEX object. An OBEX client starts the establishment of an OBEX connection. The session is started by sending a **Connect**-request (See Section 3.3.1 in [1]). The request format is:

Byte 0	Bytes 1 and 2	Byte 3	Byte 4	Bytes 5 and 6	Byte 7 to n
0x80 (opcode)	Connect request packet length	OBEX version number	Flags	Maximum OBEX packet length	Optional headers

Note. The Big Endian format is used to define the byte ordering for the PDUs (requests and responses) in this specification as well as in the IrOBEX specification; i.e. the most significant byte (MSB) is always on left and the least significant byte (LSB) on right.

At the remote host, the **Connect**-request is received by the OBEX server, if it exists. The server accepts the connection by sending the successful response to the client. Sending any other response (i.e. a non-successful response) back to the client indicates a failure to make a connection. The response format is:

Byte 0	Bytes 1 and 2	Byte 3	Byte 4	Bytes 5 and 6	Byte 7 to n
Response code	Connect response packet length	OBEX version number	Flags	Maximum OBEX packet length	Optional headers

The response codes are list in the Section 3.2.1 in the IrOBEX specification. The bytes 5 and 6 define the maximum OBEX packet length, which can be received by the server. This value may differ from the length, which can be received by the client. These **Connect**-request and response packets must each fit in a single packet.

Once a connection is established it remains ‘alive’, and is only disconnected by requests/responses or by failures (i.e. the connection is not automatically disconnected after each OBEX object has completely transmitted).

2.2.2 Disconnect Operation

The disconnection of an OBEX session occurs when an application, which is needed for an OBEX connection, is closed or the application wants to change the host to which the requests are issued. The client issues the **Disconnect**-request (See Section 3.3.2 in [1]) to the server. The request format is:

Byte 0	Bytes 1 and 2	Byte 3
0x81	Packet length	Optional headers

The request cannot be refused by the server. Thus, it has to send the response, and the response format is:

Byte 0	Bytes 1 and 2	Byte 3
0xA0	Response packet length	Optional response headers

2.2.3 Put Operation

When the connection has been established between the client and server the client is able to push OBEX objects to the server. The **Put**-request is used to push an OBEX object (See Section 3.3.3 in [1]). The request has the following format.

Byte 0	Bytes 1 and 2	Byte 3
0x02 (0x82 when Final bit set)	Packet length	Sequence of headers

A **Put**-request consists of one or more request packets, depending on how large the transferred object is, and how large the packet size is. A response packet from the server is required for every **Put**-request packet. Thus, one response is not permitted for several request packets, although they consist of one OBEX object. The response format is:

Byte 0	Bytes 1 and 2	Byte 3
Response code	Response packet length	Optional response headers

2.2.4 Get Operation

When the connection has been established between the client and server, the client is also able to pull OBEX objects from the server. The **Get**-request is used to pull an OBEX object (See Section 3.3.4 in [1]). The request has the following format.

Byte 0	Bytes 1 and 2	Byte 3
0x03 (0x83 when Final bit set)	Packet length	Sequence of headers starting with Name

The object is returned as a sequence of headers, and the client has to send a request packet for every response packet. The response format is:

Byte 0	Bytes 1 and 2	Byte 3
Response code	Response packet length	Optional response headers

2.2.5 Other Operations

Other OBEX operations consist of a **SetPath**-, and an **Abort**-operation. These are carefully explained in the Sections 3.3.5-6 in the IrOBEX specification. It is important to note that the client can send an **Abort**-request after each response – even in the middle of a request/response sequence. Thus, the whole OBEX object does not have to be received before sending an **Abort**-request. In addition to these operations, the IrOBEX specification facilitates user-defined operations, but their use may not necessarily be adopted in Bluetooth.

3 OBEX OVER RFCOMM

This section specifies how OBEX is mapped over RFCOMM, which is the multiplexing and transport protocol based on ETSI TS 07.10 [11] and which also provides a support for serial cable emulation. The Bluetooth devices supporting the OBEX protocol must satisfy the following requirements.

1. The device supporting OBEX must be able to function as either a client, a server, or both
2. All servers running simultaneously on a device must use separate RFCOMM server channels
3. Applications (service/server) using OBEX must be able to register the proper information into the service discovery database. This information for different application profiles is specified in the profile specifications

3.1 OBEX SERVER START-UP ON RFCOMM

When a client sends a connecting request, a server is assumed to be ready to receive requests. However, before the server is ready to receive (i.e. is running) certain prerequisites must be fulfilled before the server can enter the listening mode:

1. The server must open an RFCOMM server channel
2. The server must register its capabilities into the service discovery database

After this, other hosts are able to find the server if needed, and the server listens for get requests from clients.

3.2 RECEIVING OBEX PACKETS FROM SERIAL PORT

As discussed earlier, one object can be exchanged over one or more **Put**-requests or **Get**-responses (i.e. the object is received in one or several packets). However, if OBEX is running directly over the serial port, it does not receive packets from RFCOMM. Instead, a byte stream is received by OBEX from a serial port emulated by RFCOMM.

To detect packets in the byte stream, OBEX has to look for opcodes or response codes (See Chapter 2.2) depending on whether a packet is a request or a response. The opcodes and response code can be thought of as the start flags of packets. In OBEX packets, there is no ‘end flag’ that would indicate the end of a packet. However, after the opcode or response code, the length of a packet is received in the next two bytes. Thus, the whole length of a packet is known, and the boundary of two packets can be determined.

All data that is not recognized must be dumped. This could cause a synchronization problem but, considering the nature of the OBEX protocol, this is not a problem over RFCOMM, which provides reliable transport over Bluetooth.

3.3 CONNECTION ESTABLISHMENT

A client initiates the establishment of a connection. However, the following sequence of tasks must occur before the client is able to send the first request for data:

1. By using the SD protocol described in the SDP specification [12], the client must discover the proper information (e.g. RFCOMM channel) associated with the server on which the connection can be established
2. The client uses the discovered RFCOMM channel to establish the RFCOMM connection
3. The client sends the **Connect**-request to the server, to establish an OBEX session. The session is established correctly if the client receives a successful response from the server

3.4 DISCONNECTION

The disconnection of an OBEX session over RFCOMM is straightforward. The disconnection is done by using the **Disconnect**-request (See Section 2.2.2). When the client has received the response, the next operation is to close the RFCOMM channel assigned to the OBEX client.

3.5 PUSHING AND PULLING OBEX PACKETS OVER RFCOMM

Data is pushed in OBEX packets over RFCOMM by using **Put**-requests (See Section 2.2.3). After each request, a response is required before the next request with the data can be pushed.

Pulling data from a remote host happens by sending a **Get**-request (See Section 2.2.4. The data arrives in OBEX response packets. After each response, a new request has to be sent, to pull more data.

4 OBEX OVER TCP/IP

This section specifies how OBEX is mapped over the TCP/IP creating reliable connection-oriented services for OBEX. This specification does not define how TCP/IP is mapped over Bluetooth.

The Bluetooth devices, which support the OBEX protocol over TCP/IP, must satisfy the following requirements:

1. The device supporting OBEX must be able to function as either a client, or a server, or both
2. For the server, the TCP port number 650 is assigned by IANA. If an assigned number is not desirable, the port number can be a value above 1023. However, the use of the TCP port number (650) defined by IANA is highly recommended. The 0-1023 range is reserved by IANA (See [13])
3. The client must use a port number (on the client side), which is not within the 0-1023 range
4. Applications (service/server) using OBEX must be able to register the proper information into the service discovery database. This information for different application profiles is specified in the profile specifications

4.1 OBEX SERVER START-UP ON TCP/IP

When a client sends a **Put**- or **Get**-request, a server is assumed to be ready to receive requests. However, when the server is ready (i.e. is running), certain prerequisites must be fulfilled before the server can enter the listening mode:

1. The server must initialize a TCP port with the value 650 or value above 1023
2. The server registers its capabilities into the service discovery database

After this, other devices are able to find the server if needed, and the server listens for get requests from clients.

4.2 CONNECTION ESTABLISHMENT

A client initiates a connection. However, the following sequence of tasks must occur before a connection can be established:

1. By using, the SD protocol described in the SDP specification [12], the client discovers the proper information (e.g. TCP port number) associated with the server, to enable the connection can be established
2. The client initializes a socket associated to a TCP port number above 1023, and establishes a TCP connection with the host of the server
3. The client sends the **Connect**-request to the server, to establish an OBEX session. The session is established correctly if the client receives a successful response from the server.

4.3 DISCONNECTION

The disconnection of an OBEX session over TCP is straightforward. The disconnection is done by using the **Disconnect**-request (See Section 2.2.2). When the client has received the response, the next operation is to close the TCP port dedicated for this session.

4.4 PUSHING AND PULLING OBEX PACKETS OVER TCP

See Section 3.5.

5 BLUETOOTH APPLICATION PROFILES USING OBEX

Bluetooth SIG (Special Interest Group) has defined three separate application profiles using OBEX. These profiles are briefly introduced in this section.

5.1 SYNCHRONIZATION

Basically, the synchronization means comparing two object stores, determining their inequalities, and then unifying these two object stores. The Bluetooth devices supporting the synchronization may be desktop PCs, notebooks, PDAs, cellular phones, or smart phones.

The Bluetooth Synchronization profile uses the servers and clients compliant to the IrMC synchronization specified by IrDA (See Section 5 in [6]).
The Bluetooth Synchronization servers and clients must support the level 4 synchronization functionality specified in the IrMC specification.

The actual logic of the synchronization engines which process the synchronization algorithm at the client device is implementation-specific. It is therefore left to the participating software vendors, and is not considered in the Bluetooth specifications.

The synchronization is not limited to one type of application. The Bluetooth synchronization (i.e. the IrMC synchronization) enables four different application classes:

1. Phone Book – provides a means for a user to manage contact records
2. Calendar – enables a user to manage calendar items, and can also be used for ‘to-do’ or task lists
3. Messaging – lets a user manage messages (e.g. e-mails)
4. Notes – provides a means for a user to manage small notes

The interoperability requirements for the Bluetooth Synchronization profile are defined in the Synchronization Profile [8] and Generic Object Exchange Profile [7] specifications.

5.2 FILE TRANSFER

At the minimum, the File Transfer profile is intended for sending and retrieving generic files to and from the Bluetooth device. The File Transfer service also facilitates the browsing of the remote Bluetooth device’s folder.

The interoperability requirements for the Bluetooth File Transfer profile are defined in the File Transfer Profile [9] and Generic Object Exchange Profile [7] specifications.

5.3 OBJECT PUSH

The Object Push profile is the special case of the File Transfer Profile for beaming objects and optionally pulling the default objects. At a minimum, it offers the capability to exchange business cards, but is not limited to this service.

The interoperability requirements for the Object Push profile are defined in the Object Push Profile [10] and Generic Object Exchange Profile [7] specifications.

6 REFERENCES

[1] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX), Version 1.2, April 1999

[2] Bluetooth RFCOMM with TS 07.10, on page 393

[3] Internet Engineering Task Force, IETF Directory List of RFCs (http://www.ietf.org/rfc/), May 1999.

[4] The Internet Mail Consortium, vCard - The Electronic Business Card Exchange Format, Version 2.1, September 1996.

[5] The Internet Mail Consortium, vCalendar - The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996.

[6] Infrared Data Association, IrMC (Ir Mobile Communications) Specification, Version 1.1, February 1999.

[7] Bluetooth Generic Object Exchange Profile, see Volume 2.

[8] Bluetooth Synchronization Profile, see Volume 2.

[9] Bluetooth File Transfer Profile, see Volume 2.

[10] Bluetooth Object Push Profile, see Volume 2.

[11] ETSI, TS 07.10, Version 6.3.0

[12] Bluetooth Service Discovery Protocol, see Volume 2.

[13] Internet Assigned Numbers Authority, IANA Protocol/Number Assignments Directory (http://www.iana.org/numbers.html), May 1999.

7 LIST OF ACRONYMS AND ABBREVIATIONS

Abbreviation or Acronym	Meaning
GEOP	Generic Object Exchange Profile
IrDA	Infrared Data Association
IrMC	Ir Mobile Communications
L2CAP	Logical Link Control and Adaptation Protocol
LSB	Least Significant Byte
MSB	Most Significant Byte
OBEX	Object exchange protocol
PDU	Protocol Data Unit
RFCOMM	Serial cable emulation protocol based on ETSI TS 07.10
SD	Service Discovery
SDP	Service Discovery Protocol
SDDB	Service Discovery Database
TCP/IP	Transport Control Protocol/Internet Protocol

Part F:3

TELEPHONY CONTROL PROTOCOL SPECIFICATION

TCS Binary

This document describes the Bluetooth Telephony Control protocol Specification – Binary (TCS *Binary*), using a bit-oriented protocol. This protocol defines the call control signalling for the establishment of speech and data calls between Bluetooth devices.
In addition, it defines mobility management procedures for handling Bluetooth TCS devices.

CONTENTS

1 General Description 451
1.1 Overview 451
1.2 Operation between devices 451
1.3 Operation between layers 453
2 Call Control (CC) 455
2.1 Call States 455
2.2 Call Establishment 455
2.2.1 Call Request 455
2.2.2 Bearer selection 456
2.2.3 Overlap Sending 457
2.2.4 Call Proceeding 457
2.2.4.1 Call proceeding, enbloc sending 457
2.2.4.2 Call proceeding, overlap sending 458
2.2.4.3 Expiry of timer T310 458
2.2.5 Call Confirmation 458
2.2.6 Call Connection 458
2.2.7 Call Information 459
2.2.8 Non-selected user clearing 459
2.2.9 In-band tones and announcements 459
2.2.10 Failure of call establishment 460
2.2.11 Call Establishment Message Flow 461
2.3 Call Clearing 462
2.3.1 Normal Call Clearing 462
2.3.2 Abnormal Call Clearing 463
2.3.3 Clear Collision 463
2.3.4 Call Clearing Message Flow 464

3 Group Management (GM) 465
3.1 Overview 465
3.2 The Wireless User Group 465
3.2.1 Description 465
3.2.2 Encryption within the WUG 466
3.2.3 Unconscious pairing 466
3.3 Obtain Access Rights 467
3.3.1 Procedure description 467
3.3.2 Message flow 467
3.4 Configuration Distribution 468
3.4.1 Procedure Description 468
3.4.2 Message flow 468
3.5 Fast inter-member Access 469
3.5.1 Listen Request 469
3.5.2 Listen Accept 469
3.5.3 Listen Reject by the WUG Master 470
3.5.4 Listen Reject by the WUG Member 470
3.5.5 Message flow 470
4 Connectionless TCS (CL) 472
5 Supplementary Services (SS) 473
5.1 Calling Line Identity 473
5.2 DTMF start & stop 473
5.2.1 Start DTMF request 474
5.2.2 Start DTMF response 474
5.2.3 Stop DTMF request 474
5.2.4 Stop DTMF response 474
5.2.5 Message flow 474
5.3 Register recall 475

6 Message formats 476
6.1 Call Control message formats 477
6.1.1 ALERTING 477
6.1.2 CALL PROCEEDING 477
6.1.3 CONNECT 478
6.1.4 CONNECT ACKNOWLEDGE 478
6.1.5 DISCONNECT 479
6.1.6 INFORMATION 479
6.1.7 PROGRESS 480
6.1.8 RELEASE 480
6.1.9 RELEASE COMPLETE 481
6.1.10 SETUP 481
6.1.11 SETUP ACKNOWLEDGE 482
6.1.12 Start DTMF 482
6.1.13 Start DTMF Acknowledge 483
6.1.14 Start DTMF Reject 483
6.1.15 Stop DTMF 483
6.1.16 Stop DTMF Acknowledge 484
6.2 Group Management message formats 484
6.2.1 ACCESS RIGHTS REQUEST 484
6.2.2 ACCESS RIGHTS ACCEPT 484
6.2.3 ACCESS RIGHTS REJECT 485
6.2.4 INFO SUGGEST 485
6.2.5 INFO ACCEPT 485
6.2.6 LISTEN REQUEST 486
6.2.7 LISTEN SUGGEST 486
6.2.8 LISTEN ACCEPT 486
6.2.9 LISTEN REJECT 487
6.3 TCS Connectionless message formats 487
6.3.1 CL INFO 487
7 Message coding 488
7.1 Overview 488
7.2 Protocol discriminator 489
7.3 Message type 489
7.4 Other information elements 491
7.4.1 Coding rules 491
7.4.2 Audio control 493
7.4.3 Bearer capability 493
7.4.4 Call class 496

7.4.5 Called party number 497
7.4.6 Calling party number 498
7.4.7 Cause 499
7.4.8 Clock offset 499
7.4.9 Company specific 500
7.4.10 Configuration data 501
7.4.11 Destination CID 502
7.4.12 Keypad facility 502
7.4.13 Progress indicator 502
7.4.14 SCO Handle 503
7.4.15 Sending complete 503
7.4.16 Signal 503

8 Message Error handling 504
8.1 Protocol discrimination error 504
8.2 Message too short or unrecognized 504
8.3 Message type or message sequence errors 504
8.4 Information element errors 504

9 Protocol Parameters 506
9.1 Protocol Timers 506

10 Bluetooth Standard Character Set 507

11 References 508

12 List of Figures 509

13 List of Tables 510

Appendix 1 - TCS Call States 511

1 GENERAL DESCRIPTION

1.1 OVERVIEW

The Bluetooth Telephony Control protocol Specification Binary (TCS *Binary*) is based on the ITU-T Recommendation Q.931[1], applying the symmetrical provisions as stated in Annex D of Q.931. The resulting text does not discriminate between user and network side, but merely between Outgoing Side (the party originating the call) and Incoming Side (the party terminating the call). Effort was made to only apply those changes necessary for Bluetooth and foreseen applications, enabling re-use of Q.931 to the largest extent possible.

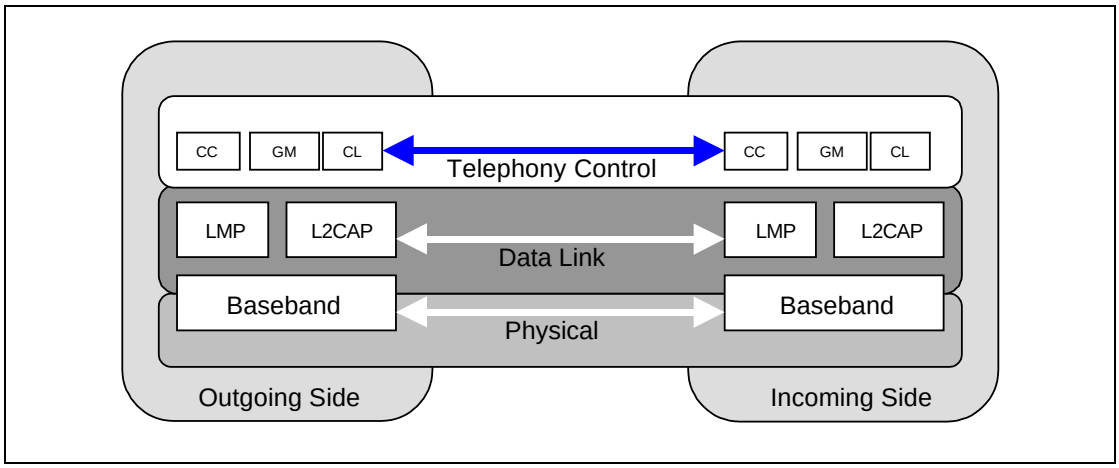


Figure 1.1: TCS within the Bluetooth stack

The TCS contains the following functionality:

- Call Control (CC) – signalling for the establishment and release of speech and data calls between Bluetooth devices
- Group Management – signalling to ease the handling of groups of Bluetooth devices
- ConnectionLess TCS (CL) – provisions to exchange signalling information not related to an ongoing call

1.2 OPERATION BETWEEN DEVICES

TCS uses point-to-point signalling and may use point-to-multipoint signalling. Point-to-point signalling is used when it is known to which side (Bluetooth device) a call needs to be established (*single-point configuration*).

Point-to-multipoint signalling may be used when there are more sides available for call establishment (*multi-point configuration*); e.g. when, for an incoming call, a home base station needs to alert all phones in range.

Point-to-point signalling is mapped towards a connection-oriented L2CAP channel, whereas point-to-multipoint signalling is mapped towards the connectionless L2CAP channel, which in turn is sent as broadcast information on the beacon channel (piconet broadcast).

Figure 1.2 illustrates point-to-point signalling to establish a voice or data call in a single-point configuration. First the other device is notified of the call request using the point-to-point signalling channel (A). Next, this signalling channel is used to further establish the speech or data channel (B).

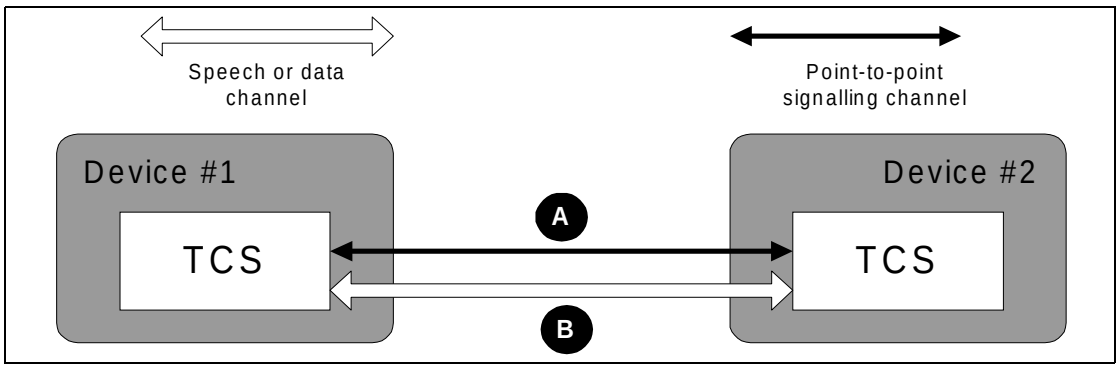


Figure 1.2: Point-to-point signalling in a single-point configuration

Figure 1.3 below illustrates how point-to-multipoint signalling and point-to-point signalling is used to establish a voice or data call in a multi-point configuration. First all devices are notified of the call request using point-to-multipoint signalling channel (A). Next, one of the devices answers the call on the point-to-point signalling channel (B); this signalling channel is used to further establish the speech or data channel (C).

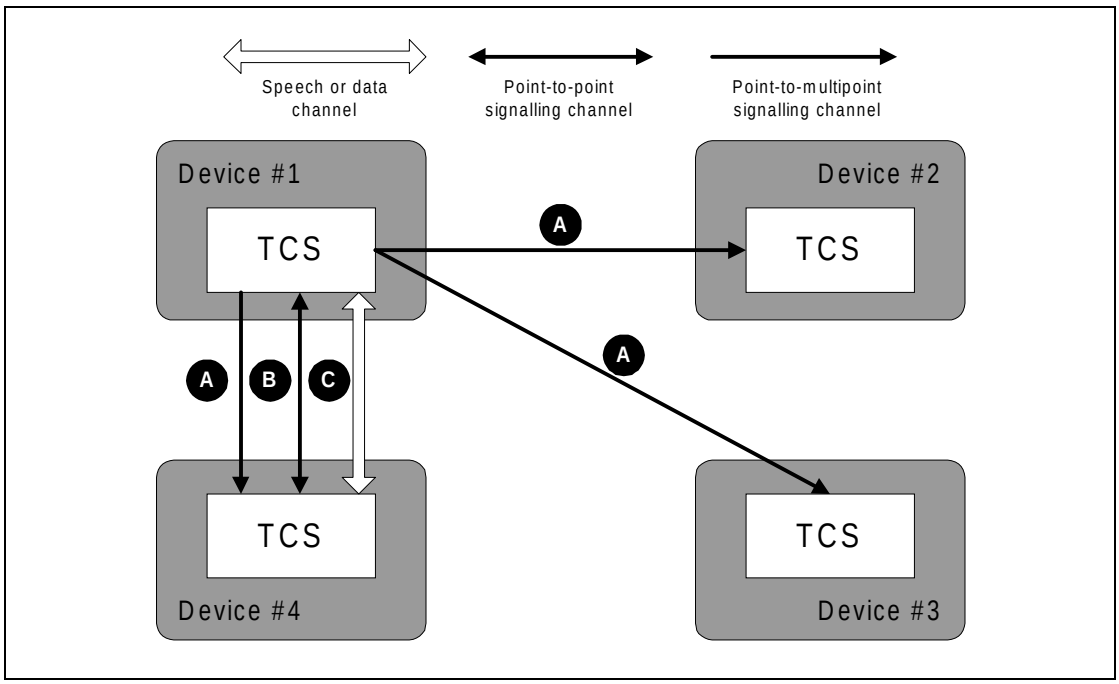


Figure 1.3: Signalling in a multi-point configuration

1.3 OPERATION BETWEEN LAYERS

TCS implementations should follow the general architecture described below (note that, for simplicity, handling of data calls is not drawn).

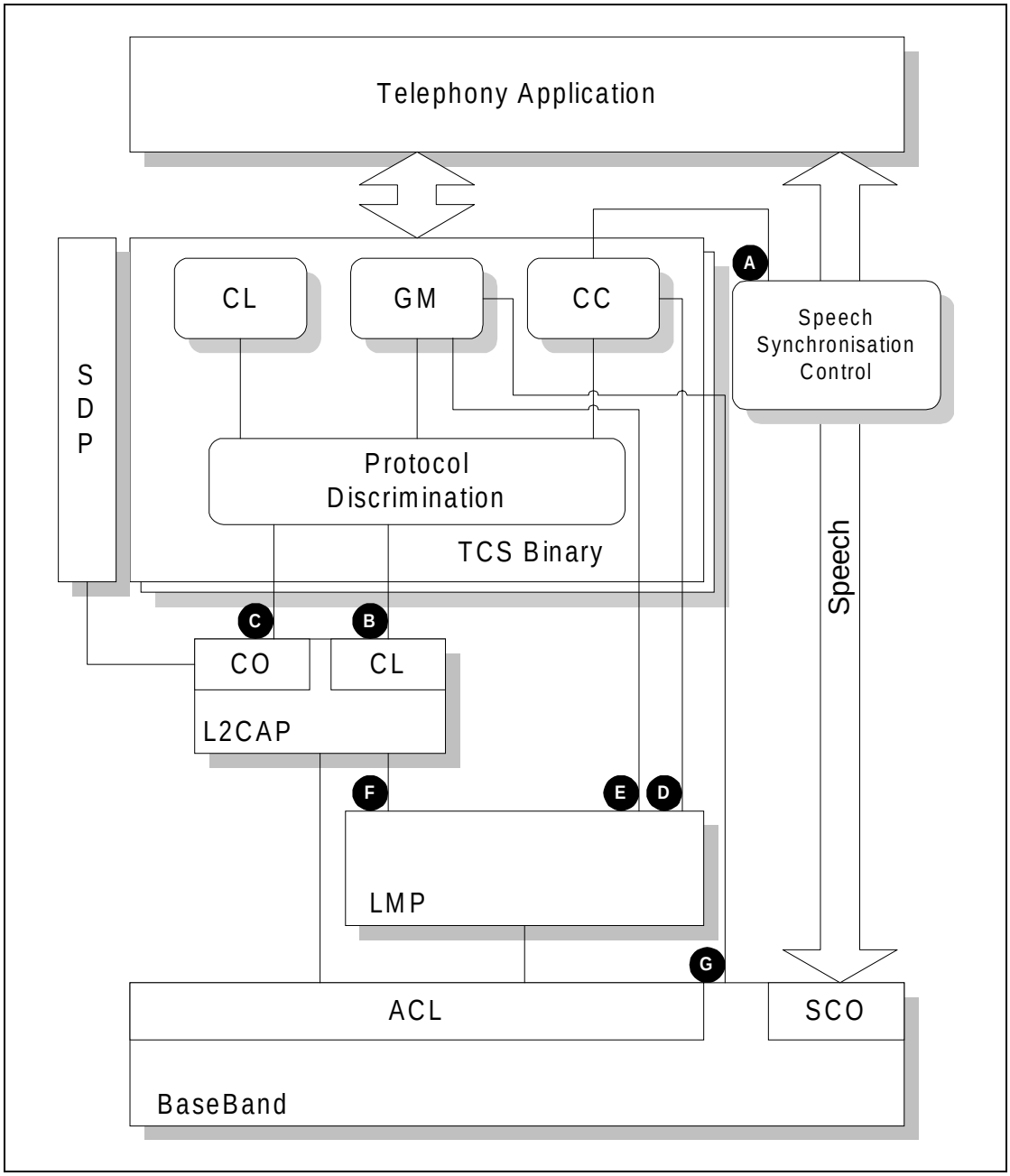


Figure 1.4: TCS Architecture

The internal structure of TCS Binary contains the functional entities Call Control, Group Management and ConnectionLess as described in Section 1.1 on page 449, complemented with the Protocol Discrimination which, based upon the TCS internal protocol discriminator, routes traffic to the appropriate functional entity.

To handle more calls simultaneously, multiple instances of TCS Binary may exist at the same time. Discrimination between the multiple instances can be based on the L2CAP channel identifier.

TCS Binary interfaces with a number of other (Bluetooth) entities to provide its (telephone) services to the application. The interfaces are identified in Figure 1.4 above, and information is exchanged across these interfaces for the following purposes:

A The Call Control entity provides information to the speech synchronization control about when to connect (disconnect) the speech paths. This information is based upon the call control messages (e.g. reception of CONNECT ACKNOWLEDGE or DISCONNECT, see Section 2 on page 453)

B To send a SETUP message (see Section 2.2.1 on page 453) using point-to-multipoint signalling, it is delivered on this interface to L2CAP for transmission on the connectionless channel. The other way round – L2CAP uses this interface to inform TCS of a SETUP message received on the connectionless channel. The connectionless L2CAP channel maps onto the piconet broadcast

C Whenever a TCS message needs to be sent using point-to-point signalling, it is delivered on this interface to L2CAP for transmission on a connection-oriented channel. During L2CAP channel establishment specific quality of service to be used for the connection will be indicated, in particular the usage of low power modes (L2CAP will inform LMP about this – interface F)

D The Call Control entity controls the LMP directly, for the purpose of establishing and releasing SCO links

E & G. The Group Management entity controls the LMP and LC/Baseband directly during initialization procedures to control (for example) the inquiry, paging and pairing.

2 CALL CONTROL (CC)

2.1 CALL STATES

The call states used by the TCS are those identified in Q.931[1], for the user side only. To allow for implementation within computing power- and memory-restricted devices, only a subset of the states is mandatory for TCS based implementations. This mandatory subset is termed **Lean TCS**.

The states are named as follows. States in bold are mandatory states, part of Lean TCS:

General States
- ***Null (0)***
- ***Active (10)***
- ***Disconnect request (11)***
- ***Disconnect indication (12)***
- ***Release request (19)***

Outgoing Side States
- ***Call initiated (1)***
- *Overlap sending (2)*
- *Outgoing call proceeding (3)*
- *Call delivered (4)*

Incoming Side States
- ***Call present (6)***
- *Call received (7)*
- ***Connect request (8)***
- *Incoming call proceeding (9)*
- *Overlap receiving (25)*

These states, together with the state transitions, have been indicated in the state diagram contained in Appendix 1 – TCS Call States. For clarity, a separate state diagram has been included for Lean TCS.

2.2 CALL ESTABLISHMENT

A connection-oriented L2CAP channel between the Outgoing and Incoming Side shall be available before any of the CC procedures can operate.

Additionally, in a multi-point configuration (see Section 1.2 on page 449), a connectionless L2CAP channel shall be available between the Outgoing and Incoming Side.

2.2.1 Call Request

The Outgoing Side initiates call establishment by sending a SETUP message, and starting timer T303.

In case of a single-point configuration (see Section 1.2 on page 449), the SETUP message is delivered on the connection-oriented channel.

In case of a multi-point configuration (see Section 1.2 on page 449), the SETUP message may be delivered on the connection-less channel. This causes the SETUP message to be transmitted as a broadcast message at every beacon instant (as described in Baseband Specification Section 10.8.4 on page 112).

If no response (as prescribed in Section 2.2.4 on page 455) is received from the Incoming Side before timer T303 expires, the Outgoing Side shall:

1. If the SETUP message was delivered on a connection-less channel, return to the Null state. This stops the transmission of the SETUP message.
2. If the SETUP message was delivered on a connection-oriented channel, send a RELEASE COMPLETE message to the Incoming Side. This message should contain cause # 102, *recovery on timer expiry*.

The SETUP message shall always contain the call class. It shall also contain all the information required by the Incoming Side to process the call. The number digits within the Called party number information element may optionally be incomplete, thus requiring the use of overlap sending (Section 2.2.3 on page 455). The SETUP message may optionally contain the Sending complete information element in order to indicate that the number is complete.

Following the transmission of the SETUP message, the Outgoing Side shall enter the Call initiated state. On receipt of the SETUP message the Incoming Side shall enter the Call present state.

2.2.2 Bearer selection

The SETUP message sent during the Call Request may contain the Bearer capability information element, to indicate the requested bearer. The Incoming Side may negotiate on the requested bearer by including a Bearer capability information element in the first message in response to the SETUP message.

The Bearer capability information element indicates which lower layer resources (the *bearer channel*) are used during a call. If bearer capability 'Synchronous Connection-Oriented (SCO) ' is indicated, an SCO link will be used, with the indicated packet type and voice coding to enable speech calls. If bearer capability 'Asynchronous Connection-Less (ACL)' is indicated, an ACL link will be used. On top of this, there will be an L2CAP channel with indicated QoS requirements, to enable data calls. If bearer capability 'None' is indicated, no separate bearer channel will be established.

Note: it is the responsibility of the implementation to assure that the bearer capability as indicated is available to the call.

2.2.3 Overlap Sending

If the received SETUP message does not contain a Sending complete indication information element, and contains either –

a) incomplete called-number information, or

b) called-number information which the Incoming Side cannot determine to be complete,

then the Incoming Side shall start timer T302, send a SETUP ACKNOWLEDGE message to the Outgoing Side, and enter the Overlap receiving state.

When the SETUP ACKNOWLEDGE message is received, the Outgoing Side shall enter the Overlap sending state, stop timer T303, and start timer T304.

After receiving the SETUP ACKNOWLEDGE message, the Outgoing Side shall send the remainder of the call information (if any) in the called party number information element of one or more INFORMATION messages.

The Outgoing Side shall restart timer T304 when each INFORMATION message is sent.

The INFORMATION message, which completes the information sending, may contain a sending complete information element. The Incoming Side shall restart timer T302 on receipt of every INFORMATION message not containing a sending complete indication, if it cannot determine that the called party number is complete.

At the expiry of timer T304, the Outgoing Side shall initiate call clearing in accordance with Section 2.3.1 with cause #102, *recovery on timer expiry*.

At the expiry of timer T302, the Incoming Side shall:

- if it determines that the call information is incomplete, initiate call clearing in accordance with Section 2.3.1 with cause #28, *invalid number format*.
- otherwise the Incoming Side shall reply with a CALL PROCEEDING, ALERTING or CONNECT message.

2.2.4 Call Proceeding

2.2.4.1 Call proceeding, enbloc sending

If enbloc sending is used (i.e. the Incoming Side can determine it has received sufficient information in the SETUP message from the Outgoing Side to establish the call) the Incoming Side shall send a CALL PROCEEDING message to the Outgoing Side to acknowledge the SETUP message and to indicate that the call is being processed. Upon receipt of the CALL PROCEEDING message, the Outgoing Side shall enter the Outgoing Call proceeding state stop

timer T303 and start timer T310. After sending the CALL PROCEEDING message, the Incoming Side shall enter the Incoming Call proceeding state.

2.2.4.2 Call proceeding, overlap sending

Following the occurrence of one of these conditions –

- the receipt by the Incoming Side of a Sending complete indication, or
- analysis by the Incoming Side that all call information necessary to effect call establishment has been received,

the Incoming Side shall send a CALL PROCEEDING message to the Outgoing Side, stop timer T302, and enter the Incoming Call proceeding state.

When the Outgoing Side receives of the CALL PROCEEDING message it shall enter the Outgoing Call proceeding state, stop timer T304 and, if applicable, start timer T310.

2.2.4.3 Expiry of timer T310

On expiry of T310 (i.e. if the Outgoing Side does not receive an ALERTING, CONNECT, DISCONNECT or PROGRESS message), the Outgoing Side shall initiate call clearing in accordance with Section 2.3.1 on page 460 with cause #102, *recovery on timer expiry*.

2.2.5 Call Confirmation

Upon receiving an indication that user alerting has been initiated at the called address, the Incoming Side shall send an ALERTING message, and shall enter the Call received state.

When the Outgoing Side receives the ALERTING message, the Outgoing Side may begin an internally generated alerting indication and shall enter the Call delivered state. The Outgoing Side shall stop timer T304 (in case of overlap receiving), stop timer T303 or T310 (if running), and start timer T301 (unless another internal altering supervision timer function exists).

On expiry of T301, the Outgoing Side shall initiate call clearing in accordance with Section 2.3.1 on page 460 with cause #102, *recovery on timer expiry*.

2.2.6 Call Connection

An Incoming Side indicates acceptance of an incoming call by sending a CONNECT message to the Outgoing Side, and stopping the user alerting. Upon sending the CONNECT message the Incoming Side shall start timer T313.

On receipt of the CONNECT message, the Outgoing Side shall stop any internally generated alerting indications, shall stop (if running) timers T301, T303, T304, and T310, shall complete the requested bearer channel to the Incoming Side, shall send a CONNECT ACKNOWLEDGE message, and shall enter the Active state.

The CONNECT ACKNOWLEDGE message indicates completion of the requested bearer channel. Upon receipt of the CONNECT ACKNOWLEDGE message, the Incoming Side shall connect to the bearer channel, stop timer T313 and enter the Active state.

When timer T313 expires prior to the receipt of a CONNECT ACKNOWLEDGE message, the Incoming Side shall initiate call clearing in accordance with Section 2.3.1 on page 460 with cause #102, *recovery on timer expiry*.

2.2.7 Call Information

While in the Active state, both sides may exchange any information related to the ongoing call using INFORMATION messages.

2.2.8 Non-selected user clearing

When the call has been delivered on a connection-less channel (in case of a multi-point configuration), in addition to sending a CONNECT ACKNOWLEDGE message to the Incoming Side selected for the call, the Outgoing Side shall send a RELEASE message (indicating cause #26, *non-selected user clearing*) to all other Incoming Sides that have sent SETUP ACKNOWLEDGE, CALL PROCEEDING, ALERTING, or CONNECT messages in response to the SETUP message. These RELEASE messages are used to notify the Incoming Sides that the call is no longer offered to them.

2.2.9 In-band tones and announcements

When the Incoming Side provides in-band tones/announcements, and if the requested bearer implies speech call, the Incoming Side will first complete the bearer channel (if not already available). Then a progress indicator #8, *in-band information or appropriate pattern is now available* is sent simultaneously with the application of the in-band tone/announcement. This progress indicator may be included in any call control message that is allowed to contain the progress indicator information element or, if there is no call state change, in a dedicated PROGRESS message.

Upon receipt of this message, the Outgoing Side may connect (if not already connected) to the bearer channel to receive the in-band tone/announcement.

2.2.10 Failure of call establishment

In the Call present, Overlap receiving, Incoming call proceeding, or Call received states, the Incoming Side may initiate clearing as described in Section 2.3 on page 460 with a cause value indicated. Examples of some the cause values that may be used to clear the call, when the Incoming Side is in the Call present, Overlap receiving, or Incoming call proceeding state are the following:

#1 unassigned (unallocated) number
#3 no route to destination
#17 user busy
#18 no user responding
#22 number changed
#28 invalid number format (incomplete number)
#34 no circuit/channel available
#44 requested circuit/channel not available
#58 bearer capability not presently available
#65 bearer capability not implemented

Examples of two of the cause values that may be used to clear the call when the Incoming Side is in the Call received state are as follows:

#19 no answer from user (user alerted)
#21 call rejected by user

2.2.11 Call Establishment Message Flow

The figure below provides a complete view of the messages exchanged during successful Call Establishment, as described in the sections above. The mandatory messages, part of the Lean TCS, are indicated by a solid arrow. A dotted arrow indicates the optional messages. A triangle indicates a running timer.

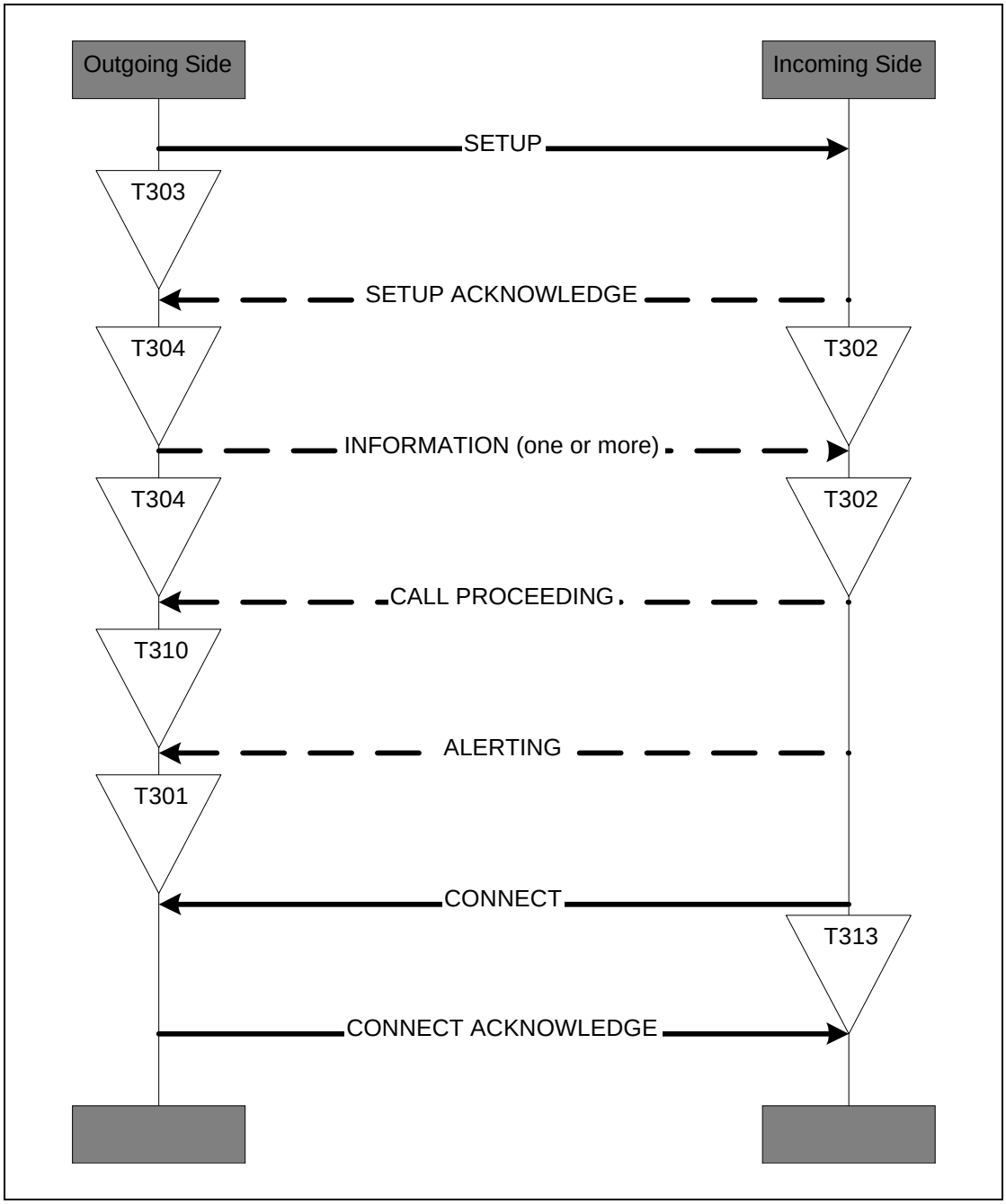


Figure 2.1: Call establishment message flow

2.3 CALL CLEARING

2.3.1 Normal Call Clearing

Apart from the exceptions identified in Section 2.3.2 on page 461, the clearing procedures are symmetrical and may be initiated by either the Outgoing or the Incoming Side. In the interest of clarity, the following procedures describe only the case where the Outgoing Side initiates clearing.

On sending or receiving any call clearing message, any protocol timer other than T305 and T308 shall be stopped.

The Outgoing Side shall initiate clearing by sending a DISCONNECT message, starting timer T305, disconnecting from the bearer channel, and entering the Disconnect request state.

The Incoming Side shall enter the Disconnect indication state upon receipt of a DISCONNECT message. This message prompts the Incoming Side to disconnect from the bearer channel. Once the channel used for the call has been disconnected, the Incoming Side shall send a RELEASE message to the Outgoing Side, start timer T308, and enter the Release request state.

On receipt of the RELEASE message the Outgoing Side shall cancel timer T305, release the bearer channel, send a RELEASE COMPLETE message, and return to the Null state.

Following the receipt of a RELEASE COMPLETE message from the Outgoing Side, the Incoming Side shall stop timer T308, release the bearer channel, and return to the Null state.

If the Outgoing Side does not receive a RELEASE message in response to the DISCONNECT message before timer T305 expires, it shall send a RELEASE message to the Incoming Side with the cause number originally contained in the DISCONNECT message, start timer T308 and enter the Release request state.

If in the Release request state, a RELEASE COMPLETE message is not received before timer T308 expires, the side that expected the message shall return to the Null state.

Clearing by the called user employing user-provided tones/announcements

In addition to the procedures described above, if the requested bearer signals a speech call, the Outgoing Side may apply in-band tones/announcements in the clearing phase. When in-band tones/announcements are provided, the Outgoing Side will first complete the bearer channel (if not already available), and next send the DISCONNECT message containing progress indicator *#8, in-band information or appropriate pattern is now available* .

Upon receipt of this message, the Incoming Side may connect (if not already connected) to the bearer channel to receive the in-band tone/announcement, and enter the Disconnect indication state.

The Incoming Side may subsequently continue clearing (before the receipt of a RELEASE from the Outgoing Side) by disconnecting from the bearer channel, sending a RELEASE message, starting timer T308, and entering the Release request state.

2.3.2 Abnormal Call Clearing

Under normal conditions, call clearing is initiated when either side sends a DISCONNECT message and follows the procedures defined in Section 2.3.1 on page 460. The only exceptions to the above rule are as follows:

a In response to a SETUP message, the Incoming Side can reject a call (e.g. because of unavailability of suitable resources) by responding with a RELEASE COMPLETE message provided no other response has previously been sent, and enter the Null state

b In case of a multi-point configuration, non-selected user call clearing will be initiated with RELEASE message(s) from the Outgoing Side (Section 2.2.8 on page 457)

c In case of a multi-point configuration, where the SETUP message is delivered on an connection-less channel, if a remote (calling) user disconnect indication is received during call establishment, any Incoming Side which has responded, or subsequently responds, shall be cleared by a RELEASE message, and the procedures of Section 2.3.1 on page 460 are then followed for that user. The Outgoing Side enters the Null state upon completion of clearing procedures for all responding Incoming Sides.

2.3.3 Clear Collision

Clear collision occurs when the Incoming and the Outgoing Sides simultaneously transfer DISCONNECT messages. When either side receives a DISCONNECT message while in the Disconnect request state, the side shall stop timer T305, disconnect the bearer channel (if not disconnected), send a RELEASE message, start timer T308, and enter the Release request state.

Clear collision can also occur when both sides simultaneously transfer RELEASE messages. The entity receiving such a RELEASE message while within the Release request state shall stop timer T308, release the bearer channel, and enter the Null state (without sending or receiving a RELEASE COMPLETE message).

2.3.4 Call Clearing Message Flow

The figure below provides the complete view on the messages exchanged during normal Call Clearing, as described in the sections above. All messages are mandatory.

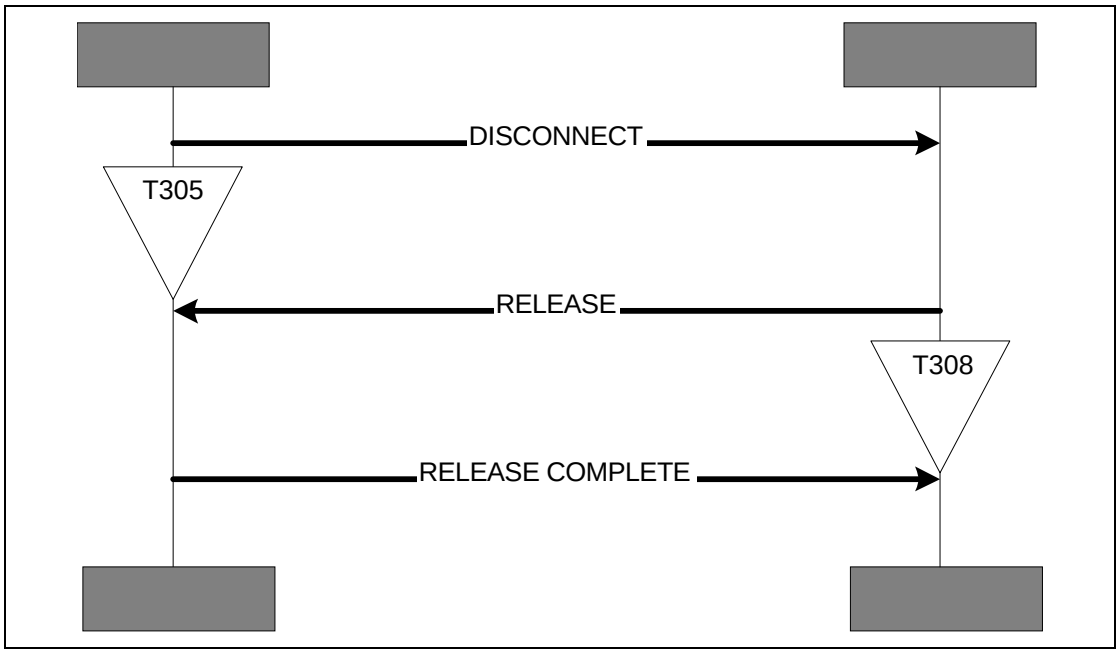


Figure 2.2: Call clearing message flow

3 GROUP MANAGEMENT (GM)

3.1 OVERVIEW

The Group Management entity provides procedures for managing a group of devices.

The following procedures are supported:

- Obtain access rights (Section 3.3 on page 465)
  enables the requesting device to use the telephony services of another device, part of a group of devices
- Configuration distribution (Section 3.4 on page 466)
  facilitates the handling and operation of a group of devices
- Fast inter-member access (Section 3.5 on page 467)
  enables faster contact establishment between devices of the same group

A connection-oriented L2CAP channel between devices shall be available before any of the GM procedures can operate.

For group management, the concept of Wireless User Group (WUG) is used.

3.2 THE WIRELESS USER GROUP

3.2.1 Description

A WUG consists of a number of Bluetooth units supporting TCS. One of the devices is called the WUG master. The WUG master is typically a gateway, providing the other Bluetooth devices – called WUG members – with access to an external network. All members of the WUG in range are members of a piconet (active or parked). Master of this piconet is always the WUG master.

The main relational characteristics of a WUG are:

- All units that are part of a WUG know which unit is the WUG master and which other units are member of this WUG. WUG members receive this information from the WUG master.
- When a new unit has paired with the WUG master, it is able to communicate and perform authentication and encryption with any other unit part of the WUG without any further pairing/initialization. The WUG master provides the required authentication and encryption parameters to the WUG members.

Both relational characteristics are maintained through the Configuration distribution procedure.

3.2.2 Encryption within the WUG

In order to allow for encrypted transmission on the connectionless L2CAP channel, the WUG master issues a temporary key (K_{master}). As a Bluetooth unit is not capable of switching between two or more encryption keys in real time, this key is normally also used for encrypted transmission on the connection-oriented channel (individually addressed traffic). Since the WUG master piconet may be in operation for extended periods without interruption, the K_{master} shall be changed periodically.

In order to allow for authentication and encryption to be performed between WUG members, the WUG master may use the Configuration distribution procedure to issue link keys that the WUG members use for communication with each other. Just as if pairing had created these keys, the keys are unique to a pair of WUG members and hence a WUG member uses a different key for every other WUG member it connects to.

The Configuration distribution shall always be performed using encrypted links. The K_{master} shall not be used for encryption; rather the WUG master shall ensure that the semi-permanent key for the specific WUG member addressed shall be used.

3.2.3 Unconscious pairing

For TCS, pairing a device with the WUG master implies pairing a device with all members of the WUG.This is achieved using the Configuration distribution procedure. This avoids the user of the device having to pair with each and every device of the WUG individually.

In Bluetooth, pairing is not related to a specific service but rather to a specific device. After pairing, all services provided by a device are accessible, if no further application- or device-specific protection is provided.

Without further provisions, pairing a device with the WUG master implies that all services provided by the new device are accessible to all other WUG members. And vice versa, without further provisions, the new device can access all services provided by other WUG members.

For this reason, implementers of TCS – and in particular the Configuration distribution procedure – are recommended to add provisions where:

1. a new device entering the WUG is not mandated to initiate the Obtain access rights procedure to become a WUG member, and is consequently only able to use the services provided by the WUG master (gateway)
2. a WUG master can reject a request to obtain access rights
3. a WUG member is not forced to accept the pairing information received during the Configuration distribution

This applies in particular to devices offering more than just TCS- related services.

3.3 OBTAIN ACCESS RIGHTS

Using the Obtain access rights procedure, a device can obtain the rights to use the telephony services provided by another device, part of a WUG.

3.3.1 Procedure description

A device requests access rights by sending an ACCESS RIGHTS REQUEST message and starting timer T401. Upon receipt of the ACCESS RIGHTS REQUEST message, the receiving device accepts the request for access rights by sending an ACCESS RIGHTS ACCEPT.

When the requesting device receives the ACCESS RIGHTS ACCEPT, it shall stop timer T401. Then, the access rights procedure has completed successfully.

If no response has been received before the expiration of timer T401, the requesting device shall consider the request for access rights to be denied.

If, upon receipt of the ACCESS RIGHTS REQUEST message, the receiving device is for some reason unable to accept the access rights, it shall reply with an ACCESS RIGHTS REJECT message. Upon receipt of an ACCESS RIGHTS REJECT message, the requesting device shall stop timer T401 and consider the request for access rights to be denied.

3.3.2 Message flow

The figure below provides the complete view on the messages exchanged during the Obtain access rights procedure.

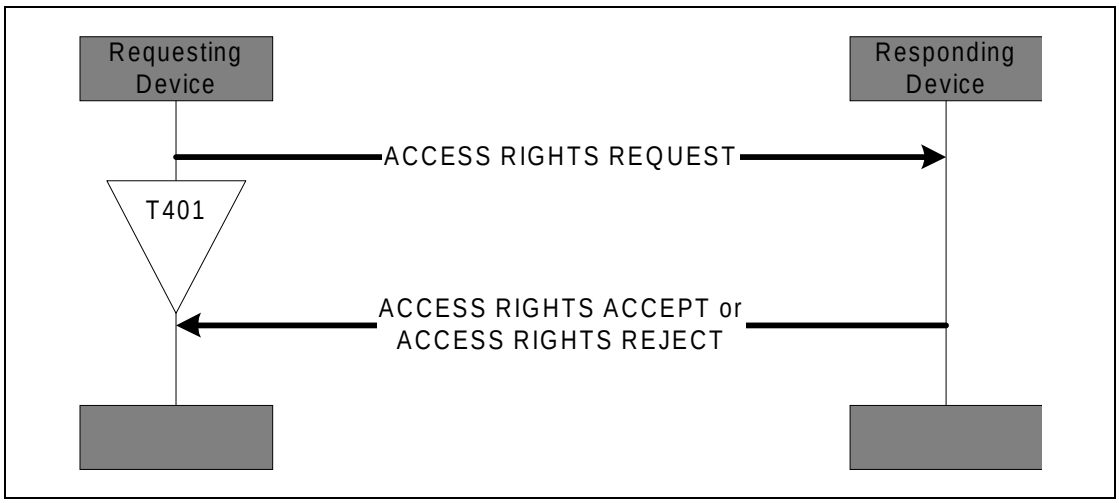


Figure 3.1: Obtain access rights message flow

3.4 CONFIGURATION DISTRIBUTION

The units in the WUG need to be informed about changes in the WUG; e.g. when a unit is added or removed. The Configuration distribution procedure is used to exchange this data.

When a WUG configuration change occurs, the WUG master initiates the Configuration distribution procedure on all WUG members. The WUG master keeps track of which WUG members have been informed of WUG configuration changes.

Some WUG members may be out of range and may therefore not be reached. The update of these WUG members will be performed when these members renew contact with the WUG master.

3.4.1 Procedure Description

The WUG master initiates the Configuration distribution procedure by starting timer T403, and transferring the INFO SUGGEST message. The INFO SUGGEST message contains the complete WUG configuration information. Upon receipt of the INFO SUGGEST message, the WUG member shall send an INFO ACCEPT message, to acknowledge the proper receipt of the WUG configuration information.

When the WUG master receives the INFO ACCEPT, the timer T401 is stopped, and the Configuration distribution procedure has completed successfully. On expiry of timer T403, the Configuration distribution procedure is terminated.

3.4.2 Message flow

The figure below provides the complete view on the messages exchanged during the Configuration distribution procedure, as described in the sections above.

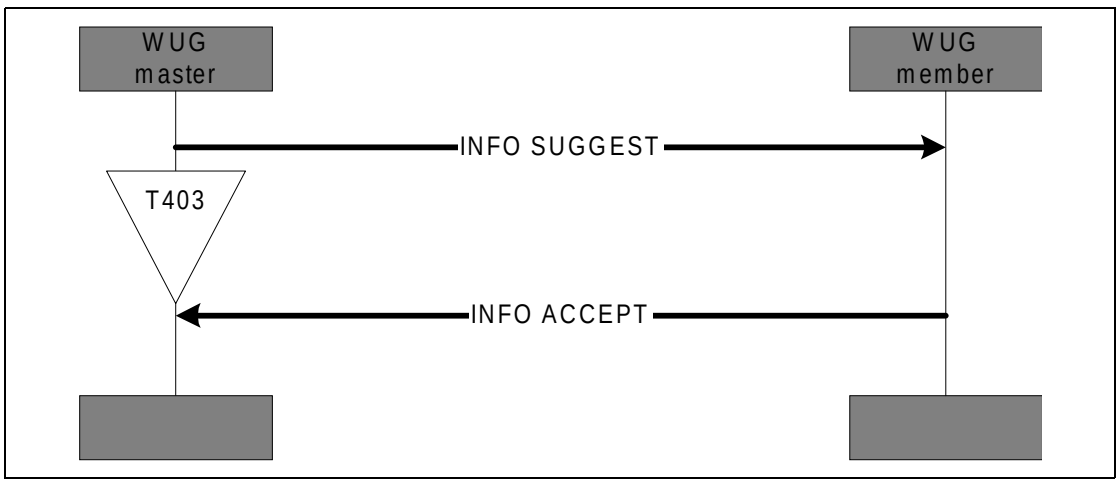


Figure 3.2: Configuration distribution message flow

3.5 FAST INTER-MEMBER ACCESS

When two WUG members are both active in the WUG master piconet, a WUG member can use the Fast inter-member access procedure to obtain fast access to another WUG member. With the Fast inter-member access procedure, the originating WUG member obtains clock information from the terminating WUG member and forces the terminating WUG member to go into PAGE_SCAN for a defined period (T406).

3.5.1 Listen Request

The originating WUG member initiates the Fast inter-member access procedure by starting timer T404 and transferring the LISTEN REQUEST message to the WUG master, indicating the WUG member with which it wishes to establish contact.

If, before expiry of timer T404, the originating WUG member receives no response to the LISTEN REQUEST message, the originating WUG member shall terminate the procedure.

3.5.2 Listen Accept

Upon receipt of the LISTEN REQUEST message, the WUG master checks that the indicated WUG member is part of the WUG. If this is the case, the WUG master initiates the Fast inter-member access towards the terminating WUG member side by starting timer T405 and sending the LISTEN SUGGEST message to the terminating WUG member.

Upon receipt of the LISTEN SUGGEST message, the terminating WUG member confirms the suggested action (internal call) by sending a LISTEN ACCEPT message to the WUG master. This message contains the terminating WUG member's clock offset. After sending the LISTEN ACCEPT, the terminating WUG member shall go to PAGE-SCAN state, for T406 seconds, to enable connection establishment by the originating WUG member.

Upon receipt of the LISTEN ACCEPT message, the WUG master stops timer T405, and informs the originating WUG member of the result of the WUG fast inter-member access by sending a LISTEN ACCEPT message. This message contains the terminating WUG member's clock offset. Upon receipt of the LISTEN ACCEPT message, the originating WUG member stops timer T404, and starts paging the terminating WUG member.

If no response to the LISTEN SUGGEST message is received by the WUG master before the first expiry of timer T405, then the WUG master shall terminate the Fast inter-member access procedure by sending a LISTEN REJECT message to both originating and terminating WUG member using cause #102, *recovery on timer expiry*.

3.5.3 Listen Reject by the WUG Master

If the WUG master rejects the Fast inter-member access procedure, it sends a LISTEN REJECT message to the originating WUG member.

Valid cause values are:
#1, *Unallocated (unassigned) number* (when the indicated WUG member is not part of the WUG)
#17, *User busy* (in case terminating WUG member is engaged in an external call)
#20, *Subscriber absent* (upon failure to establish contact with the terminating WUG member), or
any cause value indicated in a LISTEN REJECT message received from/sent to the terminating WUG member.

Upon receipt of the LISTEN REJECT message, the originating WUG member stops timer T404, and terminates the procedure.

3.5.4 Listen Reject by the WUG Member

If the terminating WUG member rejects the suggested action received in the LISTEN SUGGEST message, it sends a LISTEN REJECT message to the WUG master. Valid cause value is #17, *User busy* (in case terminating WUG member is engaged in another internal call).

Upon receipt of the LISTEN REJECT, the WUG master stops timer T405, and continues as described in Section 3.5.3 on page 468.

3.5.5 Message flow

The figure below provides a view of the messages exchanged during Fast inter-member access, as described in the sections above. A successful Fast inter-member access procedure ends with the terminating WUG member going into page scan, thus allowing the originating WUG member to contact him directly.

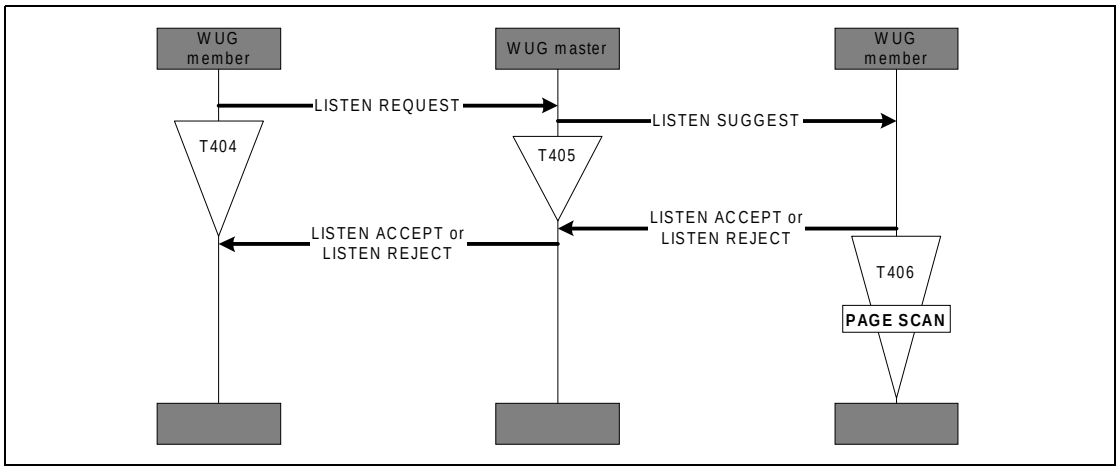


Figure 3.3: Fast inter-member access message flow

4 CONNECTIONLESS TCS (CL)

A connectionless TCS message can be used to exchange signalling information without establishing a TCS call. It is thus a connectionless service offered by TCS.

A connectionless TCS message is a CL INFO message (as defined in Section 6.3.1 on page 484).

A connection-oriented L2CAP channel between the Outgoing and Incoming Side shall be available before a CL INFO message can be sent.

Note: In the case of a connection-oriented channel, it may choose to delay the termination of the channel for a defined period to exchange more CL INFO messages.

Alternatively, in a multi-point configuration (see Section 1.2 on page 449), a connectionless L2CAP channel may be used and, if so, shall be available before a CL INFO can be sent.

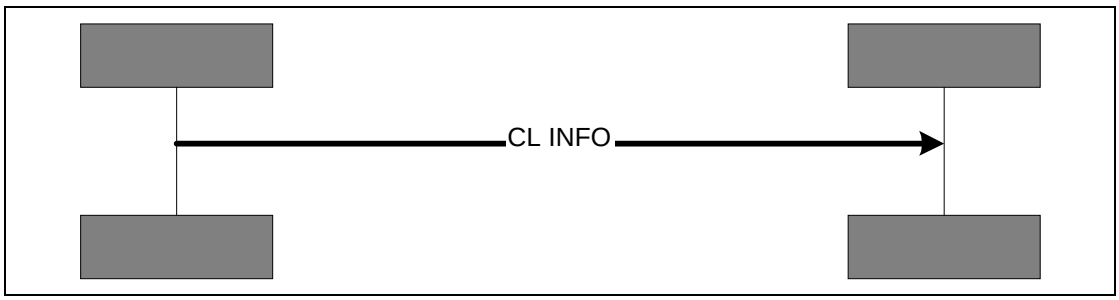


Figure 4.1: Connectionless TCS message flow

5 SUPPLEMENTARY SERVICES (SS)

The TCS provides explicit support for only one supplementary service, the Calling Line Identity (seeSection 5.1 on page 470).

For supplementary services provided by an external network, using DTMF sequences for the activation/de-activation and interrogation of supplementary services, the DTMF start & stop procedure is supported (see Section 5.2 on page 470). This procedure allows both finite and infinite tone lengths.

Section 5.3 on page 472 specifies how a specific supplementary service, provided by an external network, called register recall is supported.

For other means of supplementary service control, no explicit support is specified. Support may be realized by either using the service call, or use the company specific information element, or a combination.

5.1 CALLING LINE IDENTITY

To inform the Incoming Side of the identity of the originator of the call, the Outgoing Side may include the calling party number information element (see Section 7.4.6 on page 495) in the SETUP message transferred as part of the call request. If the CLIP information is first available after having sent the SETUP message, it is also allowed to send the CLIP information included in a INFORMATION message.

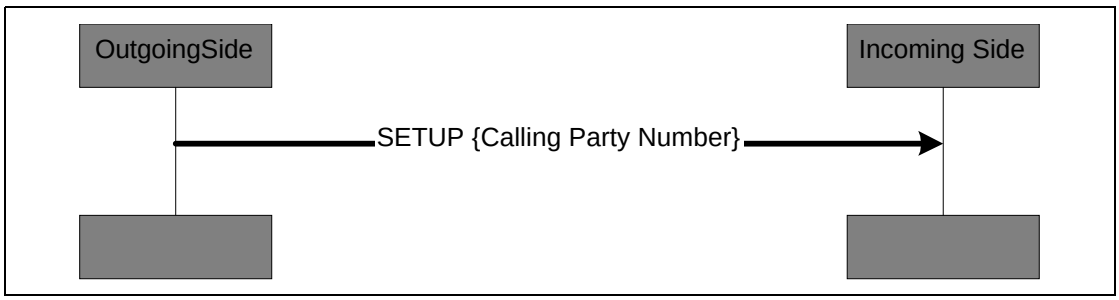


Figure 5.1: Calling line identity message flow

5.2 DTMF START & STOP

The DTMF start & stop procedure is supported to provide supplementary service control on PSTN type of networks.

In principle DTMF messages can be initiated by either (Outgoing or Incoming) Side; in practice, however, the Side (gateway) connected to the external PSTN network will be the recipient.

DTMF messages can be transmitted only in the active state of a call. Tone generation shall end when the call is disconnected.

5.2.1 Start DTMF request

A user may cause a DTMF tone to be generated; e.g. by depression of a key. The relevant action is interpreted as a requirement for a DTMF digit to be sent in a START DTMF message on an established signalling channel. This message contains the value of the digit to be transmitted (0, 1...9, A, B, C, D, *, #).

Only a single digit will be transferred in each START DTMF message.

5.2.2 Start DTMF response

The side receiving the START DTMF message will reconvert the received digit back into a DTMF tone which is applied toward the remote user, and return a START DTMF ACKNOWLEDGE message to the initiating side. This acknowledgment may be used to generate an indication as a feedback for a successful transmission.

If the receiving side cannot accept the START DTMF message, a START DTMF REJECT message will be sent to the initiating side, using cause #29, *Facility rejected*, indicating that sending DTMF is not supported by the external network.

5.2.3 Stop DTMF request

When the user indicates the DTMF sending should cease (e.g. by releasing the key) the initiating side will send a STOP DTMF message to the other side.

5.2.4 Stop DTMF response

Upon receiving the STOP DTMF message, the receiving side will stop sending the DTMF tone (if still being sent) and return a STOP DTMF ACKNOWLEDGE message to the initiating side.

5.2.5 Message flow

The figure below provides a view of the messages exchanged when a single DTMF tone needs to be generated.

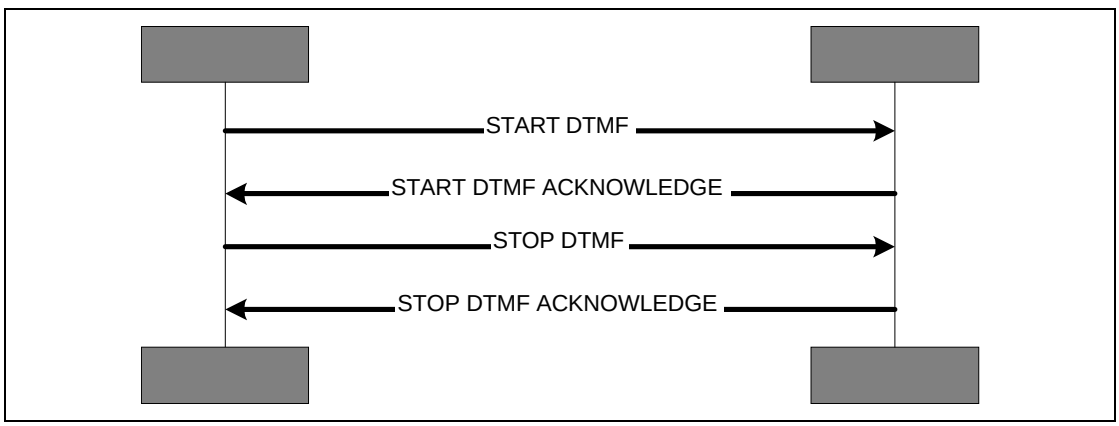


Figure 5.2: DTMF start & stop message flow

5.3 REGISTER RECALL

Register recall means to seize a register (with dial tone) to permit input of further digits or other action. In some markets, this is referred to as 'hook flash'. Register recall is supported by sending an INFORMATION message with a keypad facility information element, indicating 'register recall' (value 16H). Further digits are sent using the procedures as indicated in Section 5.2 above.

6 MESSAGE FORMATS

This section provides an overview of the structure of messages used in this specification, and defines the function and information contents (i.e. semantics) of each message.

Whenever a message is sent according to the procedures of Sections 2, 3 and 4, it shall contain the mandatory information elements, and optionally any combination of the optional information elements specified in this section for that message.

A message shall always be delivered in a single L2CAP packet. The start of a message (the LSB of the first octet) shall be aligned with the start of the L2CAP payload.

Each definition includes:

a) A brief description of the message direction and use

b) A table listing the information elements in order of their appearance in the message (same relative order for all message types)

c) Indications for each information element in the table, specifying –

- the section of this specification describing the information element
- whether inclusion in mandatory (’M’) or optional (’O’)
- the length (or length range) of the information element, where ’*’ denotes an undefined maximum length which may be application dependent.

d) Further explanatory notes, as necessary

All message formats are denoted in octets.

6.1 CALL CONTROL MESSAGE FORMATS

6.1.1 ALERTING

This message is sent by the incoming side to indicate that the called user alerting has been initiated.

Message Type: ALERTING

Direction: incoming to outgoing

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Bearer capability	7.4.3	O Note 1)	4(26)
Progress indicator	7.4.13	O	2
SCO Handle	7.4.14	O	2
Destination CID	7.4.11	O	4
Company specific	7.4.9	O	3-*

Table 6.1: ALERTING message content

Note 1: Allowed only in the first message sent by the incoming side.

6.1.2 CALL PROCEEDING

This message is sent by the incoming side to indicate that the requested call establishment has been initiated and no more call establishment information will be accepted.

Message Type: CALL PROCEEDING

Direction: incoming to outgoing

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Bearer capability	7.4.3	O Note 1)	4(26)
Progress indicator	7.4.13	O	2
SCO Handle	7.4.14	O	2
Destination CID	7.4.11	O	4
Company specific	7.4.9	O	3-*

Table 6.2: CALL PROCEEDING message content

Note 1: Allowed only in the first message sent by the incoming side.

6.1.3 CONNECT

This message is sent by the incoming side to indicate call acceptance by the called user.

Message Type: CONNECT

Direction: incoming to outgoing

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Bearer capability	7.4.3	O[Note 1)]	4(26)
SCO Handle	7.4.14	O	2
Company specific	7.4.9	O	3-*

Table 6.3: CONNECT message content

Note 1: Allowed only in the first message sent by the incoming side.

6.1.4 CONNECT ACKNOWLEDGE

This message is sent by the outgoing side to acknowledge the receipt of a CONNECT message.

Message Type: CONNECT ACKNOWLEDGE

Direction: outgoing to incoming

Information Element	Ref.	Type	Length
Message type	7.3	M	1
SCO Handle	7.4.14	O	2
Destination CID	7.4.11	O	4
Company specific	7.4.9	O	3-*

Table 6.4: CONNECT ACKNOWLEDGE message content

6.1.5 DISCONNECT

This message is sent by either side as an invitation to terminate the call.

Message Type: DISCONNECT

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Cause	7.4.7	O	2
Progress indicator	7.4.13	O	2
SCO Handle	7.4.14	O	2
Destination CID	7.4.11	O	4
Company specific	7.4.9	O	3-*

Table 6.5: DISCONNECT message content

6.1.6 INFORMATION

This message is sent by either side to provide additional information during call establishment (in case of overlap sending).

Message Type: INFORMATION

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Sending complete	7.4.15	O	1
Keypad facility	7.4.12	O	2
Calling party number	7.4.6	O	3-*
Called party number	7.4.5	O	3-*
Audio control	7.4.2	O	3-*
Company specific	7.4.9	O	3-*

Table 6.6: INFORMATION message content

6.1.7 PROGRESS

This message is sent by the incoming side to indicate the progress of a call in the event of interworking or by either side in the call with the provision of optional in-band information/patterns.

Message Type: PROGRESS

Direction: incoming to outgoing

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Progress indicator	7.4.13	M	2
SCO Handle	7.4.14	O	2
Destination CID	7.4.11	O	4
Company specific	7.4.9	O	3-*

Table 6.7: PROGRESS message content

6.1.8 RELEASE

This message is used to indicate that the device sending the message had disconnected the channel (if any) and intends to release the channel, and that receiving device should release the channel after sending RELEASE COMPLETE.

Message Type: RELEASE

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Cause	7.4.7	O(Note 1)	2
SCO Handle	7.4.14	O	2
Company specific	7.4.9	O	3-*

Table 6.8: RELEASE message content

Note 1: Mandatory in the first call clearing message.

6.1.9 RELEASE COMPLETE

This message is used to indicate that the device sending the message has released the channel (if any), and that the channel is available for re-use.

Message Type: RELEASE COMPLETE

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Cause	7.4.7	O[Note 1)]	2
SCO Handle	7.4.14	O	2
Company specific	7.4.9	O	3-*

Table 6.9: RELEASE COMPLETE message content

Note 1: Mandatory in the first call clearing message.

6.1.10 SETUP

This message is sent by the outgoing side to initiate call establishment.

Message Type:

Direction:

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Call class	7.4.4	M	2
Sending complete	7.4.15	O	1
Bearer capability	7.4.3	O	4(26)
Signal	7.4.16	O	2
Calling party number	7.4.6	O	3-*
Called party number	7.4.5	O	3-*
Company specific	7.4.9	O	3-*

Table 6.10: SETUP message content

6.1.11 SETUP ACKNOWLEDGE

This message is sent by the incoming side to indicate that call establishment has been initiated, but additional information may be required.

Message Type: SETUP ACKNOWLEDGE

Direction: incoming to outgoing

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Bearer capability	7.4.3	O[Note 1)]	4(26)
Progress indicator	7.4.13	O	2
SCO Handle	7.4.14	O	2
Destination CID	7.4.11	O	4
Company specific	7.4.9	O	3-*

Table 6.11: SETUP ACKNOWLEDGE message content

Note 1: Allowed only in the first message sent by the incoming side.

6.1.12 Start DTMF

This message contains the digit the other side should reconvert back into a DTMF tone, which is then applied towards the remote user.

Message Type: Start DTMF

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Keypad facility	7.4.12	M	2

Table 6.12: Start DTMF message content

6.1.13 Start DTMF Acknowledge

This message is sent to indicate the successful initiation of the action required by the Start DTMF message.

Message Type: Start DTMF Acknowledge

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Keypad facility	7.4.12	M	2

Table 6.13: Start DTMF Acknowledge message content

6.1.14 Start DTMF Reject

This message is sent to indicate that the other side cannot accept the Start DTMF message.

Message Type: Start DTMF Reject

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Cause	7.4.7	O	2

Table 6.14: Start DTMF Reject message content

6.1.15 Stop DTMF

This message is used to stop the DTMF tone sent towards the remote user.

Message Type: Stop DTMF

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1

Table 6.15: Stop DTMF message content

6.1.16 Stop DTMF Acknowledge

This message is sent to indicate that the sending of the DTMF tone has been stopped.

Message Type: Stop DTMF Acknowledge

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Keypad facility	7.4.12	M	2

Table 6.16: Stop DTMF Acknowledge message content

6.2 GROUP MANAGEMENT MESSAGE FORMATS

6.2.1 ACCESS RIGHTS REQUEST

This message is sent by the initiating side to obtain access rights.

Message Type: ACCESS RIGHTS REQUEST

Direction:

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Company specific	7.4.9	O	3-*

Table 6.17: ACCESS RIGHTS REQUEST message content

6.2.2 ACCESS RIGHTS ACCEPT

This message is sent by the responding side to indicate granting of access rights.

Message Type: ACCESS RIGHTS ACCEPT

Direction:

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Company specific	7.4.9	O	3-*

Table 6.18: ACCESS RIGHTS ACCEPT message content

6.2.3 ACCESS RIGHTS REJECT

This message is sent by the responding side to indicate denial of access rights.

Message Type: ACCESS RIGHTS REJECT

Direction:

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Company specific	7.4.9	O	3-*

Table 6.19: ACCESS RIGHTS REJECT message content

6.2.4 INFO SUGGEST

This message is sent by the WUG master to indicate that a change has occurred in the WUG configuration.

Message Type: INFO SUGGEST

Direction: WUG master to WUG member

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Configuration Data	7.4.10	M	*
Company specific	7.4.9	O	3-*

Table 6.20: INFO SUGGEST message content

6.2.5 INFO ACCEPT

This message is sent by the WUG member to indicate the acceptance of the updated WUG configuration.

Message Type: INFO ACCEPT

Direction: WUG member to WUG master

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Company specific	7.4.9	O	3-*

Table 6.21: INFO ACCEPT message content

6.2.6 LISTEN REQUEST

This message is sent by a WUG member to indicate to the WUG master the request for a Fast inter-member access to the indicated WUG member.

Message Type: LISTEN REQUEST

Direction: WUG member to WUG master

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Called party number	7.4.6	M	3-*
Company specific	7.4.9	O	3-*

Table 6.22: LISTEN REQUEST message content

6.2.7 LISTEN SUGGEST

This message is sent by a WUG master to indicate to the WUG member the request for a Fast inter-member access.

Message Type: LISTEN SUGGEST

Direction: WUG master to WUG member

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Company specific	7.4.9	O	3-*

Table 6.23: LISTEN SUGGEST message content

6.2.8 LISTEN ACCEPT

This message is sent to indicate the acceptance of the previous request for a Fast inter-member access.

Message Type: LISTEN ACCEPT

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Clock offset	7.4.8	O	4
Company specific	7.4.9	O	3-*

Table 6.24: LISTEN ACCEPT message content

6.2.9 LISTEN REJECT

This message is sent to indicate the rejection of the previous request for a Fast inter-member access.

Message Type: LISTEN REJECT

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Cause	7.4.7	O	2
Company specific	7.4.9	O	3-*

Table 6.25: LISTEN REJECT message content

6.3 TCS CONNECTIONLESS MESSAGE FORMATS

6.3.1 CL INFO

This message is sent by either side to provide additional information in a connectionless manner.

Message Type: CL INFO

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	M	1
Audio control	7.4.2	O	3-*
Company specific	7.4.9	O	3-*

Table 6.26: CL INFO message content

7 MESSAGE CODING

The figures and text in this section describe message contents. Within each octet, the bit designated ’bit 1’ is transmitted first, followed by bit 2, 3, 4, etc. Similarly, the octet shown at the top of the figure is sent first.

Whenever a message is sent, according to the procedures of Sections 2, 3 and 4, it shall be coded as specified in this section.

7.1 OVERVIEW

The coding rules follow ITU-T Recommendation Q.931, but is tailored to the specific needs of TCS.

Every message consists of:

a) Protocol discriminator

b) Message type, and

c) Other information elements, as required

The Protocol discriminator and Message type is part of every TCS message, while the other information elements are specific to each message type.

8	7	6	5	4	3	2	1	
Protocol discriminator			Message type					octet 1
Other information elements as required								octet 2

Table 7.1: General message format

A particular information element shall be present only once in a given message.

The term ’default’ implies that the value defined shall be used in the absence of any assignment or negotiation of alternative values.

For notation purposes – when a field extends over more than one octet, the order of bit values progressively decreases as the octet number increases. The least significant bit of the field is represented by the lowest numbered bit of the highest-numbered octet of the field. In general, bit 1 of each octet contains the least significant bit of a field.

7.2 PROTOCOL DISCRIMINATOR

The purpose of the protocol discriminator is to distinguish the TCS messages into different functional groups. The protocol discriminator is the first part of every message.

The protocol discriminator is coded according to Figure 7.1 and Table 7.2.

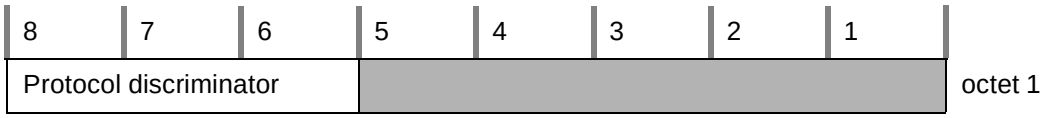


Figure 7.1: Protocol discriminator

Bits			
8	**7**	**6**	
0	0	0	Bluetooth TCS Call Control
0	0	1	Bluetooth TCS Group management
0	1	0	Bluetooth TCS Connectionless
All other values reserved			

Table 7.2: Protocol discriminator

7.3 MESSAGE TYPE

The purpose of the message type is to identify the function of the message being sent.

The Message type is the first part of every message and it is coded as shown in Figure 7.2 and Table 7.3.

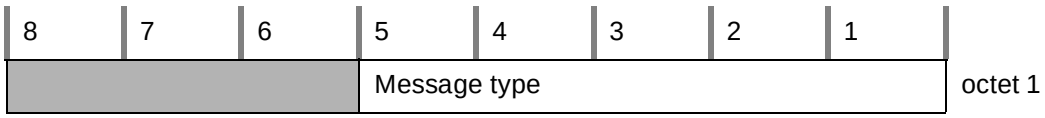


Figure 7.2: Message type

Bits					
5	**4**	**3**	**2**	**1**	
					Call Control messages
					Call Establishment
0	0	0	0	0	ALERTING

Table 7.3: Message type

Bits					
5	**4**	**3**	**2**	**1**	
0	0	0	0	1	CALL PROCEEDING
0	0	0	1	0	CONNECT
0	0	0	1	1	CONNECT ACKNOWLEDGE
0	0	1	0	0	PROGRESS
0	0	1	0	1	SETUP
0	0	1	1	0	SETUP ACKNOWLEDGE
					Call clearing
0	0	1	1	1	DISCONNECT
0	1	0	0	0	RELEASE
0	1	0	0	1	RELEASE COMPLETE
					Miscellaneous
0	1	0	1	0	INFORMATION
1	0	0	0	0	START DTMF
1	0	0	0	1	START DTMF ACKNOWLEDGE
1	0	0	1	0	START DTMF REJECT
1	0	0	1	1	STOP DTMF
1	0	1	0	0	STOP DTMF ACKNOWLEDGE
					Group management messages
0	0	0	0	0	INFO SUGGEST
0	0	0	0	1	INFO ACCEPT
0	0	0	1	0	LISTEN REQUEST
0	0	0	1	1	LISTEN ACCEPT
0	0	1	0	0	LISTEN SUGGEST
0	0	1	0	1	LISTEN REJECT
0	0	1	1	0	ACCESS RIGHTS REQUEST
0	0	1	1	1	ACCESS RIGHTS ACCEPT
0	1	0	0	0	ACCESS RIGHTS REJECT
					Connectionless messages
0	0	0	0	0	CL INFO

Table 7.3: Message type

7.4 OTHER INFORMATION ELEMENTS

7.4.1 Coding rules

The coding of other information elements follows the coding rules described below.

Three categories of information elements are defined:

a) single octet information elements (see Figure 7.3 on pag e488)

b) double octet information element (see Figure 7.4 on pa g e488)

c) variable length information elements (see Figure 7.5 on page488).

Table 7.4 on page 488 summarizes the coding of the information element identified bits for those information elements used in this specification.

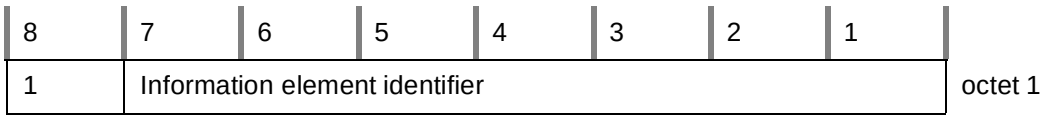


Figure 7.3: Single octet information element format

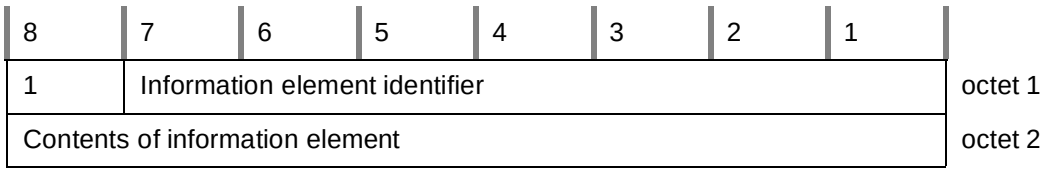


Figure 7.4: Double octet information element format

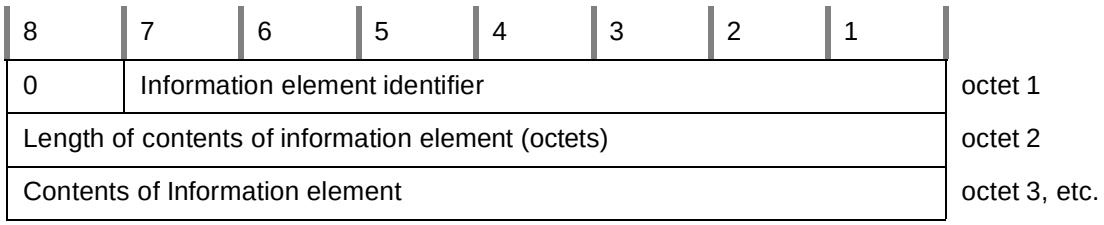


Figure 7.5: Variable length information element format

Coding									Ref.	Max Length (octets)
8	7	6	5	4	3	2	1			
1								*Single octet information elements*		
	0	1	0	0	0	0	1	Sending complete	7.4.15	1
1								*Double octet information elements*		

Table 7.4: Information element identifier coding

Coding									Ref.	Max Length (octets)
8	7	6	5	4	3	2	1			
	1	0	0	0	0	0	0	Call class	7.4.4	2
	1	0	0	0	0	0	1	Cause	7.4.7	2
	1	0	0	0	0	1	0	Progress indicator	7.4.13	2
	1	0	0	0	0	1	1	Signal	7.4.16	2
	1	0	0	0	1	0	0	Keypad facility	7.4.12	2
	1	0	0	0	1	0	1	SCO handle	7.4.14	2
0								*Variable length information elements*		
	0	0	0	0	0	0	0	Clock offset	7.4.8	4
	0	0	0	0	0	0	1	Configuration data	7.4.2	*
	0	0	0	0	0	1	0	Bearer capability	7.4.3	4(26)
	0	0	0	0	0	1	1	Destination CID	7.4.11	4
	0	0	0	0	1	0	0	Calling party number	7.4.6	*
	0	0	0	0	1	0	1	Called party number	7.4.5	*
	0	0	0	0	1	1	0	Audio control	7.4.2	*
	0	0	0	0	1	1	1	Company specific	7.4.9	*

Table 7.4: Information element identifier coding

The descriptions of the information elements below are organized in alphabetical order. However, there is a particular order of appearance for each information element in a message. The code values of the information element identifier for the variable length formats are assigned in ascending numerical order, according to the actual order of appearance of each information element in a message. This allows the receiving devices to detect the presence or absence of a particular information element without scanning through an entire message.

Where the description of information elements in this specification contains spare bits, these bits are indicated as being set to ’0’. In order to allow compatibility with future implementation, messages should not be rejected simply because a spare bit is set to ’1’.

The second octet of a variable length information element indicates the total length of the contents of that information element regardless of the coding of the first octet (i.e. the length is calculated starting from octet 3). It is the binary coding of the number of octets of the contents, with bit 1 as the least significant bit (2°).

An optional variable-length information element may be present, but empty (zero length). The receiver should interpret this as if that information element

was absent. Similarly, an absent information element should be interpreted by the receiver as if that information element was empty.

7.4.2 Audio control

The purpose of the Audio control information elements is to indicate information relating to the control of audio.

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	1	1	0	1
Length of contents of information element (octets)								2
Control information								3

Figure 7.6:

Control information (octet 3)								
	Bits							
	7	**6**	**5**	**4**	**3**	**2**	**1**	
	0	0	0	0	0	0	0	Volume increase
	0	0	0	0	0	0	1	Volume decrease
	0	0	0	0	0	1	0	Microphone gain increase
	0	0	0	0	0	1	1	Microphone gain decrease
	0	X	X	X	X	X	X	Reserved for Bluetooth standardization
	1	X	X	X	X	X	X	Company specific

Table 7.5: Audio Control information element coding

7.4.3 Bearer capability

The purpose of the Bearer capability information elements is to indicate a requested or available bearer service.

If this information element is absent, the default Bearer capability is Link type Synchronous Connection-Oriented with packet type HV3, using CVSD coding for the User information layer 1.

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	0	1	0	1
Length of contents of information element (octets)								2
Link type								3

Figure 7.7:

Link type element coding = 00000000 (SCO)

User information layer 1	Packet type	4

Figure 7.8:

Link type element coding = 00000001 (ACL)

Flags		4
Service type		5
Token Rate		6
		7
		8
		9
Token Bucket Size (bytes)		10
		11
		12
		13
Peak Bandwidth (bytes/second)		14
		15
		16
		17
Latency (microseconds)		18
		19
		20
		21
Delay Variation (microseconds)		22
		23
		24
		25
User information layer 3	User information layer 2	26

Figure 7.9:

Note: the Quality of Service is repeated at TCS level, as only TCS has the knowledge of end-to-end Quality of Service requirements.

Link type (octet 3)

Bits

8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	Synchronous Connection-Oriented
0	0	0	0	0	0	0	1	Asynchronous Connection-Less
0	0	0	0	0	0	1	0	None

All other values are reserved

Octet 4 coding (Link type element coding = 000000000)

Packet type (octet 4)

Bits

5	4	3	2	1	
0	0	1	0	1	HV1
0	0	1	1	0	HV2
0	0	1	1	1	HV3
0	1	0	0	0	DV

All other values are reserved

User information layer 1 (octet 4)

Bits

8	7	6	
0	0	1	CVSD
0	1	0	PCM A-law
0	1	1	PCM µ-law

All other values reserved

Octets 4-26 coding (Link type element coding = 000000001)

The details of the coding Octets 4-25 can be found in L2CAP, see L2CAP, Section 6 on page 297

User information layer 2 (octet 26)

Bits

4	3	2	1	
0	0	0	0	RFCOMM over L2CAP

All other values are reserved

User information layer 3 (octet 26)

Bits

8	7	6	5	
0	0	0	0	Not specified
0	0	0	1	PPP
0	0	1	0	IP

All other values reserved

Octet 4 coding (Link type element coding = 000000010)

Octet 4 is absent

Table 7.6: Bearer capability information element coding

7.4.4 Call class

The purpose of the Call class is to indicate the basic aspects of the service requested. This element allows the user to indicate the use of default attributes, thereby reducing the length of the set-up message.

8	7	6	5	4	3	2	1	Octets
1	1	0	0	0	0	0	0	1
Call Class								2

Figure 7.10:

Call class (octet 2)								
	Bits							
8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	External call
0	0	0	0	0	0	0	1	Intercom call
0	0	0	0	0	0	1	0	Service call
0	0	0	0	0	0	1	1	Emergency call
	All other values reserved							

Table 7.7: Call class information element coding

Note

- An external call is a call to/from an external network; e.g. the PSTN.
- An intercom call is a call between Bluetooth devices.
- A service call is a call for configuration purposes.
- An emergency call is an external call using a dedicated emergency call number, using specific properties.

7.4.5 Called party number

The purpose of the Called party number information element is to identify the called party of a call.

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	1	0	1	1
Length of contents of information element (octets)								2
0	Type of number			Numbering plan identification				3
0	Number digits (Bluetooth character set) (Note)							4 etc.

Note – The number digits appear in multiple octet 4's in the same order in which they would be entered, that is, the number digit which would be entered first is located in the first octet 4.

Figure 7.11:

Type of number (octet 3)

Bits			
7	6	5	
0	0	0	Unknown
0	0	1	International number
0	1	0	National number
0	1	1	Network specific number
1	0	0	Subscriber number
1	1	0	Abbreviated number
1	1	1	Reserved for extension

All other values are reserved

Numbering plan identification (octet 3)

Bits				
4	3	2	1	
0	0	0	0	Unknown
0	0	0	1	ISDN/telephony numbering plan E.164
0	0	1	1	Data numbering plan Rec. X.121
0	1	0	0	Reserved
1	0	0	0	National standard numbering plan
1	0	0	1	Private numbering plan

All other values are reserved

Table 7.8: Called party information element coding

7.4.6 Calling party number

The purpose of the Calling party number information element is to identify the origin of a call.

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	1	0	0	1
Length of contents of information element (octets)								2
0	Type of number			Numbering plan identification				3
0	Presentation indicator		0	0	0	Screening indicator		4
0	Number digits (IA5 characters)							5 etc.

Figure 7.12:

Type of number (octet 3)

Bits			
7	6	5	
0	0	0	Unknown
0	0	1	International number
0	1	0	National number
0	1	1	Network specific number
1	0	0	Subscriber number
1	1	0	Abbreviated number
1	1	1	Reserved for extension

All other values are reserved

Numbering plan identification (octet 3)

Bits				
4	3	2	1	
0	0	0	0	Unknown
0	0	0	1	ISDN/telephony numbering plan E.164
0	0	1	1	Data numbering plan Rec. X.121
0	1	0	0	Reserved
1	0	0	0	National standard numbering plan
1	0	0	1	Private numbering plan

All other values are reserved

Presentation indicator (octet 4)

Bits		
7	6	
0	0	Presentation allowed
0	1	Presentation restricted
1	0	Number not available due to interworking
1	1	Reserved

All other values are reserved

Table 7.9: Calling party information element coding

Screening indicator (octet 4)	
Bits	
2 1	
0 0	User-provided, not screened
0 1	User-provided, verified and passed
1 0	User-provided, verified and failed
1 1	Network provided
All other values are reserved	

Table 7.9: Calling party information element coding

7.4.7 Cause

The purpose of the Cause is to indicate the remote side of the cause of the failure of the requested service.

8	7	6	5	4	3	2	1	Octets
1	1	0	0	0	0	0	1	1
Cause value								2

Figure 7.13:

Cause (octet 2)	
Bits	
8 7 6 5 4 3 2 1	
0	These 7 bits are coded alike the Cause value subfield defined in Section 2.2.5 of ITU-T Recommendation Q.850[2].

Table 7.10: Cause information element coding

7.4.8 Clock offset

The purpose of the Clock offset information element is to indicate the Bluetooth clock offset used.

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	0	0	0	1
Length of contents of information element (octets)								2
Clock offset								3
								4

Figure 7.14:

Clock offset coding (octet 3 and 4)															
	Bits (octet 3)								Bits (octet 4)						
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
0	Contains bits 16-2 of Bluetooth clock														

Table 7.11: Clock offset information element coding

7.4.9 Company specific

The purpose of the Company specific information element is to send non-standardized information.

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	1	1	1	1
Length of contents of information element (octets)								2
Company Identification								3
Company Identification								4
Company specific contents								
								L+2

Figure 7.15:

Company identification coding (octet 3 and octet 4)																
	Bits (octet 3)								Bits (octet 4)							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ericsson
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Nokia Mobile Phones
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	Intel Corporation
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	IBM Corporation
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Toshiba Corporation
	All other values are reserved															

Table 7.12: Company specific information element coding

7.4.10 Configuration data

The purpose of the Configuration data information element is to indicate the Configuration data.

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	0	0	1	1
Length of contents of information element (octets)								2
0	Internal number of WUG member 1 (IA5 characters)							3
0	Internal number of WUG member 1 (IA5 characters)							4
Bluetooth address of WUG member 1								5
								…
								10
Link key to be used towards WUG member 1								11
								..
								26
……								
0	Internal number of WUG member n (IA5 character)							**3+((n-1)*24)**
0	Internal number of WUG member n (IA5 character)							**4+((n-1)*24)**
Bluetooth address of WUG member n								**5+((n-1)*24)**
								…
								10+((n-1)*24)
Link key to be used towards WUG member n								**11+((n-1)*24)**
								..
								2+(n*24)

Note – The internal number (2 digits) appears in octets 3 and 4 in the same order in which they would be entered; that is, the number digit which would be entered first is located in octet 3.

Note – The octets 3-26 are repeated for all *n* WUG members.

Figure 7.16:

7.4.11 Destination CID

The purpose of the Destination CID information element is to enable the remote side to associate the established L2CAP channel with the ongoing call. The Destination CID is identical to the Destination CID (DCID) exchanged in the Configuration Request packet (see L2CAP, Section 5.4 on page 288).

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	0	1	1	1
Length of contents of information element (octets)								2
DCID byte 1								3
DCID byte 0								4

Figure 7.17:

7.4.12 Keypad facility

The purpose of the Keypad facility information element is to convey Bluetooth character sets; e.g. entered by means of a terminal keypad.

8	7	6	5	4	3	2	1	Octets
1	1	0	0	0	1	0	0	1
0	Keypad facility information (Bluetooth character set)							2

Figure 7.18:

7.4.13 Progress indicator

The purpose of the Progress indicator information element is to describe an event that has occurred during the life of a call.

8	7	6	5	4	3	2	1	Octets
1	1	0	0	0	0	1	0	1
0	Progress description							2

Figure 7.19:

Progress information (octet 2)	
Bits 7 6 5 4 3 2 1	
0 0 0 1 0 0 0	In-band information or appropriate pattern is now available
All other values reserved	

Table 7.13: Progress indicator information element coding

7.4.14 SCO Handle

The purpose of the SCO handle information element is to enable the remote side to associate the established SCO link with the ongoing call. The SCO handle is identical to the SCO handle exchanged in the LMP_SCO_link_req sent by the piconet master (see LMP, Section 3.21 on page 226).

8	7	6	5	4	3	2	1	Octets
1	1	0	0	0	1	0	1	1
SCO handle value								2

Figure 7.20:

7.4.15 Sending complete

The purpose of the Sending complete information element is to optionally indicate completion of called party number.

8	7	6	5	4	3	2	1	Octet
1	0	1	0	0	0	0	1	1

Figure 7.21:

7.4.16 Signal

The purpose of the Signal information element is to convey information to a user regarding tones and alerting signals.

8	7	6	5	4	3	2	1	Octets
1	1	0	0	0	0	1	1	1
Signal value								2

Figure 7.22:

Signal value (octet 2)								
	Bits							
8	7	6	5	4	3	2	1	
0	1	0	0	0	0	0	0	External call
0	1	0	0	0	0	0	1	Internal call
0	1	0	0	0	0	1	0	Call back
0	X	X	X	X	X	X	X	Reserved for Bluetooth standardization
1	X	X	X	X	X	X	X	Company specific

Table 7.14: Signal information element coding

8 MESSAGE ERROR HANDLING[1]

8.1 PROTOCOL DISCRIMINATION ERROR

When a message is received with a protocol discriminator coded other than the ones defined in Section 7.2 on page 486, that message shall be ignored.

8.2 MESSAGE TOO SHORT OR UNRECOGNIZED

When a message is received that is too short to contain a complete message type information element, that message shall be ignored.

When a message is received that contains a complete message type information element, but with a value which is not recognized as a defined message type, that message shall be ignored.

8.3 MESSAGE TYPE OR MESSAGE SEQUENCE ERRORS

Whenever an unexpected message, except RELEASE or RELEASE COMPLETE message is received in any state other than the Null state, that message shall be ignored.

When an unexpected RELEASE message is received, the receiving side shall disconnect and release the bearer channel if established, return a RELEASE COMPLETE message, stop all timers, and enter the Null state.

When an unexpected RELEASE COMPLETE message is received, the receiving side shall disconnect and release the bearer channel if established, stop all timers, and enter the Null state.

8.4 INFORMATION ELEMENT ERRORS

The information elements in a message shall appear (if present for information elements indicated as optional) in the exact order as indicated in Section 6.

When a message is received which misses a mandatory information element, or which contains a mandatory information element with invalid content, the message shall be ignored.

In case the error occurred with a mandatory information element in a SETUP message, a RELEASE COMPLETE message shall be returned, either with cause #96, *mandatory information element is missing*, or with cause #100, *invalid information element contents*.

1. In this section, when it is stated to ignore a certain message or part of a message (information element), this shall be interpreted as to do nothing – as if the (part of the) message had never been received.

When a message is received which has an unrecognized information element, or has an optional information element with an invalid content, or has a recognized information element not defined to be contained in that message, the receiving side shall ignore the information element.

Information elements with a length exceeding the maximum length (as given in Section 7 on page 485) shall be treated as an information element with invalid content.

9 PROTOCOL PARAMETERS

9.1 PROTOCOL TIMERS

Timer name	Value
T301	Minimum 3 minutes
T302	15 seconds
T303	20 seconds
T304	30 seconds
T305	30 seconds
T308	4 seconds
T310	30 –120 seconds
T313	4 seconds
T401	8 second
T402	8 seconds
T403	4 second
T404	2.5 seconds
T405	2 seconds
T406	20 seconds

Table 9.1: Timer values

10 BLUETOOTH STANDARD CHARACTER SET

The Bluetooth character set is based on the first 128 characters of ITU-T Recommendation T.50, except for the first 32 (control) characters, which are redefined as Bluetooth "control codes".

Character codes 00 Hex to 1F Hex are specific to the Bluetooth character set. They are not used in the standard IA5 sense. The following values are defined:

Code (Hex)	*Control character*
05	Dialling pause[Note 1)]
16	Register recall
All other values are reserved	

Table 10.1: Bluetooth standard character set

Note 1: The duration of the dialling pause is determined by the side receiving the pause character.

11 REFERENCES

[1] Q.931, “Digital Subscriber Signalling System No. 1(DSS 1) – ISDN User-Network interface Layer 3 Specification for Basic Call Control”, 03/93

[2] Q.850, “Digital Subscriber Signalling System No. 1 General – Usage of cause of location in the Digital Subscriber Signalling system No. 1 and the signalling system No. 7 ISDN User Part”, 03/93

12 LIST OF FIGURES

Figure 1.1: TCS within the Bluetooth stack 437
Figure 1.2: Point-to-point signalling in a single-point configuration 438
Figure 1.3: Signalling in a multi-point configuration 438
Figure 1.4: TCS Architecture 439
Figure 2.1: Call establishment message flow 447
Figure 2.2: Call clearing message flow 450
Figure 3.1: Obtain access rights message flow 453
Figure 3.2: Configuration distribution message flow 455
Figure 3.3: Fast inter-member access message flow 457
Figure 4.1: Connectionless TCS message flow 458
Figure 5.1: Calling line identity message flow 459
Figure 5.2: DTMF start & stop message flow 460
Figure 7.1: Protocol discriminator 475
Figure 7.2: Message type 475
Figure 7.3: Single octet information element format 477
Figure 7.4: Double octet information element format 477
Figure 7.5: Variable length information element format 477
Figure 7.6: 479
Figure 7.7: 479
Figure 7.8: 480
Figure 7.9: 480
Figure 7.10: 482
Figure 7.11: 483
Figure 7.12: 484
Figure 7.13: 485
Figure 7.14: 485
Figure 7.15: 486
Figure 7.16: 487
Figure 7.17: 488
Figure 7.18: 488
Figure 7.19: 488
Figure 7.20: 489
Figure 7.21: 489
Figure 7.22: 489
Figure A: Full TCS State Diagram 497
Figure B: Lean TCS State Diagram 498

13 LIST OF TABLES

Table 6.1: ALERTING message content....463
Table 6.2: CALL PROCEEDING message content....463
Table 6.3: CONNECT message content....464
Table 6.4: CONNECT ACKNOWLEDGE message content....464
Table 6.5: DISCONNECT message content....465
Table 6.6: INFORMATION message content....465
Table 6.7: PROGRESS message content....466
Table 6.8: RELEASE message content....466
Table 6.9: RELEASE COMPLETE message content....467
Table 6.10: SETUP message content....467
Table 6.11: SETUP ACKNOWLEDGE message content....468
Table 6.12: Start DTMF message content....468
Table 6.13: Start DTMF Acknowledge message content....469
Table 6.14: Start DTMF Reject message content....469
Table 6.15: Stop DTMF message content....469
Table 6.16: Stop DTMF Acknowledge message content....470
Table 6.17: ACCESS RIGHTS REQUEST message content....470
Table 6.18: ACCESS RIGHTS ACCEPT message content....470
Table 6.19: ACCESS RIGHTS REJECT message content....471
Table 6.20: INFO SUGGEST message content....471
Table 6.21: INFO ACCEPT message content....471
Table 6.22: LISTEN REQUEST message content....472
Table 6.23: LISTEN SUGGEST message content....472
Table 6.24: LISTEN ACCEPT message content....472
Table 6.25: LISTEN REJECT message content....473
Table 6.26: CL INFO message content....473
Table 7.1: General message format....474
Table 7.2: Protocol discriminator....475
Table 7.3: Message type....475
Table 7.4: Information element identifier coding....477
Table 7.5: Audio Control information element coding....479
Table 7.6: Bearer capability information element coding....481
Table 7.7: Call class information element coding....482
Table 7.8: Called party information element coding....483
Table 7.9: Calling party information element coding....484
Table 7.10: Cause information element coding....485
Table 7.11: Clock offset information element coding....486
Table 7.12: Company specific information element coding....486
Table 7.13: Progress indicator information element coding....488
Table 7.14: Signal information element coding....489
Table 9.1: Timer values....492
Table 10.1: Bluetooth standard character set....493

APPENDIX 1 - TCS CALL STATES

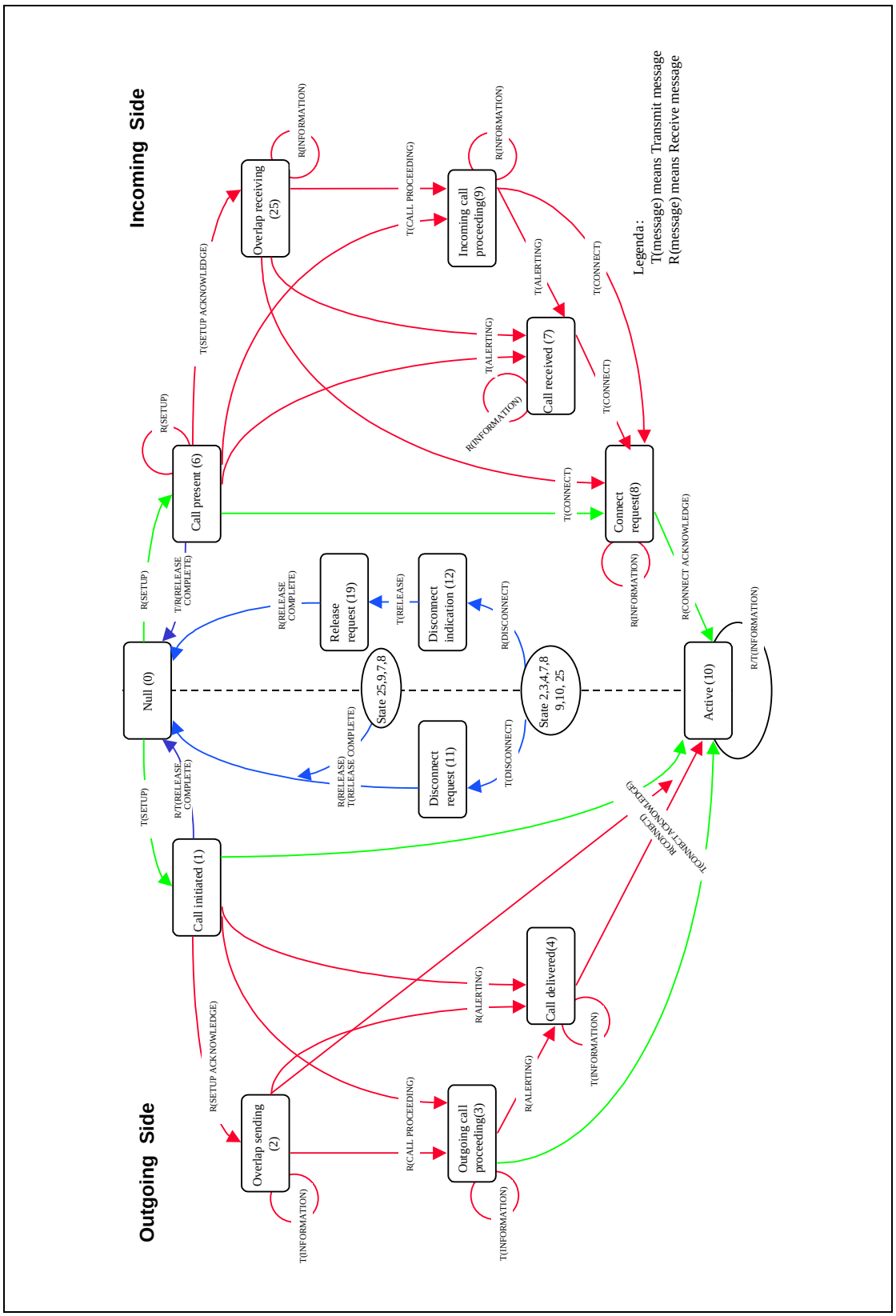


Figure A: Full TCS State Diagram

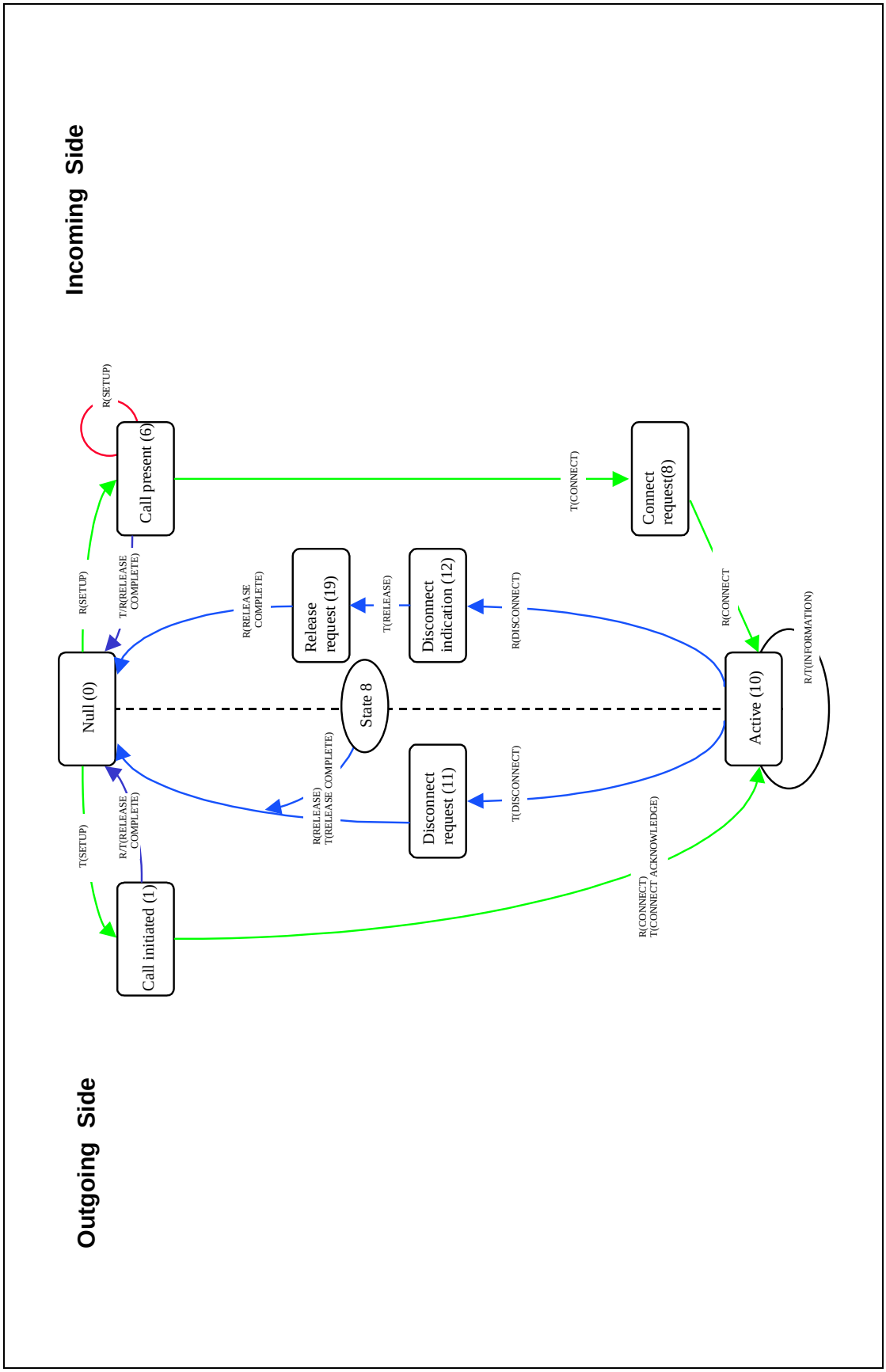


Figure B: Lean TCS State Diagram

Part F:4

INTEROPERABILITY REQUIREMENTS FOR BLUETOOTH AS A WAP BEARER

PPP Adaptation

Many of the characteristics of Bluetooth devices are shared with the target platforms for the Wireless Application Protocol. In some cases, the same device may be enabled for both types of communication. This document describes the interoperability requirements for using Bluetooth with PPP as the communications bearer for WAP protocols and applications.

CONTENTS

1 Introduction 517
1.1 Document Scope 517
2 The Use of WAP In the Bluetooth Environment 518
2.1 Value-added Services 518
2.2 Usage Cases 518
2.2.1 Briefcase Trick 518
2.2.2 Forbidden Message 519
2.2.3 WAP Smart Kiosk 519
3 WAP Services Overview 520
3.1 WAP Entities 520
3.1.1 WAP Client 520
3.1.2 WAP Proxy/Gateway 521
3.1.3 WAP Server 521
3.2 WAP Protocols 521
3.2.1 Wireless Datagram Protocol (WDP) 522
3.2.2 Wireless Transaction Protocol (WTP) 522
3.2.3 Wireless Transport Layer Security (WTLS) 522
3.2.4 Wireless Session Protocol (WSP) 522
3.3 Contrasting WAP and Internet Protocols 522
3.3.1 UDP/WDP 522
3.3.2 WTP/TCP 523
3.3.3 WTLS/SSL 523
3.3.4 WSP/HTTP 523
3.3.5 WML/HTML 523
3.3.6 WMLScript/JavaScript 523
4 WAP in the Bluetooth Piconet 524
4.1 WAP Server Communications 524
4.1.1 Initiation by the Client Device 524
4.1.1.1 Discovery of Services 525
4.1.2 Termination by the Client Device 525
4.1.3 Initiation by the Server Device 525
4.1.3.1 Discovery of Services 526
4.2 Implementation of WAP for Bluetooth 526
4.2.1 WDP Management Entity 526
4.2.1.1 Asynchronous Notifications 526
4.2.1.2 Alternate Bearers 526
4.2.2 Addressing 527
4.3 Network Support for WAP 527
4.3.1 PPP/RFCOMM 527

5 Interoperability Requirements 529
5.1 Stage 1 – Basic Interoperability 529
5.2 Stage 2 – Advanced Interoperability 529
6 Service Discovery 530
6.1 SDP Service Records 530
6.2 SDP Protocol Data Units 533
6.3 Service Discovery Procedure 533
6.4 Device Discovery 533
7 Link Manager 534
8 Generic Modes 535
9 References 536

1 INTRODUCTION

1.1 DOCUMENT SCOPE

This document is intended for Bluetooth implementers who wish to take advantage of the dynamic, ad-hoc characteristics of the Bluetooth environment in providing access to value-added services using the WAP environment and protocols.

Bluetooth provides the physical medium and link control for communications between WAP client and server. This document describes how PPP may be used to achieve this communication.

The information contained in this document is not sufficient to allow the implementation of a general-purpose WAP client or server device. Instead, this document provides the following information:

- An overview of the use of WAP in the Bluetooth environment will explain how the concept of value-added services fits within the Bluetooth vision. Examples are given of how the WAP value-added services model can be used to fulfil specific Bluetooth usage models.
- The WAP Services Overview attempts to place the WAP environment in a familiar context. Each component of WAP is introduced, and is contrasted with equivalent Internet protocols (where applicable).
- A discussion of WAP in the Bluetooth Piconet describes how the particular structure of Bluetooth communications relates to WAP behaviors.
- Finally, the Interoperability Requirements describe the specific Bluetooth features that must be implemented in order to ensure interoperability between any two WAP enabled Bluetooth devices.

2 THE USE OF WAP IN THE BLUETOOTH ENVIRONMENT

2.1 VALUE-ADDED SERVICES

The presence of communications capabilities in a device is unlikely to be an end in itself. The end users are generally not as interested in the technology as in what the technology allows them to do.

Traditional telecommunications relies on voice communications as the single application of the technology, and this approach has been successful in the marketplace. As data communications services have become more widely available, there is increasing pressure to provide services that take advantage of those data capabilities.

The Wireless Application Protocol Forum was formed to create a standards-based framework, in which value-added data services can be deployed, ensuring some degree of interoperability.

2.2 USAGE CASES

The unique quality of Bluetooth, for the purposes of delivering value-added services, is the limited range of the communications link. Devices that incorporate Bluetooth are ideally suited for the receipt of location-dependent services. The following are examples of how the WAP client / server model can be applied to Bluetooth usage cases.

2.2.1 Briefcase Trick

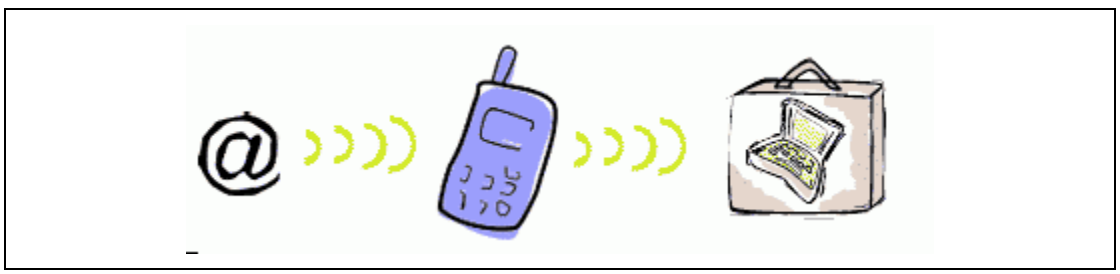


Figure 2.1: The 'Briefcase Trick' Hidden Computing Scenario

The Briefcase Trick usage case allows the user's laptop and mobile phone to communicate, without user intervention, in order to update the user's e-mail. The user can review the received messages from the handset, all without removing the laptop from its storage in a briefcase.

2.2.2 Forbidden Message

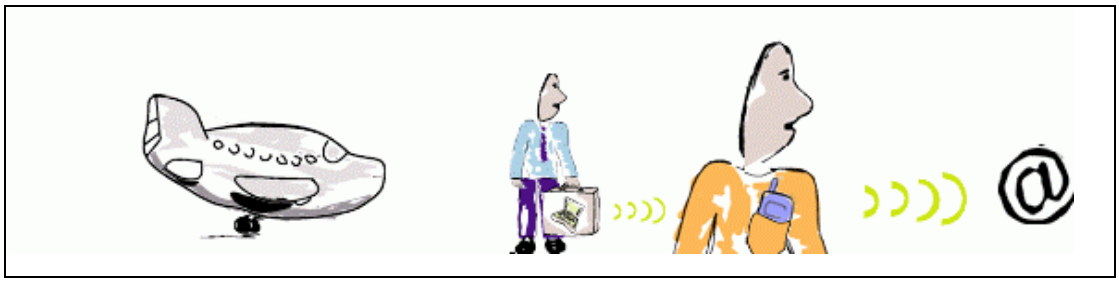


Figure 2.2: The 'Forbidden Message' Hidden Computing Scenario

The Forbidden Message usage case is similar to the briefcase trick. The user can compose messages in an environment where no dial-up connection is possible. At a later time the laptop wakes up, and checks the mobile phone to see if it is possible to send the pending messages. If the communications link is present, then the mail is transmitted.

2.2.3 WAP Smart Kiosk

The WAP Smart Kiosk usage case allows a user to connect a mobile PC or handheld device to communicate with a kiosk in a public location. The kiosk can provide information to the device that is specific to the user's location. For example, information on flights and gates in an airport, store locations in a shopping centre, or train schedules or destination information on a railway platform.

3 WAP SERVICES OVERVIEW

The Wireless Application Protocol is designed to provide Internet and Internet-like access to devices that are constrained in one or more ways. Limited communications bandwidth, memory, processing power, display capabilities and input devices are all factors driving the development of WAP. Although some devices may only exhibit some of the above constraints, WAP can still provide substantial benefit for those devices as well.

The WAP environment typically consists of three types of device: the WAP Client device, the WAP Proxy/gateway and WAP Server. In some cases the WAP Proxy/gateway may also include the server functionality.

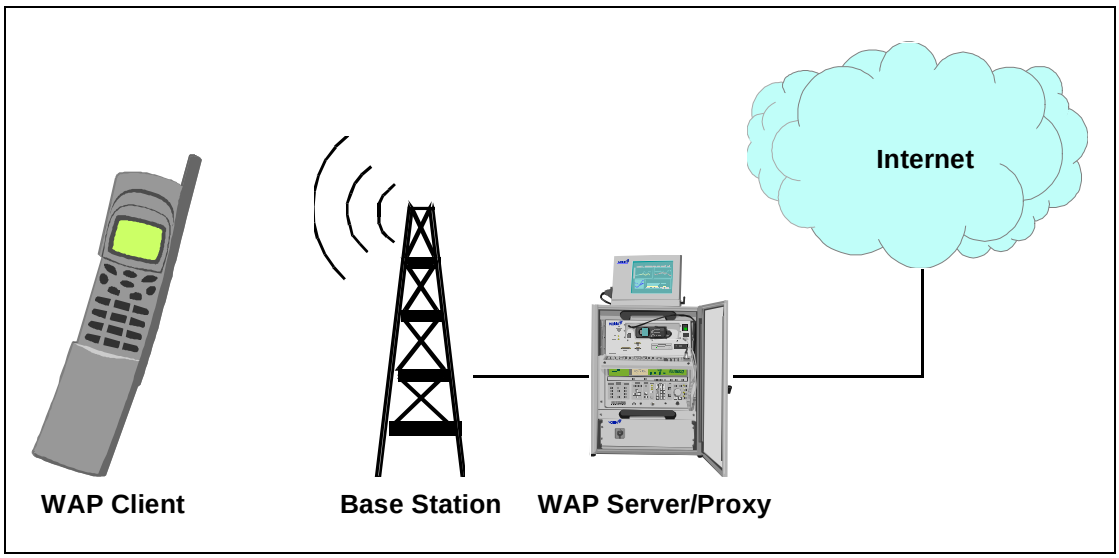


Figure 3.1: Typical WAP Environment

3.1 WAP ENTITIES

3.1.1 WAP Client

The WAP Client device is usually found in the hands of the end user. This device can be as powerful as a portable computer, or as compact as a mobile phone. The essential feature of the client is the presence of some type of display and some type of input device.

The WAP Client is typically connected to a WAP Proxy/gateway through a wireless network. (Figure 3.2 on page 519) This network may be based on any available technology. The WAP protocols allow the network to exhibit low reliability and high latency without interruption in service.

3.1.2 WAP Proxy/Gateway

The WAP Proxy/gateway acts as an interface between the wireless network, and the larger Internet. The primary functions of the proxy are to provide DNS name resolution services to WAP client devices and translation of Internet protocols and content formats to their WAP equivalents.

3.1.3 WAP Server

The WAP Server performs a function that is similar to a server in the Internet world. In fact, the WAP server is often an HTTP server. The server exists as a storage location for information that the user can access. This 'content' may include text, graphics, and even scripts that allow the client device to perform processing on behalf of the server.

The WAP Server logic may exist on the same physical device as the Proxy/gateway, or it may reside anywhere in the network that is reachable from the Proxy/gateway.

The server may fill the role of an HTTP server, a WSP server, or both.

3.2 WAP PROTOCOLS

The WAP environment consists of a layered protocol stack that is used to isolate the user agents from the details of the communications network. Figure 4.1 on page 522 illustrates the general architecture of the WAP protocol stack. Bluetooth will provide an additional data bearer service, appearing at the bottom of this diagram.

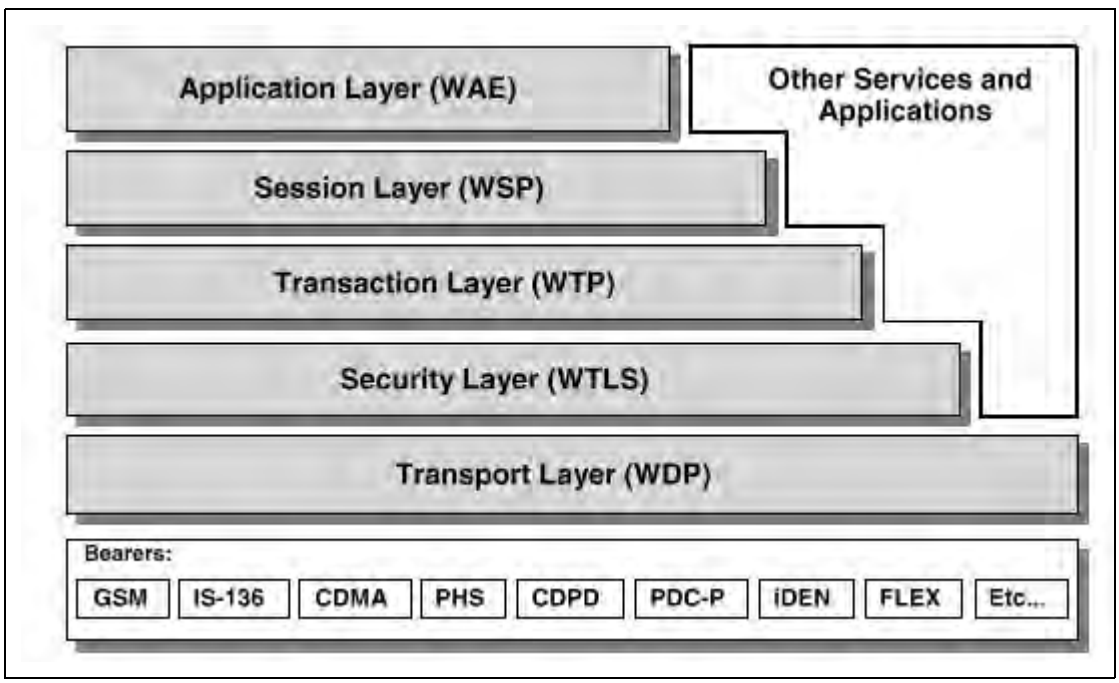


Figure 3.2: WAP Protocol Stack

3.2.1 Wireless Datagram Protocol (WDP)

The WDP layer provides a service interface that behaves as a socket-based UDP implementation. For a bearer service based on IP, then this layer is UDP. For bearer which do not provide a UDP service interface, then an implementation of WDP must be provided to act as an adaptation layer to allow socket-based UDP datagrams over the native bearer.

3.2.2 Wireless Transaction Protocol (WTP)

The WTP layer provides a reliable datagram service on top of the WDP (UDP) layer below.

3.2.3 Wireless Transport Layer Security (WTLS)

The WTLS layer is an optional component of the protocol stack that provides a secure data pipe between a client WSP session and its peer server WSP session. In the current version of the WAP specification, this session will terminate at the WAP server. There is currently a proposal before the WAP Forum for a proxy protocol, which will allow the intermediate WAP proxy to pass WTLS traffic across the proxy/gateway without decrypting the data stream.

3.2.4 Wireless Session Protocol (WSP)

The WSP layer establishes a relationship between the client application, and the WAP server. This session is relatively long-lived and able to survive service interruptions. The WSP uses the services of the WTP for reliable transport to the destination proxy/gateway.

3.3 CONTRASTING WAP AND INTERNET PROTOCOLS

The intent and implementation of the WAP protocol stack has many parallels with those of the Internet Engineering Task Force (IETF). The primary objective of the WAP Forum has been to make Internet content available to devices that are constrained in ways that make Internet protocols unsuitable for deployment.

This section compares the roles of the WAP protocol stack's layers with those of the IETF.

3.3.1 UDP/WDP

At the most basic layer, WAP and Internet protocols are the same. The WAP stack uses the model of a socket-based datagram (UDP) service as its transport interface.

Some Internet protocols also use the UDP service, but most actually use a connection-oriented stream protocol (TCP).

3.3.2 WTP/TCP

The wireless transport protocol (WTP) provides services that, in some respects, fill the same requirements as TCP. The Internet Transmission Control Protocol (TCP) provides a reliable, connection-oriented, character-stream protocol that is based on IP services. In contrast, WTP provides both reliable and unreliable, one-way and reliable two-way message transports. The transport is optimized for WAP's 'short request, long response' dialogue characteristic. WTP also provides message concatenation to reduce the number of messages transferred.

3.3.3 WTLS/SSL

The Wireless Transport Layer Security (WTLS) is derived from the Secure Sockets Layer (SSL) specification. As such, it performs the same authentication and encryption services as SSL.

3.3.4 WSP/HTTP

Session services in WAP are provided by the Wireless Session Protocol (WSP). This protocol incorporates the semantics and functionality of HTTP 1.1, while adding support for long-lived sessions, data push, suspend and resume. Additionally, the protocol uses compact encoding methods to adapt to narrow-band communications channels.

3.3.5 WML/HTML

The markup language used by WAP is a compact implementation that is similar to HTML, but optimized for use in hand-held devices. WML is an XML-defined markup language.

3.3.6 WMLScript/JavaScript

WAP also incorporates a scripting language that is similar to JavaScript, but adapted to the types of constrained devices that WAP is targeted for.

4 WAP IN THE BLUETOOTH PICONET

In many ways, Bluetooth can be used like other wireless networks with regard to WAP. Bluetooth can be used to provide a bearer for transporting data between the WAP Client and its adjacent WAP Server.

Additionally, Bluetooth's *ad hoc* nature provides capabilities that are exploited uniquely by the WAP protocols.

4.1 WAP SERVER COMMUNICATIONS

The traditional form of WAP communications involves a client device that communicates with a Server/Proxy device using the WAP protocols. In this case the Bluetooth medium is expected to provide a bearer service as specified by the WAP architecture.

4.1.1 Initiation by the Client Device

When a WAP client is actively 'listening' for available Bluetooth devices, it can discover the presence of a WAP server using Bluetooth's Service Discovery Protocol.

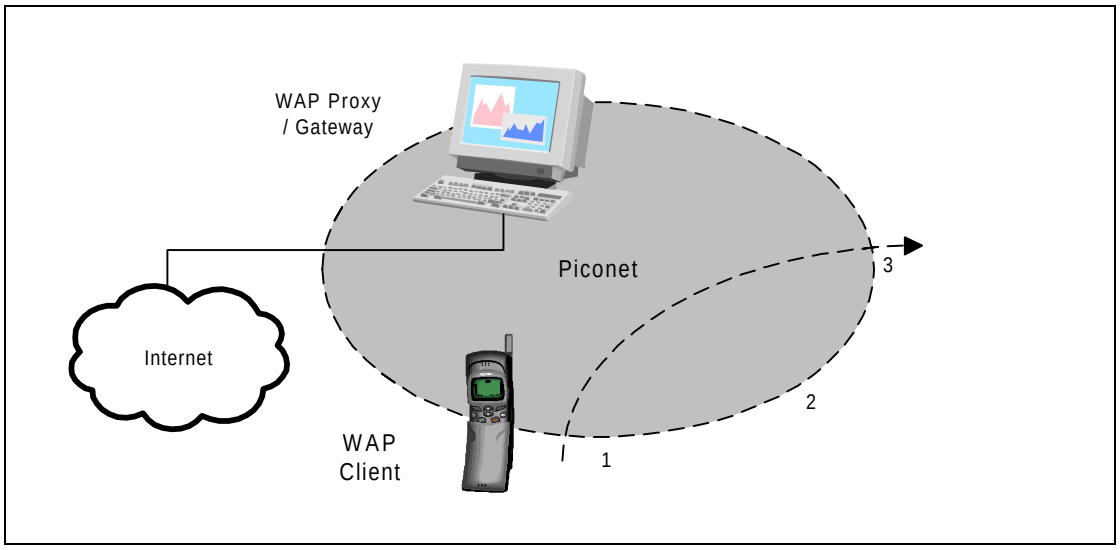


Figure 4.1: WAP Server / Proxy in Piconet

In Figure 4.1, stage 1 the WAP Client device is moving into range of the WAP Proxy/gateway's piconet. When the client detects the presence of the WAP proxy/gateway, it can automatically, or at the client's request, connect to the server.

4.1.1.1 Discovery of Services

The client must be able to determine the specific nature of the WAP proxy/gateway that it has detected. It is expected that the Bluetooth Service Discovery Protocol will be used to learn the following information about the server:

- Server Name – this is a user readable descriptive name for the server.
- Server Home Page Document Name – this is the home page URL for the server.
- Server/Proxy Capability – indicates if the device is a WAP content server or a Proxy.If the device is a Proxy, it must be able to resolve URLs that are not local to the Server/Proxy device.

In Figure 4.1, stage 2, the device is communicating with the WAP proxy/gateway. All WAP data services normally available are possible.

4.1.2 Termination by the Client Device

In Figure 4.1, stage 3, the device is exiting the piconet. When the device detects that communication has been lost with the WAP proxy/gateway, it may optionally decide to resume communications using the information obtained at discovery.

For example, a client device that supports alternate bearers may query the alternate address information of the server when that capability is indicated. The information should be cached for later access because the client device may leave the piconet at any time, and that information will no longer be available.

In the WAP Smart Kiosk example above, if the user wishes to continue receiving information while out of Bluetooth range, the Kiosk would provide an Internet address to the client device. When Bluetooth communications are not possible, the device could use cellular packet data to resume the client-server session.

This capability is implementation-dependent, and is provided here for illustrative purposes only.

4.1.3 Initiation by the Server Device

An alternative method of initiating communications between a client and server is for the server to periodically check for available client devices. When the server device discovers a client that indicates that it has WAP Client capability, the server may optionally connect and push data to the client.

The client device has the option of ignoring pushed data at the end user's discretion.

4.1.3.1 Discovery of Services

Through the Bluetooth Service Discovery Protocol, the server can determine the following information about the client:

- Client Name – this is a friendly format name that describes the client device
- Client capabilities – this information allows the server to determine basic information regarding the client's Bluetooth-specific capabilities

4.2 IMPLEMENTATION OF WAP FOR BLUETOOTH

In order to effectively implement support for WAP over Bluetooth, certain capabilities must be considered.

4.2.1 WDP Management Entity

Associated with an instance of the WDP layer in the WAP Protocol Stack is an entity that is responsible for managing the services provided by that layer. The WDP Management Entity (WDP-ME) acts as an out-of-band mechanism for controlling the protocol stack.

4.2.1.1 Asynchronous Notifications

The WDP-ME will need to be able to generate asynchronous notifications to the application layer when certain events occur. Example notifications are:

- New Client Node Detected
- New Server Node Detected
- Client Node Signal Lost
- Server Node Signal Lost
- Server Push Detected (detected as unsolicited content)

Platform support for these events is implementation-specific. All of the listed events may be derived through the Bluetooth Host Controller Interface (page 535), with the exception of Server Push.

4.2.1.2 Alternate Bearers

An implementation of WAP on a particular device may choose to support multiple bearers. Methods of performing bearer selection are beyond the scope of this document. The procedure to be followed is implementation-dependent. See Section 4.1.2 above.

4.2.2 Addressing

Two basic types of addressing are being used in the WAP environment: User Addressing and Proxy/gateway Addressing. User addressing describes the location of objects within the network, and is independent of the underlying bearer. Proxy/Gateway Addressing describes the location of the WAP proxy/gateway that the device is communicating with. Proxy/Gateway addressing is dependent on the bearer type.

The end user deals mainly with Uniform Resource Locators (URL). These addresses are text strings that describe the document that is being accessed. Typically, the Proxy/gateway in conjunction with Internet Domain Name.

Servers resolve these strings into network addresses.

The address of the WAP Proxy/gateway is usually a static value that is configured by the user or network operator. When the user enters a URL, the request is forwarded to the configured WAP proxy/gateway. If the URL is within the domain of a co-located server, then it indicates that the document is actually WAP content. If the URL is outside of the WAP proxy/gateway's domain, then the WAP Proxy/gateway typically uses DNS name resolution to determine the IP address of the server on which the document resides.

The client device would first identify a proxy/gateway that is reachable through Bluetooth, then it would use the service discovery protocol to present the user with a server name or description. When the user selects a server, then the WAP client downloads the home page of the server (as determined by the discovery process; see section 4.1.1.1 on page 523) Once the user has navigated to the home page of the desired server, then all subsequent URLs are relative to this home page. This scenario presumes that the WAP Proxy/gateway and WAP Content server are all co-located in the Bluetooth device, although this structure is not required for interoperability.

A WAP Proxy/gateway/Server will typically provide a default URL containing the home page content for the server. A proxy-only device typically provides no URL or associated content.

4.3 NETWORK SUPPORT FOR WAP

The following specifies a protocol stack, which may be used below the WAP components. Support for other protocol stack configurations is optional, and must be indicated through the Bluetooth Service Discovery Protocol.

4.3.1 PPP/RFCOMM

Devices that support Bluetooth as a bearer for WAP services using PPP provide the following protocol stack support:

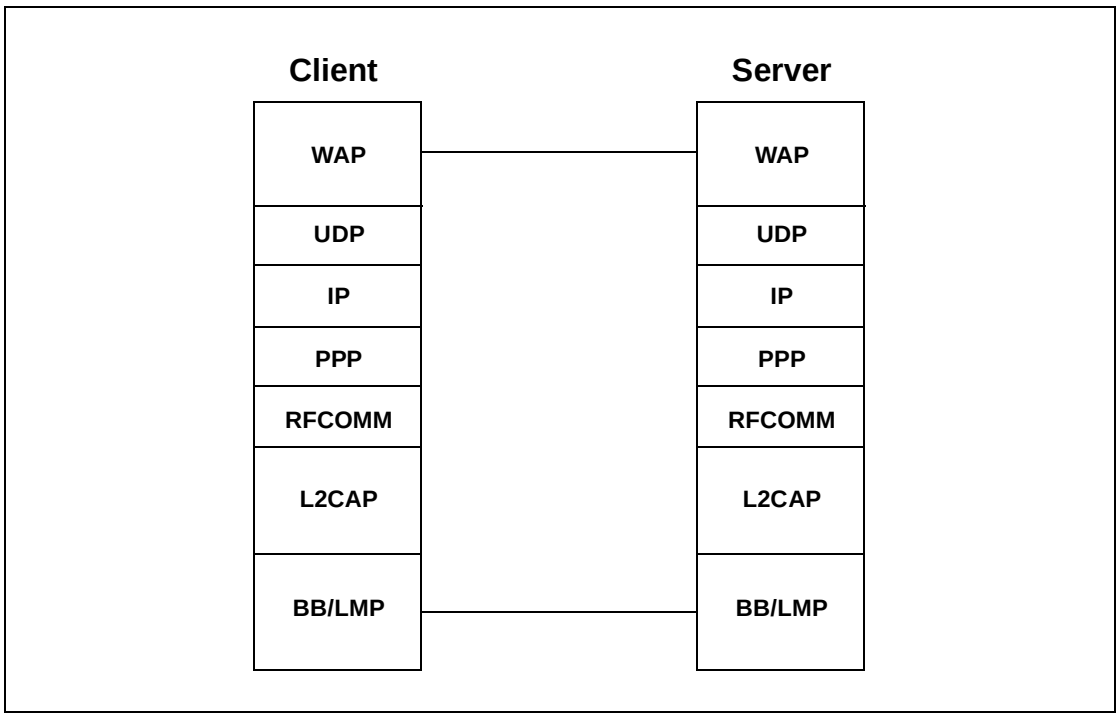


Figure 4.2: Protocol Support for WAP

For the purposes of interoperability, this document assumes that a WAP client conforms to the role of Data Terminal as defined in LAN Access Profile using PPP [6]. Additionally, the WAP server or proxy device is assumed to conform to the role of the LAN Access Point defined in [6].

The Baseband (page 33), LMP (page 183) and L2CAP (page 253) are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM (page 393) is the Bluetooth adaptation of GSM TS 07.10 [1]. SDP (page 331) is the Bluetooth Service Discovery Protocol.

PPP is the IETF Point-to-Point Protocol [3]. WAP is the Wireless Application Protocol stack and application environment [5].

5 INTEROPERABILITY REQUIREMENTS

5.1 STAGE 1 – BASIC INTEROPERABILITY

Stage 1 interoperability for WAP over Bluetooth (all mandatory):

- Provide WAP Class Cdevice compliance [7]
- Provide, through service discovery mechanisms, the network address for devices that support WAP proxy/gateway functionality.

5.2 STAGE 2 – ADVANCED INTEROPERABILITY

Stage 2 interoperability for WAP over Bluetooth (mandatory):

- All Stage 1 interoperability requirements are supported
- Provide Server Name and information about Server/Proxy capabilities through service discovery.
- Provide Client Name and information about Client Capabilities through service discovery.
- Asynchronous Notifications for Server.
- Asynchronous Notifications for Client.

6 SERVICE DISCOVERY

6.1 SDP SERVICE RECORDS

Service records are provided as a mechanism through which WAP client devices and proxy/gateways become aware of each other dynamically. This usage differs from other WAP bearers in that the relationship between the two devices will be transitory. That is, a Bluetooth device will not have a bearer-specific address configured or provisioned to a specific proxy/gateway.

Clients and proxy/gateways become aware of each other as they come in proximity of one another. The Bluetooth Service Discovery Protocol allows the devices to query the capabilities of each other as listed in the Interoperability Requirements section of this document.

Table 6.1 shows the service record for the WAP Proxy/gateway device.

Item	Definition	Type	Value	AttrID	Req
Service-ClassIDList				0x0001	M
ServiceClass0	WAP Proxy/Gate-way	UUID	WAP[4]		M
BluetoothProfile DescriptorList					M
ProfileDescrip tor0				0x0009	M
Profile	Supported Profile	UUID	LANAccess UsingPPP [4]		M
Version	Profile Version	Uint16	*(varies)*		M
Protocol DescriptorList					O
Descriptor0	UDP	UUID	UDP		O
Parameter0	WSP Connection-less Session Port No.	Uint16	9200 *(default)*		O
Parameter1	WTP Session Port No.	Uint16	9201 *(default)*		O
Parameter2	WSP Secure Connectionless Port No.	Uint16	9202 *(default)*		O
Parameter3	WTP Secure Ses-sion Port No.	Uint16	9203 *(default)*		O
Parameter4	WAP vCard Port No.	Uint16	9204 *(default)*		O

Table 6.1: Service Record format for WAP Proxy/Gateway devices

Item	Definition	Type	Value	AttrID	Req
Parameter5	WAP vCal Port No.	Uint16	9205 *(default)*		O
Parameter6	WAP vCard Secure Port No.	Uint16	9206 *(default)*		O
Parameter7	WAP vCal Secure Port No.	Uint16	9207 *(default)*		O
ServiceName	Displayable Text name	String	*(varies, e.g.* 'Airport information')	See [4]	O* M†.
NetworkAddress	IP Network Address of Server	Uint32	(varies)	See [4]	M
WAPStackType	What type of stack is supported	Uint8	0x01=Connectionless 0x02=Connection oriented 0x03=both stack types	See [4]	Mandatory if WAPGateway is set to 0x01. Otherwise optional but recommended
WAPGateway†	Indicates if device is origin server or proxy	Uint8	0x01 = Origin Server; 0x02 = Proxy; V‡	See [4]	M
HomePageURL	URL of home page document	URL	V**	See [4]	M††

Table 6.1: Service Record format for WAP Proxy/Gateway devices

*. Stage 1 interoperability requirements

†. Stage 2 interoperability requirements.

‡. An origin server is not connected to a network and only has local content. A proxy device can resolve external URLs, and it may also have local content.

**. For devices with WAPGateway set to 0x01 the HomePageURL is used by the client to:

- Identify the home page URL of the device.
- Allow the client to turn the host name part of the HomePAgeURL into an IP address (found in the NetworkAddress attribute)
- Allow the client to select the correct bearer and connect to the correctBluetooth device address for a specific URL request. By keeping a list of host name to Bluetooth deviceaddress mappings the client can select the correct ionterface for each URL request. The host names in the list are retrieved from the HomePageURL attribute. It is implementation specific whether or not to keep such a list and how to maintain it.

For devices with WAPGateway set to 0x02 the Hom,ePageURL is used by the client to:

- Identify the home page URL of the device.

††.For devices with WAPGateway set to 0x01:

- The HomePageURL is mandatory.
- The host part of the HomePageURL must correspond to the IP address given in the NetworkAddress attribute to allow clients to turn this host name into an IP address using this information.
- It is highly recommended that the HomePageURL does not change over time.
- It is highly recommended that the host part of the HomePageURL is unique distinguishing it from the host names of the other devices.

For devices with WAPGateway set to 0x02:

- The HomePageURL is mandatory

Item	Definition	Type	Value	AttrID	Req
ServiceClassIDList				0x0001	M
ServiceClass0	WAP Client	UUID	WAP_CLIENT[4]		M
BluetoothProfile DescriptorList					M
ProfileDescriptor0				0x0009	M
Profile	Supported Profile	UUID	LANAccess UsingPPP [4]		M
Version	Profile Version	Uint16	*(varies)*		M
ServiceName	Displayable Text name of client	String	*(varies)*		O

Table 6.2: Service Record format for WAP Client devices

6.2 SDP PROTOCOL DATA UNITS

Table 6.3 shows the specified SDP PDUs (Protocol Data Units), which are required for WAP Interoperability.

PDU No.	SDP PDU	Ability to Send		Ability to Retrieve	
		WAP Client	WAP Proxy	WAP Client	WAP Proxy
1	SdpErrorResponse	M	M	M	M
2	SdpServiceSearchAttributeRequest	M	O	M	M
3	SdpServiceSearchAttributeResponse	M	M	M	M

Table 6.3: SDP PDU:s

6.3 SERVICE DISCOVERY PROCEDURE

In the simplest form, the signaling can be like this:

WAP Client or Proxy		WAP Client or Proxy
	SdpServiceSearchAttributeRequest =======================>	
	SdpServiceSearchAttributeResponse <=======================	

WAP service discovery procedures are symmetrical. Each device must be able to handle all of the PDUs without regard for the current device role. A minimal implementation must return the service name string.

6.4 DEVICE DISCOVERY

For the CoD field of the FHS packet the following requirements must be met:

- Devices that have set WAPGateway to 0x01 must set COD bit 23 to indicate that it is a WAP server.
- Device that have set WAPGateway to 0x02 must set COD bit 17 to indicate that it supplies access to a network.

7 LINK MANAGER

The Link Manager requirements as defined in section 9 on page 288 of LAN Access Profile are relaxed with respect to security. For WAP over Bluetooth the following requirements occur:L

Procedure	Support in WAP Server/ Proxy/ Gateway	Support in WAP Client	Use in WAP Server/ Proxy/ Gateway	Use in WAP Client
Authentica-tion	O	O	O	O
Pairing	O	O	O	O
ENcryption	O	O	O	O

Table 7.1: Link Manager Requirements different to those for LAN Profile

8 GENERIC MODES

The following modes are defined in section 4 on page 29 of Generic Access Profile. WAP over Bluetooth requires the following support (note: this is different from LAN Access Profile Requirements).

Procedure/Mode	Support in WAP server/ proxy gateway	Support in WAP client
General Discovery Procedure	O	M
Limited Discovery Procedure	O	O
Non-discoverable mode	O	O
Limited-discoverable mode	O	O
General-discoverable mode	M	O
Initiate link establishment	O	M
Accept link establishment	M	O
Non-connectable mode	O	O
Connectable mode	M	O

Table 8.1: Generic mode requirements table

9 REFERENCES

[1] TS 101 369 (GSM 07.10) version 6.2.0

[2] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 50, RFC 1661, Daydreamer, July 1994.

[3] Simpson, W., Editor, "PPP in HDLC Framing", STD 51, RFC 1662, Daydreamer, July 1994.

[4] See , “Bluetooth Assigned Numbers” (http://www.bluetooth.org/assigned-numbers.htm)

[5] Wireless Application Protocol Forum, "Wireless Application Protocol", version 1.0, 1998

[6] Bluetooth Special Interest Group, "Bluetooth LAN Access Profile using PPP", Paul Moran, Ed., version 1.0, 1999

[7] Wireless Application Protocol Forum, "WAP Class Conformance Requirements"; Prototype 01-July-1999

Part H:1

HOST CONTROLLER INTERFACE FUNCTIONAL SPECIFICATION

This document describes the functional specification for the Host Controller Interface (HCI). The HCI provides a command interface to the baseband controller and link manager, and access to hardware status and control registers. This interface provides a uniform method of accessing the Bluetooth baseband capabilities.

CONTENTS

1 Introduction 546
1.1 Lower Layers of the Bluetooth Software Stack 546
1.2 Bluetooth Hardware Block Diagram 547
1.2.1 Link Controller 548
1.2.2 CPU Core 548
1.3 Possible Physical Bus Architectures 549
1.3.1 USB HCI Architecture 549
1.3.2 PC Card HCI Architecture 549
2 Overview of Host Controller Transport Layer 550
3 HCI Flow Control 551
4 HCI Commands 553
4.1 Introduction 553
4.2 Terminology 553
4.3 Data and Parameter Formats 554
4.4 Exchange of HCI-Specific Information 554
4.4.1 HCI Command Packet 554
4.4.2 HCI Event Packet 557
4.4.3 HCI Data Packets 558
4.5 Link Control Commands 562
4.5.1 Inquiry 564
4.5.2 Inquiry_Cancel 566
4.5.3 Periodic_Inquiry_Mode 567
4.5.4 Exit_Periodic_Inquiry_Mode 570
4.5.5 Create_Connection 571
4.5.6 Disconnect 574
4.5.7 Add_SCO_Connection 575
4.5.8 Accept_Connection_Request 577
4.5.9 Reject_Connection_Request 579
4.5.10 Link_Key_Request_Reply 580
4.5.11 Link_Key_Request_Negative_Reply 582
4.5.12 PIN_Code_Request_Reply 583
4.5.13 PIN_Code_Request_Negative_Reply 585
4.5.14 Change_Connection_Packet_Type 586
4.5.15 Authentication_Requested 589
4.5.16 Set_Connection_Encryption 590
4.5.17 Change_Connection_Link_Key 591
4.5.18 Master_Link_Key 592

4.5.19 Remote_Name_Request....593
4.5.20 Read_Remote_Supported_Features....595
4.5.21 Read_Remote_Version_Information....596
4.5.22 Read_Clock_Offset....597
4.6 Link Policy Commands....598
4.6.1 Hold_Mode....600
4.6.2 Sniff_Mode....602
4.6.3 Exit_Sniff_Mode....605
4.6.4 Park_Mode....606
4.6.5 Exit_Park_Mode....608
4.6.6 QoS_Setup....609
4.6.7 Role_Discovery....611
4.6.8 Switch_Role....612
4.6.9 Read_Link_Policy_Settings....613
4.6.10 Write_Link_Policy_Settings....615
4.7 Host Controller & Baseband Commands....617
4.7.1 Set_Event_Mask....623
4.7.2 Reset....625
4.7.3 Set_Event_Filter....626
4.7.4 Flush....632
4.7.5 Read_PIN_Type....634
4.7.6 Write_PIN_Type....635
4.7.7 Create_New_Unit_Key....636
4.7.8 Read_Stored_Link_Key....637
4.7.9 Write_Stored_Link_Key....639
4.7.10 Delete_Stored_Link_Key....641
4.7.11 Change_Local_Name....643
4.7.12 Read_Local_Name....644
4.7.13 Read_Connection_Accept_Timeout....645
4.7.14 Write_Connection_Accept_Timeout....646
4.7.15 Read_Page_Timeout....647
4.7.16 Write_Page_Timeout....648
4.7.17 Read_Scan_Enable....649
4.7.18 Write_Scan_Enable....650
4.7.19 Read_Page_Scan_Activity....651
4.7.20 Write_Page_Scan_Activity....653
4.7.21 Read_Inquiry_Scan_Activity....655
4.7.22 Write_Inquiry_Scan_Activity....657
4.7.23 Read_Authentication_Enable....659

4.7.24 Write_Authentication_Enable 660
4.7.25 Read_Encryption_Mode 661
4.7.26 Write_Encryption_Mode 662
4.7.27 Read_Class_of_Device 664
4.7.28 Write_Class_of_Device 665
4.7.29 Read_Voice_Setting 666
4.7.30 Write_Voice_Setting 668
4.7.31 Read_Automatic_Flush_Timeout 670
4.7.32 Write_Automatic_Flush_Timeout 672
4.7.33 Read_Num_Broadcast_Retransmissions 674
4.7.34 Write_Num_Broadcast_Retransmissions 675
4.7.35 Read_Hold_Mode_Activity 676
4.7.36 Write_Hold_Mode_Activity 678
4.7.37 Read_Transmit_Power_Level 679
4.7.38 Read_SCO_Flow_Control_Enable 681
4.7.39 Write_SCO_Flow_Control_Enable 682
4.7.40 Set_Host_Controller_To_Host_Flow_Control 683
4.7.41 Host_Buffer_Size 685
4.7.42 Host_Number_Of_Completed_Packets 687
4.7.43 Read_Link_Supervision_Timeout 689
4.7.44 Write_Link_Supervision_Timeout 691
4.7.45 Read_Number_Of_Supported_IAC 693
4.7.46 Read_Current_IAC_LAP 694
4.7.47 Write_Current_IAC_LAP 695
4.7.48 Read_Page_Scan_Period_Mode 697
4.7.49 Write_Page_Scan_Period_Mode 699
4.7.50 Read_Page_Scan_Mode 700
4.7.51 Write_Page_Scan_Mode 701
4.8 Informational Parameters 702
4.8.1 Read_Local_Version_Information 703
4.8.2 Read_Local_Supported_Features 705
4.8.3 Read_Buffer_Size 706
4.8.4 Read_Country_Code 708
4.8.5 Read_BD_ADDR 709
4.9 Status Parameters 710
4.9.1 Read_Failed_Contact_Counter 711
4.9.2 Reset_Failed_Contact_Counter 713
4.9.3 Get_Link_Quality 715
4.9.4 Read_RSSI 717

4.10 Testing Commands 719
4.10.1 Read_Loopback_Mode 720
4.10.2 Write_Loopback_Mode 723
4.10.3 Enable_Device_Under_Test_Mode 726

5 Events 727
5.1 Event 727
5.2 Possible Events 730
5.2.1 Inquiry Complete event 730
5.2.2 Inquiry Result event 731
5.2.3 Connection Complete event 733
5.2.4 Connection Request event 735
5.2.5 Disconnection Complete event 736
5.2.6 Authentication Complete event 737
5.2.7 Remote Name Request Complete event 738
5.2.8 Encryption Change event 739
5.2.9 Change Connection Link Key Complete event 740
5.2.10 Master Link Key Complete event 741
5.2.11 Read Remote Supported Features Complete event 742
5.2.12 Read Remote Version Information Complete event 743
5.2.13 QoS Setup Complete event 745
5.2.14 Command Complete event 747
5.2.15 Command Status event 748
5.2.16 Hardware Error event 749
5.2.17 Flush Occurred event 750
5.2.18 Role Change event 751
5.2.19 Number Of Completed Packets event 752
5.2.20 Mode Change event 754
5.2.21 Return Link Keys event 756
5.2.22 PIN Code Request event 757
5.2.23 Link Key Request event 758
5.2.24 Link Key Notification event 759
5.2.25 Loopback Command event 760
5.2.26 Data Buffer Overflow event 761
5.2.27 Max Slots Change event 762
5.2.28 Read Clock Offset Complete event 763
5.2.29 Connection Packet Type Changed event 764
5.2.30 QoS Violation event 766
5.2.31 Page Scan Mode Change event 767
5.2.32 Page Scan Repetition Mode Change event 768

6 List of Error Codes 769
6.1 List of Error Codes 769
6.2 HCI Error Code Usage Descriptions 771
6.3 Unknown HCI Command (0x01) 771
6.4 No Connection (0x02) 772
6.5 Hardware Failure (0x03) 772
6.6 Page Timeout (0x04) 772
6.7 Authentication Failure (0x05) 772
6.8 Key Missing (0x06) 773
6.9 Memory Full (0x07) 773
6.10 Connection Timeout (0x08) 773
6.11 Max Number Of Connections (0x09) 773
6.12 Max Number Of SCO Connections To A Device (0x0A) 774
6.13 ACL Connection Already Exists (0x0B) 774
6.14 Command Disallowed (0x0C) 774
6.15 Host Rejected due to … (0x0D-0x0F) 774
6.16 Host Timeout (0x10) 775
6.17 Unsupported Feature or Parameter Value (0x11) 775
6.18 Invalid HCI Command Parameters (0x12) 775
6.19 Other End Terminated Connection: … (0x13-0x15) 776
6.20 Connection Terminated By Local Host (0x16) 776
6.21 Repeated Attempts (0x17) 776
6.22 Pairing Not Allowed (0x18) 776
6.23 Unsupported Remote Feature (0x1A) 777
6.24 Unspecified error (0x1F) 777
6.25 Unsupported LMP Parameter Value (0x20) 777
6.26 Role Change Not Allowed (0x21) 777
6.27 LMP Response Timeout (0x22) 777
6.28 LMP Error Transaction Collision (0X23) 778
6.29 LMP PDU Not Allowed (0X24) 778
6.30 ENCRYPTION MODE NOT ACCEPTABLE (0X25) 778
6.31 UNIT KEY USED (0X26) 778
6.32 QoS IS NOT SUPPORTED (0X27) 778
6.33 INSTANT PASSED (0X28) 778
6.34 PAIRING WITH UNIT KEY NOT SUPPORTED (0X29) 778
7 List of Acronyms and Abbreviations 780
8 List of Figures 781
9 List of Tables 782

1 INTRODUCTION

This document describes the functional specifications for the Host Controller Interface (HCI). The HCI provides a uniform interface method of accessing the Bluetooth hardware capabilities. The next two sections provide a brief overview of the lower layers of the Bluetooth software stack and of the Bluetooth hardware. Section 2, provides an overview of the Lower HCI Device Driver Interface on the host device. Section 3, describes the flow control used between the Host and the Host Controller. Section 4, describes each of the HCI Commands in details, identifies parameters for each of the commands, and lists events associated with each command.

1.1 LOWER LAYERS OF THE BLUETOOTH SOFTWARE STACK

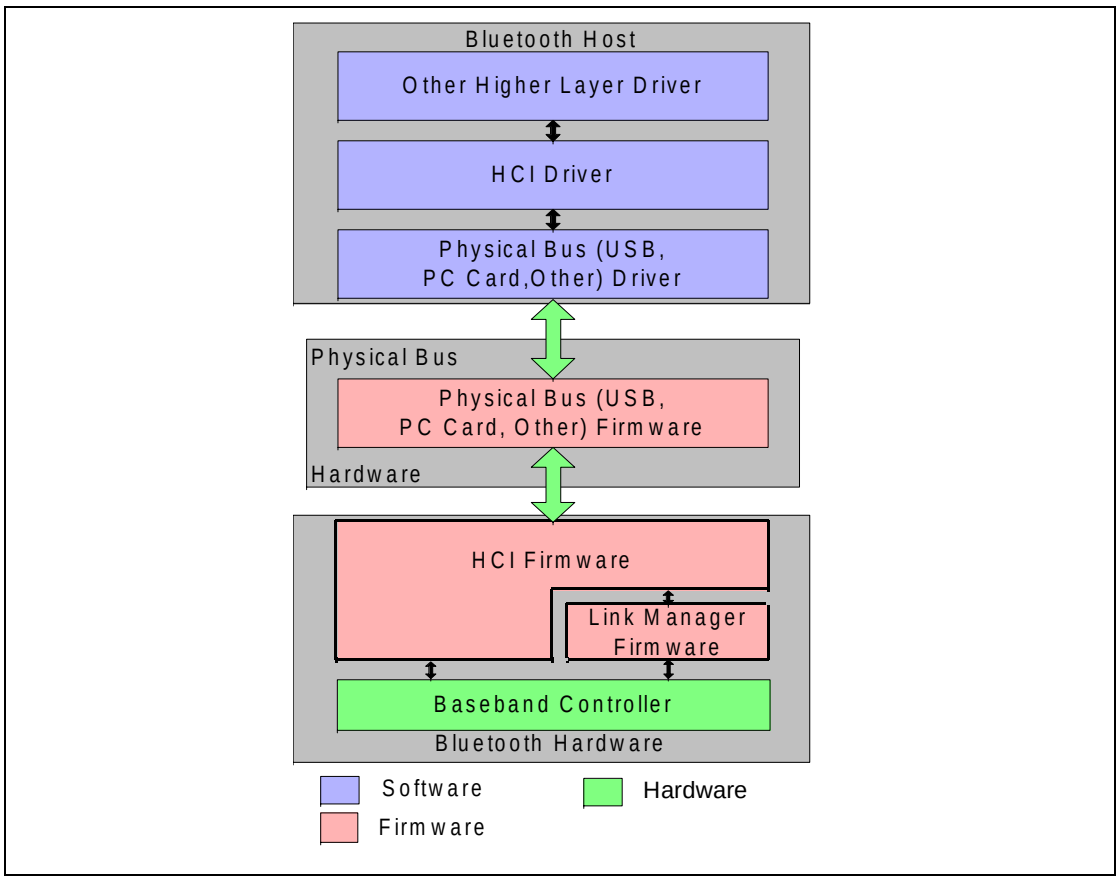


Figure 1.1: Overview of the Lower Software Layers

Figure 1.1, provides an overview of the lower software layers. The HCI firmware implements the HCI Commands for the Bluetooth hardware by accessing baseband commands link manager commands, hardware status registers, control registers, and event registers.

Several layers may exist between the HCI driver on the host system and the HCI firmware in the Bluetooth hardware. These intermediate layers, the Host

Controller Transport Layer, provide the ability to transfer data without intimate knowledge of the data.

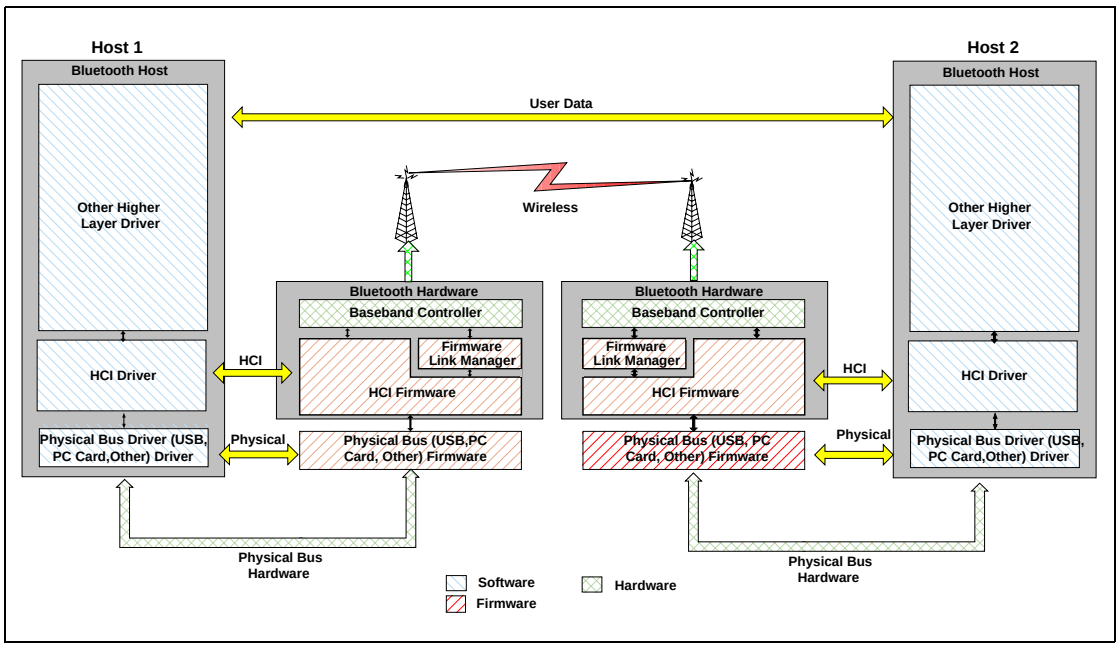


Figure 1.2: End to End Overview of Lower Software Layers to Transfer Data

Figure 1.2, illustrates the path of a data transfer from one device to another. The HCI driver on the Host exchanges data and commands with the HCI firmware on the Bluetooth hardware. The Host Control Transport Layer (i.e. physical bus) driver provides both HCI layers with the ability to exchange information with each other.

The Host will receive asynchronous notifications of HCI events independent of which Host Controller Transport Layer is used. HCI events are used for notifying the Host when something occurs. When the Host discovers that an event has occurred it will then parse the received event packet to determine which event occurred.

1.2 BLUETOOTH HARDWARE BLOCK DIAGRAM

A general overview of the Bluetooth hardware is outlined in Figure 1.3 on page 545. It consists of an analog part – the Bluetooth radio, and a digital part – the Host Controller. The Host Controller has a hardware digital signal processing part –the Link Controller (LC), a CPU core, and it interfaces to the host environment. The hardware and software parts of the Host Controller are described below.

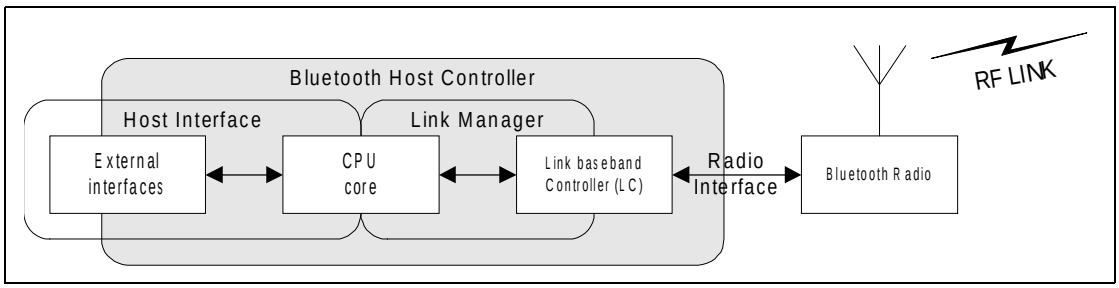


Figure 1.3: Bluetooth Hardware Architecture Overview.

1.2.1 Link Controller

The Link Controller (LC) consists of hardware and software parts that perform Bluetooth baseband processing, and physical layer protocols such as ARQ-protocol and FEC coding.

The functions performed by the Link Controller include:

- Transfer types with selected Quality-of-Service (QoS) parameters
- Asynchronous transfers with guaranteed delivery using hardware fast Automatic Repeat reQuest (fARQ). Frames can be flushed from the retransmission buffer, for use with isochronous data
- Synchronous transfers
- Audio coding. A power-efficient hardware implementation of a robust 64 Kbits/s Continuous Variable Slope Delta (CVSD) coding, as well as 64 Kbits/s log-PCM
- Encryption

1.2.2 CPU Core

The CPU core will allow the Bluetooth module to handle Inquiries and filter Page requests without involving the host device. The Host Controller can be programmed to answer certain Page messages and authenticate remote links.

The Link Manager (LM) software runs on the CPU Core. The LM discovers other remote LMs and communicates with them via the Link Manager Protocol (LMP) to perform its service provider role using the services of the underlying Link Controller (LC). For details see “Link Manager Protocol” on page 183

1.3 POSSIBLE PHYSICAL BUS ARCHITECTURES

Bluetooth devices will have various physical bus interfaces that could be used to connect to the Bluetooth hardware. These buses may have different architectures and different parameters. The Bluetooth Host Controller will initially support two physical bus architectures, USB, and PC Card.

1.3.1 USB HCI Architecture

The following block diagram shows the Bluetooth connection to the Host PC via the USB HCI. USB can handle several logic channels over the same single physical channel (via Endpoints). Therefore control, data, and voice channels do not require any additional physical interfaces. Note that there is no direct access to registers/memory on the Bluetooth module over USB. Instead, this is done by using the appropriate HCI Commands and by using the Host Controller Transport Layer interface.

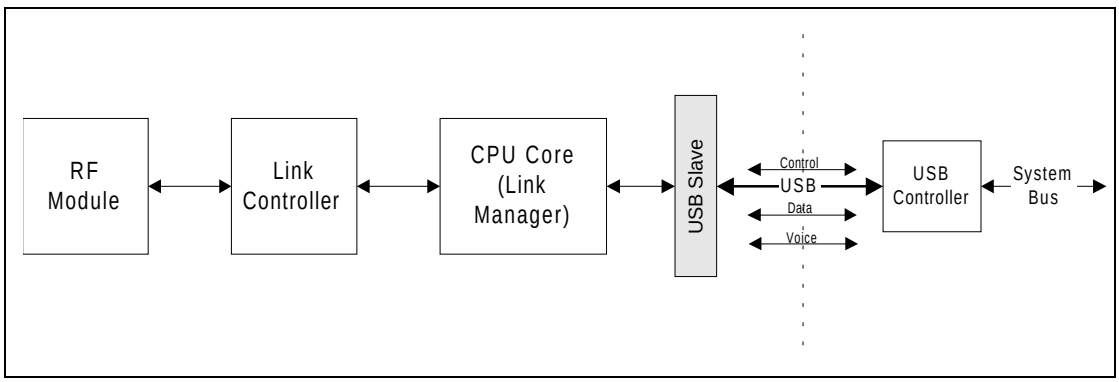


Figure 1.4: Bluetooth Block Diagram with USB HCI

1.3.2 PC Card HCI Architecture

Besides the USB interface, derivatives of the ISA bus (Compact Flash/PC Card interfaces) are an option for an integrated PC solution. Unlike USB, all traffic between the Host and the Bluetooth module will go across the PC Card bus interface. Communications between the host PC and the Bluetooth module will be primarily done directly via registers/memory. The following block diagram shows the data flow for a PC-Card HCI.

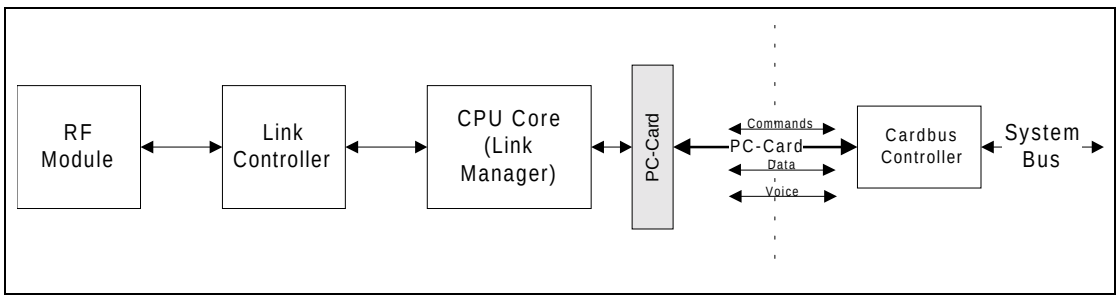


Figure 1.5: Bluetooth Block Diagram with PC-Card HCI

2 OVERVIEW OF HOST CONTROLLER TRANSPORT LAYER

The host driver stack has a transport layer between the Host Controller driver and the Host Controller. On a laptop, this transport layer might be PC Card or Universal Serial Bus (USB).

The main goal of this transport layer is transparency. The Host Controller driver (which talks to the Host Controller) should not care whether it is running over USB or a PC Card. Nor should USB or PC Card require any visibility into the data that the Host Controller driver passes to the Host Controller. This allows the interface (HCI) or the Host Controller to be upgraded without affecting the transport layer.

The Host Controller Transport Layer is described in separate documents for each physical media.

- “HCI USB Transport Layer” on page781 .
- “HCI RS232 Transport Layer” on pag e797.
- “HCI UART Transport Layer” on page819 .

3 HCI FLOW CONTROL

Flow control is used in the direction from the Host to the Host Controller to avoid filling up the Host Controller data buffers with ACL data destined for a remote device (connection handle) that is not responding. It is the Host that manages the data buffers of the Host Controller.

On Initialization, the Host will issue the Read_Buffer_Size command. Two of the return parameters of this command determine the maximum size of HCI ACL and SCO Data Packets (excluding header) sent from the Host to the Host Controller. There are also two additional return parameters that specify the total number of HCI ACL and SCO Data Packets that the Host Controller can have waiting for transmission in its buffers. When there is at least one connection to another device, or when in local loopback mode, the Host Controller uses the Number Of Completed Packets event to control the flow of data from the Host. This event contains a list of connection handles and a corresponding number of HCI Data Packets that have been completed (transmitted, flushed, or looped back to the Host) since the previous time the event was returned (or since the connection was established, if the event has not been returned before for a particular connection handle). Based on the information returned in this event, and the return parameters of the Read_Buffer_Size command that specify the total number of HCI ACL and SCO Data Packets that can be stored in the Host Controller, the Host can decide for which Connection Handles the following HCI Data Packets should be sent. After every time it has sent an HCI Data Packet, the Host must assume that the free buffer space for the corresponding link type (ACL or SCO) in the Host Controller has decreased by one HCI Data Packet. When the Host receives a new Number Of Completed Packets event, the Host gets information about how much the buffer usage has decreased since the previous time the event was returned. It can then calculate the actual current buffer usage. While the Host Controller has HCI data packets in its buffer, it must keep sending the Number Of Completed Packets event to the Host at least periodically, until it finally reports that all the pending ACL Data Packets have been transmitted or flushed. The rate with which this event is sent is manufacturer specific. Note that Number Of Completed Packets events will not report on SCO connection handles if SCO Flow Control is disabled. (See Read/Write_SCO_Flow_Control_Enable on page 678 and page 679.)

Note that for each individual Connection Handle, the data must be sent to the Host Controller in HCI Data Packets in the order in which it was created in the Host. The Host Controller must also transmit data on the air that is received from the Host for a given Connection Handle in the same order as it is received from the Host. Furthermore, data that is received on the air from another device must, for the corresponding Connection Handle, be sent in HCI Data Packets to the Host in the same order as it is received. This means that the scheduling is made on a Connection Handle basis. For each individual Connection Handle, the order of the data must not be changed from the order in which the data has been created.

In certain cases, flow control may also be necessary in the direction from the Host Controller to the Host. There is therefore a command – Set_Host_Controller_To_Host_Flow_Control – to turn flow control on or off in that direction. If turned on, it works in exactly the same way as described above. On initialization, the Host uses the Host_Buffer_Size command to notify the Host Controller about the maximum size of HCI ACL and SCO Data Packets sent from the Host Controller to the Host. The command also contains two additional command parameters to notify the Host Controller about the total number of ACL and SCO Data Packets that can be stored in the data buffers of the Host. The Host then uses the Host_Number_Of_Completed_Packets command in exactly the same way as the Host Controller uses the Number Of Completed Packets event (as was previously described in this section). The Host_Number_Of_Completed_Packets command is a special command for which no command flow control is used, and which can be sent anytime there is a connection or when in local loopback mode. This makes it possible for the flow control to work in exactly the same way in both directions, and the flow of normal commands will not be disturbed.

When the Host receives a Disconnection Complete event, the Host can assume that all HCI Data Packets that have been sent to the Host Controller for the returned Connection_Handle have been flushed, and that the corresponding data buffers have been freed. The Host Controller does not have to notify the Host about this in a Number Of Completed Packets event. If flow control is also enabled in the direction from the Host Controller to the Host, the Host Controller can after it has sent a Disconnection_Complete event assume that the Host will flush its data buffers for the sent Connection_Handle when it receives the Disconnection_Complete event. The Host does not have to notify the Host Controller about this in a Host_Number_Of_Completed_Packets command.

4 HCI COMMANDS

4.1 INTRODUCTION

The HCI provides a uniform command method of accessing the Bluetooth hardware capabilities. The HCI Link commands provide the Host with the ability to control the link layer connections to other Bluetooth devices. These commands typically involve the Link Manager (LM) to exchange LMP commands with remote Bluetooth devices. For details see “Link Manager Protocol” on pa g e183

The HCI Policy commands are used to affect the behavior of the local and remote LM. These Policy commands provide the Host with methods of influencing how the LM manages the piconet. The Host Controller & Baseband, Informational, and Status commands provide the Host access to various registers in the Host Controller.

HCI commands may take different amounts of time to be completed. Therefore, the results of commands will be reported back to the Host in the form of an event. For example, for most HCI commands the Host Controller will generate the Command Complete event when a command is completed. This event contains the return parameters for the completed HCI command. For enabling the Host to detect errors on the HCI-Transport Layer, there needs to be a timeout between the transmission of the Host’s command and the reception of the Host Controller’s response (e.g. a Command Complete or Command Status event). Since the maximum response timeout is strongly dependent on the HCI-Transport Layer used, it is recommended to use a default value of one second for this timer. This amount of time is also dependent on the number of commands unprocessed in the command queue.

4.2 TERMINOLOGY

Baseband Packet: The smallest unit of data that is transmitted by one device to another, as defined by the “Baseband Specification” on page33 .

Packet: A higher-level protocol message than the baseband packet, currently only L2CAP (see “Logical Link Control and Adaptation Protocol Specification” on page 253) is defined, but additional packet types may be defined later.

Connection Handle: A connection handle is a 12-bit identifier which is used to uniquely address a data/voice connection from one Bluetooth device to another. The connection handles can be visualized as identifying a unique data pipe that connects two Bluetooth devices. The connection handle is maintained for the lifetime of a connection, including when a device enters Park, Sniff, or Hold mode. The Connection Handle value has local scope between Host and Host Controller. There can be multiple connection handles for any given pair of Bluetooth devices but only one ACL connection.

Event: A mechanism that the HCI uses to notify the Host for command completion, link layer status changes, etc.

4.3 DATA AND PARAMETER FORMATS

- All values are in Binary and Hexadecimal Little Endian formats unless otherwise noted
- In addition, all parameters which can have negative values must use 2's complement when specifying values
- Arrayed parameters are specified using the following notation: ParameterA[i]. If more than one set of arrayed parameters are specified (e.g. ParameterA[i], ParameterB[i]), then the order of the parameters are as follows: ParameterA[0], ParameterB[0], ParameterA[1], ParameterB[1], ParameterA[2], ParameterB[2], … ParameterA[n], ParameterB[n]
- Unless noted otherwise, all parameter values are sent and received in Little Endian format (i.e. for multi-byte parameters the rightmost (Least Signification Byte) is transmitted first)
- All command and event parameters that are not-arrayed and all elements in an arrayed parameter have fixed sizes (an integer number of bytes). The parameters and the size of each not arrayed parameter (or of each element in an arrayed parameter) contained in a command or an event is specified for each command or event. The number of elements in an arrayed parameter is not fixed.
- Where bit strings are specified, the low order bit is the right hand bit, e.g. 0 is the low order bit in '10'.

4.4 EXCHANGE OF HCI-SPECIFIC INFORMATION

The Host Controller Transport Layer provides transparent exchange of HCI-specific information. These transporting mechanisms provide the ability for the Host to send HCI commands, ACL data, and SCO data to the Host Controller. These transport mechanisms also provide the ability for the Host to receive HCI events, ACL data, and SCO data from the Host Controller.

Since the Host Controller Transport Layer provides transparent exchange of HCI-specific information, the HCI specification specifies the format of the commands, events, and data exchange between the Host and the Host Controller. The next sections specify the HCI packet formats.

4.4.1 HCI Command Packet

The HCI Command Packet is used to send commands to the Host Controller from the Host. The format of the HCI Command Packet is shown in Figure 4.1, and the definition of each field is explained below. When the Host Controller completes most of the commands, a Command Complete event is sent to the Host. Some commands do not receive a Command Complete event when they have been completed. Instead, when the Host Controller receives one of these commands the Host Controller sends a Command Status event back to the Host when it has begun to execute the command. Later on, when the actions

associated with the command have finished, an event that is associated with the sent command will be sent by the Host Controller to the Host. However, if the command does not begin to execute (there may be a parameter error or the command may currently not be allowed), the event associated with the sent command will not be returned. The Command Status event will, in this case, return the appropriate error code in the Status parameter. On initial power-on, and after a reset, the Host can send a maximum of one outstanding HCI Command Packet until a Command Complete or Command Status event has been received. If an error occurs for a command for which a Command Complete event is returned, the Return_Parameters field may not contain all the return parameters specified for the command. The Status parameter, which explains the error reason and which is the first return parameter, will always be returned. If there is a Connection_Handle parameter or a BD_ADDR parameter right after the Status parameter, this parameter will also be returned so that the Host can identify to which instance of a command the Command Complete event belongs. In this case, the Connection_Handle or BD_ADDR parameter will have exactly the same value as that in the corresponding command parameter. It is implementation specific whether more parameters will be returned in case of an error.

Note: The BD_ADDR return parameter of the command Read_BD_ADDR is not used to identify to which instance of the Read_BD_ADDR command the Command Complete event belongs. It is therefore not mandatory for the Host Controller to return this parameter in case of an error.

If an error occurs for a command for which no Command Complete event is returned, all parameters returned with the event associated with this command may not be valid. The Host must take care as to which parameters may have valid values depending on the value of the Status parameter of the Complete event associated with the given command. The Command Complete and Command Status events contain a parameter called Num_HCI_Command_Packets, which indicates the number of HCI Command Packets the Host is currently allowed to send to the Host Controller. The Host Controller may buffer one or more HCI command packets, but the Host Controller must start performing the commands in the order in which they are received. The Host Controller can start performing a command before it completes previous commands. Therefore, the commands do not always complete in the order they are started. The Host Controller must be able to accept HCI Command Packets with up to 255 bytes of data excluding the HCI Command Packet header.

Each command is assigned a 2 byte Opcode used to uniquely identify different types of commands. The Opcode parameter is divided into two fields, called the OpCode Group Field (OGF) and OpCode Command Field (OCF). The OGF occupies the upper 6 bits of the Opcode, while the OCF occupies the remaining 10 bits. The OGF of 0x3F is reserved for vendor-specific debug commands. The OGF of 0x3E is reserved for Bluetooth Logo Testing. The organization of the Opcodes allows additional information to be inferred without fully decoding the entire Opcode.

Note: the OGF composed of all ‘ones’ has been reserved for vendor-specific debug commands. These commands are vendor-specific and are used during manufacturing, for a possible method for updating firmware, and for debugging.

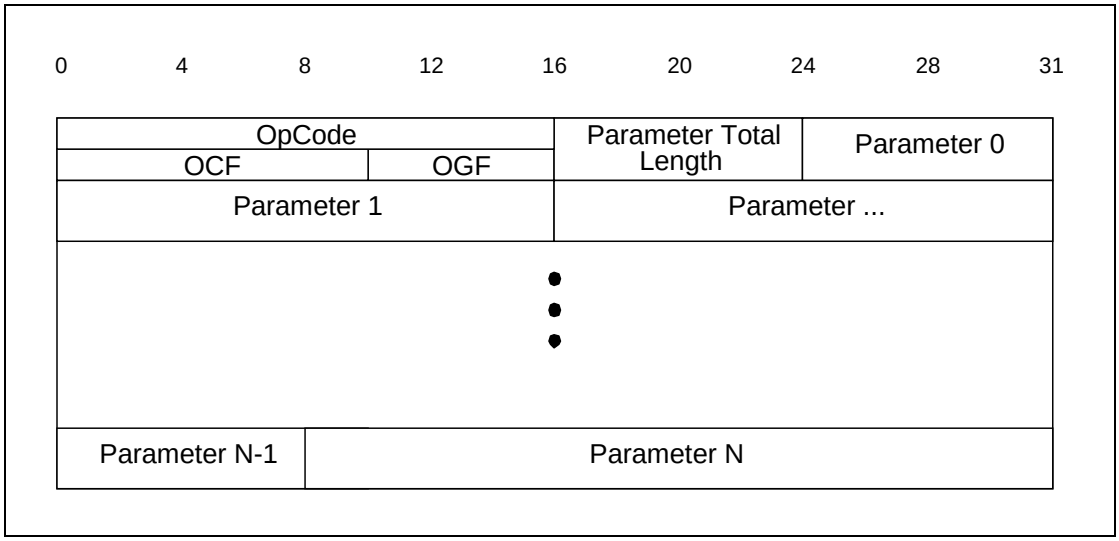


Figure 4.1: HCI Command Packet

Op_Code: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	OGFRange (6 bits): 0x00-0x3F (0x3E reserved for Bluetooth logo testing and 0x3F reserved for vendor-specific debug commands) OCF Range (10 bits): 0x0000-0x03FF

Parameter_Total_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Lengths of all of the parameters contained in this packet measured in bytes. (N.B.: total length of parameters, not number of parameters)

Parameter 0 - N: *Size: Parameter Total Length*

Value	Parameter Description
0xXX	Each command has a specific number of parameters associated with it. These parameters and the size of each of the parameters are defined for each command. Each parameter is an integer number of bytes in size.

4.4.2 HCI Event Packet

The HCI Event Packet is used by the Host Controller to notify the Host when events occur. The Host must be able to accept HCI Event Packets with up to 255 bytes of data excluding the HCI Event Packet header. The format of the HCI Event Packet is shown in Figure 4.2, and the definition of each field is explained below.

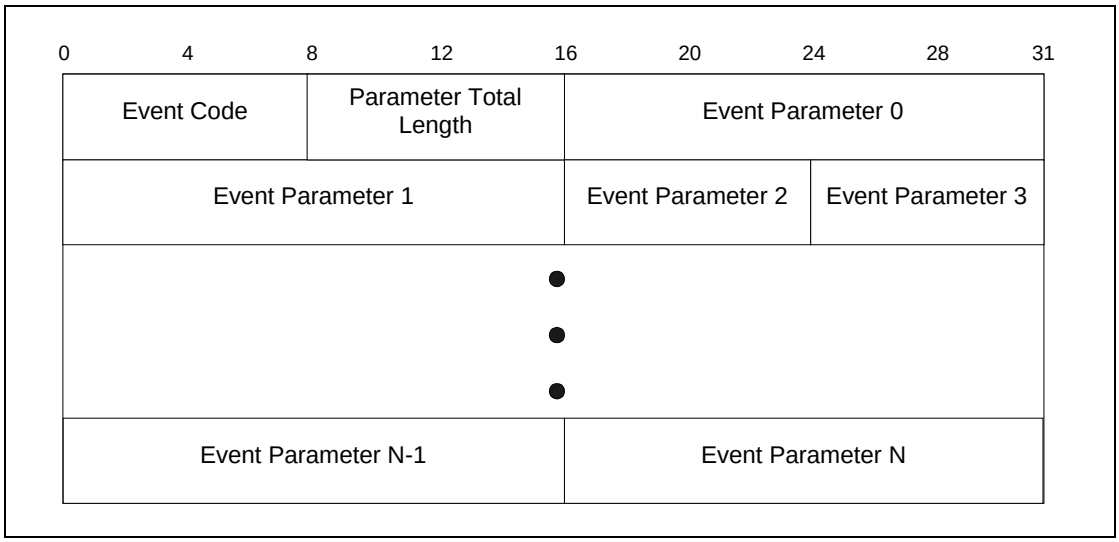


Figure 4.2: HCI Event Packet

Event_Code: *Size: 1 Byte*

Value	Parameter Description
0xXX	Each event is assigned a 1-Byte event code used to uniquely identify different types of events. Range: 0x00-0xFF (The event code 0xFF is reserved for the event code used for vendor-specific debug events. In addition, the event code 0xFE is also reserved for Bluetooth Logo Testing)

Parameter_Total_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Length of all of the parameters contained in this packet, measured in bytes

Event_Parameter 0 - N: *Size: Parameter Total Length*

Value	Parameter Description
0xXX	Each event has a specific number of parameters associated with it. These parameters and the size of each of the parameters are defined for each event. Each parameter is an integer number of bytes in size.

4.4.3 HCI Data Packets

HCI Data Packets are used to exchange data between the Host and Host Controller. The data packets are defined for both ACL and SCO data types. The format of the HCI ACL Data Packet is shown in Figure 4.3, and the format of the SCO Data Packet is shown in Figure 4.4. The definition for each of the fields in the data packets is explained below.

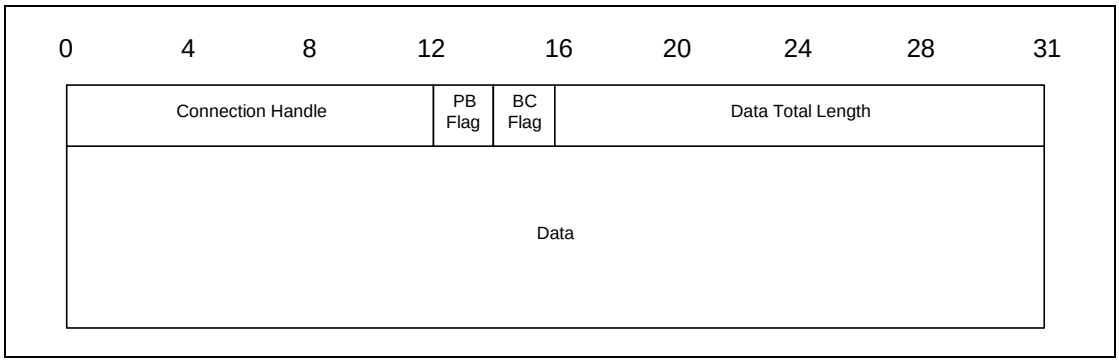


Figure 4.3: HCI ACL Data Packet

Connection_Handle: *Size: 12 Bits*

Value	Parameter Description
0xXXX	Connection Handle to be used for transmitting a data packet or segment. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use) The first time the Host sends an HCI Data Packet with Broadcast_Flag set to 01b (active broadcast) or 10b (piconet broadcast) after a power-on or a reset, the value of the Connection_Handle parameter must be a value which is not currently assigned by the Host Controller. The Host must use different connection handles for active broadcast and piconet broadcast. The Host Controller must then continue to use the same connection handles for each type of broadcast until a reset is made. Note: The Host Controller must not send a Connection Complete event containing a new Connection_Handle that it knows is used for broadcast. Note: In some situations, it may happen that the Host Controller sends a Connection Complete event before having interpreted a Broadcast packet received from the Host, and that the Connection_Handles of both Connection Complete event and HCI Data packet are the same. This conflict has to be avoided as follows: If a Connection Complete event is received containing one of the connection handles used for broadcast, the Host has to wait before sending any packets for the new connection until it receives a Number Of Completed Packets event indicating that there are no pending broadcast packets belonging to the connection handle. In addition, the Host must change the Connection_Handle used for the corresponding type of broadcast to a Connection_Handle which is currently not assigned by the Host Controller. This Connection_Handle must then be used for all the following broadcasts of that type until a reset is performed or the same conflict situation happens again. However, this will occur very rarely. The Host Controller must, in the above conflict case, be able to distinguish between the Broadcast message sent by the Host and the new connection made (this could be even a new SCO link) even though the connection handles are the same. For an HCI Data Packet sent from the Host Controller to the Host where the Broadcast_Flag is 01 or 10, the Connection_Handle parameter should contain the connection handle for the ACL connection to the master that sent the broadcast. Note: Connection handles used for Broadcast do not identify an ACL point-to-point connection, so they must not be used in any command having a Connection_Handle parameter and they will not be returned in any event having a Connection_Handle parameter except the Number Of Completed Packets event.

Flags: *Size: 2 Bits*

The Flag Bits consist of the Packet_Boundary_Flag and Broadcast_Flag. The Packet_Boundary_Flag is located in bit 4 and bit 5, and the Broadcast_Flag is located in bit 6 and 7 in the second byte of the HCI ACL Data packet.

Packet_Boundary_Flag: *Size: 2 Bits*

Value	Parameter Description
00	Reserved for future use
01	Continuing fragment packet of Higher Layer Message
10	First packet of Higher Layer Message (i.e. start of an L2CAP packet)
11	Reserved for future use

Broadcast_Flag (in packet from Host to Host Controller): *Size: 2 Bits*

Value	Parameter Description
00	No broadcast. Only point-to-point.
01	Active Broadcast: packet is sent to all active slaves (i.e. packet is usually not sent during park beacon slots), and it may be received by slaves in sniff or park mode. See note below!
10	Piconet Broadcast: packet is sent to all slaves and all slaves in park mode (i.e. packet is sent during park beacon slots if there are parked slaves), and it may be received by slaves in sniff mode. See note below!
11	Reserved for future use.

Broadcast_Flag (in packet from Host Controller to Host): *Size: 2 Bits*

Value	Parameter Description
00	Point-to-point
01	Packet received as a slave not in park mode (either Active Broadcast or Piconet Broadcast)
10	Packet received as a slave in park mode (Piconet Broadcast)
11	Reserved for future use.

Note: active broadcast packets may be sent in park beacon slots for synchronization since a slave can synchronize to any baseband packet that is preceded by the proper channel access code.

Slaves in sniff mode may or may not receive an active or piconet broadcast packet depending on whether they happen to be listening at sniff slots, when the packet is sent.

Data_Total_Length: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Length of data measured in bytes.

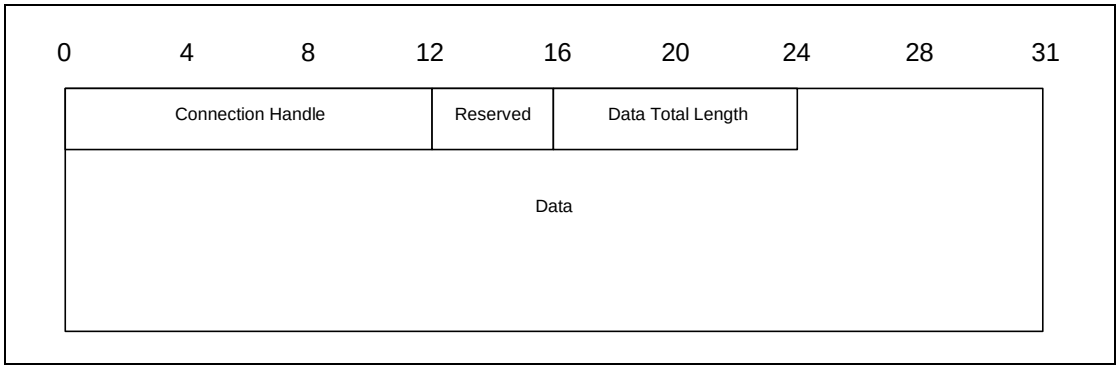


Figure 4.4: HCI SCO Data Packet

Connection_Handle: *Size: 12 Bits*

Value	Parameter Description
0xXXX	Connection handle to be used to for transmitting a SCO data packet or segment. Range: 0x0000-0x0EFF (0x0F00- 0x0FFF Reserved for future use)

The Reserved Bits consist of four bits which are located from bit 4 to bit 7 in the second byte of the HCI SCO Data packet.

Reserved: *Size: 4 Bits*

Value	Parameter Description
XXXX	Reserved for future use.

Data_Total_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Length of SCO data measured in bytes

4.5 LINK CONTROL COMMANDS

The Link Control commands allow the Host Controller to control connections to other Bluetooth devices. When the Link Control commands are used, the Link Manager (LM) controls how the Bluetooth piconets and scatternets are established and maintained. These commands instruct the LM to create and modify link layer connections with Bluetooth remote devices, perform Inquiries of other Bluetooth devices in range, and other LMP commands. For the Link Control commands, the OGF is defined as 0x01.

Command	Command Summary Description
Inquiry	The Inquiry command will cause the Bluetooth device to enter Inquiry Mode. Inquiry Mode is used to discovery other nearby Bluetooth devices.
Inquiry_Cancel	The Inquiry_Cancel command will cause the Bluetooth device to stop the current Inquiry if the Bluetooth device is in Inquiry Mode.
Periodic_Inquiry_Mode	The Periodic_Inquiry_Mode command is used to configure the Bluetooth device to perform an automatic Inquiry based on a specified period range.
Exit_Periodic_Inquiry_Mode	The Exit_Periodic_Inquiry_Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inquiry Mode.
Create_Connection	The Create_Connection command will cause the link manager to create an ACL connection to the Bluetooth device with the BD_ADDR specified by the command parameters.
Disconnect	The Disconnect command is used to terminate an existing connection.
Add_SCO_Connection	The Add_SCO_Connection command will cause the link manager to create a SCO connection using the ACL connection specified by the Connection Handle command parameter.
Accept_Connection_Request	The Accept_Connection_Request command is used to accept a new incoming connection request.
Reject_Connection_Request	The Reject_Connection_Request command is used to decline a new incoming connection request.
Link_Key_Request_Reply	The Link_Key_Request_Reply command is used to reply to a Link Key Request event from the Host Controller, and specifies the Link Key stored on the Host to be used as the link key for the connection with the other Bluetooth device specified by BD_ADDR.

Command	Command Summary Description
Link_Key_Request_Negative_Reply	The Link_Key_Request_Negative_Reply command is used to reply to a Link Key Request event from the Host Controller if the Host does not have a stored Link Key for the connection with the other Bluetooth Device specified by BD_ADDR.
PIN_Code_Request_Reply	The PIN_Code_Request_Reply command is used to reply to a PIN Code Request event from the Host Controller and specifies the PIN code to use for a connection.
PIN_Code_Request_Negative_Reply	The PIN_Code_Request_Negative_Reply command is used to reply to a PIN Code Request event from the Host Controller when the Host cannot specify a PIN code to use for a connection.
Change_Connection_Packet_Type	The Change_Connection_Packet_Type command is used to change which packet types can be used for a connection that is currently established.
Authentication_Requested	The Authentication_Requested command is used to establish authentication between the two devices associated with the specified Connection Handle.
Set_Connection_Encryption	The Set_Connection_Encryption command is used to enable and disable the link level encryption.
Change_Connection_Link_Key	The Change_Connection_Link_Key command is used to force both devices of a connection associated to the connection handle, to generate a new link key.
Master_Link_Key	The Master_Link_Key command is used to force both devices of a connection associated to the connection handle to use the temporary link key of the Master device or the regular link keys.
Remote_Name_Request	The Remote_Name_Request command is used to obtain the user-friendly name of another Bluetooth device.
Read_Remote_Supported_Features	The Read_Remote_Supported_Features command requests a list of the supported features of a remote device.
Read_Remote_Version_Information	The Read_Remote_Version_Information command will read the values for the version information for the remote Bluetooth device.
Read_Clock_Offset	The Read_Clock_Offset command allows the Host to read the clock offset of remote devices.

4.5.1 Inquiry

Command	OCF	Command Parameters	Return Parameters
HCI_Inquiry	0x0001	LAP, Inquiry_Length, Num_Responses	

Description:

This command will cause the Bluetooth device to enter Inquiry Mode. Inquiry Mode is used to discover other nearby Bluetooth devices. The LAP input parameter contains the LAP from which the inquiry access code shall be derived when the inquiry procedure is made. The Inquiry_Length parameter specifies the total duration of the Inquiry Mode and, when this time expires, Inquiry will be halted. The Num_Responses parameter specifies the number of responses that can be received before the Inquiry is halted. A Command Status event is sent from the Host Controller to the Host when the Inquiry command has been started by the Bluetooth device. When the Inquiry process is completed, the Host Controller will send an Inquiry Complete event to the Host indicating that the Inquiry has finished. The event parameters of Inquiry Complete event will have a summary of the result from the Inquiry process, which reports the number of nearby Bluetooth devices that responded. When a Bluetooth device responds to the Inquiry message, an Inquiry Result event will occur to notify the Host of the discovery.

A device which responds during an inquiry or inquiry period should always be reported to the Host in an Inquiry Result event if the device has not been reported earlier during the current inquiry or inquiry period and the device has not been filtered out using the command Set_Event_Filter. If the device has been reported earlier during the current inquiry or inquiry period, it may or may not be reported depending on the implementation (depending on if earlier results have been saved in the Host Controller and in that case how many responses that have been saved). It is recommended that the Host Controller tries to report a particular device only once during an inquiry or inquiry period.

Command Parameters:

LAP: *Size: 3 Bytes*

Value	Parameter Description
0x9E8B00–0X9E8B3F	This is the LAP from which the inquiry access code should be derived when the inquiry procedure is made; see “Bluetooth Assigned Numbers” (http://www.bluetooth.org/assigned-numbers.htm).

Inquiry_Length: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Maximum amount of time specified before the Inquiry is halted. Size: 1 byte Range: 0x01 – 0x30 Time = N * 1.28 sec Range: 1.28 – 61.44 Sec

Num_Responses: *Size: 1 Byte*

Value	**Parameter Description**
0x00	Unlimited number of responses.
0xXX	Maximum number of responses from the Inquiry before the Inquiry is halted. Range: 0x01 – 0xFF

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event is sent from the Host Controller to the Host when the Host Controller has started the Inquiry process. An Inquiry Result event will be created for each Bluetooth device which responds to the Inquiry message. In addition, multiple Bluetooth devices which respond to the Inquire message may be combined into the same event. An Inquiry Complete event is generated when the Inquiry process has completed.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed Instead, the Inquiry Complete event will indicate that this command has been completed. No Inquiry Complete event will be generated for the canceled Inquiry process.

4.5.2 Inquiry_Cancel

Command	OCF	Command Parameters	Return Parameters
HCI_Inquiry_Cancel	0x0002		Status

Description:

This command will cause the Bluetooth device to stop the current Inquiry if the Bluetooth device is in Inquiry Mode. This command allows the Host to interrupt the Bluetooth device and request the Bluetooth device to perform a different task. The command should only be issued after the Inquiry command has been issued, a Command Status event has been received for the Inquiry command, and before the Inquiry Complete event occurs.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Inquiry_Cancel command succeeded.
0x01-0xFF	Inquiry_Cancel command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Inquiry Cancel command has completed, a Command Complete event will be generated. No Inquiry Complete event will be generated for the canceled Inquiry process.

4.5.3 Periodic_Inquiry_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Periodic_ Inquiry_Mode	0x0003	Max_Period_Length, Min_Period_Length, LAP, Inquiry_Length, Num_Responses	Status

Description:

The Periodic_Inquiry_Mode command is used to configure the Bluetooth device to enter the Periodic Inquiry Mode that performs an automatic Inquiry. Max_Period_Length and Min_Period_Length define the time range between two consecutive inquiries, from the beginning of an inquiry until the start of the next inquiry. The Host Controller will use this range to determine a new random time between two consecutive inquiries for each Inquiry. The LAP input parameter contains the LAP from which the inquiry access code shall be derived when the inquiry procedure is made. The Inquiry_Length parameter specifies the total duration of the InquiryMode and, when time expires, Inquiry will be halted. The Num_Responses parameter specifies the number of responses that can be received before the Inquiry is halted. This command is completed when the Inquiry process has been started by the Bluetooth device, and a Command Complete event is sent from the Host Controller to the Host. When each of the periodic Inquiry processes are completed, the Host Controller will send an Inquiry Complete event to the Host indicating that the latest periodic Inquiry process has finished. The event parameters of Inquiry Complete event will have a summary of the result from the previous Periodic Inquiry process, which reports the number of nearby Bluetooth devices that responded. When a Bluetooth device responds to the Inquiry message an Inquiry Result event will occur to notify the Host of the discovery.
Note: Max_Period_Length > Min_Period_Length > Inquiry_Length

A device which responds during an inquiry or inquiry period should always be reported to the Host in an Inquiry Result event if the device has not been reported earlier during the current inquiry or inquiry period and the device has not been filtered out using the command Set_Event_Filter. If the device has been reported earlier during the current inquiry or inquiry period, it may or may not be reported depending on the implementation (depending on if earlier results have been saved in the Host Controller and in that case how many responses that have been saved). It is recommended that the Host Controller tries to report a particular device only once during an inquiry or inquiry period.

Command Parameters:

Max_Period_Length: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Maximum amount of time specified between consecutive inquiries. Size: 2 bytes Range: 0x03 – 0xFFFF Time = N * 1.28 sec Range: 3.84 – 83884.8 Sec 0.0 – 23.3 hours

Min_Period_Length: *Size: 2 Bytes*

Value	**Parameter Description**
N = 0xXXXX	Minimum amount of time specified between consecutive inquiries. Size: 2 bytes Range: 0x02 – 0xFFFE Time = N * 1.28 sec Range: 2.56 – 83883.52 Sec 0.0 – 23.3 hours

LAP: *Size: 3 Bytes*

Value	**Parameter Description**
0x9E8B00– 0X9E8B3F	This is the LAP from which the inquiry access code should be derived when the inquiry procedure is made, see “Bluetooth Assigned Numbers” (http://www.bluetooth.org/assigned-numbers.htm).

Inquiry_Length: *Size: 1 Byte*

Value	**Parameter Description**
N = 0xXX	Maximum amount of time specified before the Inquiry is halted. Size: 1 byte Range: 0x01 – 0x30 Time = N * 1.28 sec Range: 1.28 – 61.44 Sec

Num_Responses: *Size: 1 Byte*

Value	**Parameter Description**
0x00	Unlimited number of responses.
0xXX	Maximum number of responses from the Inquiry before the Inquiry is halted. Range: 0x01 – 0xFF

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Periodic Inquiry Mode command succeeded.
0x01-0xFF	Periodic Inquiry Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

The Periodic Inquiry Mode begins when the Host Controller sends the Command Complete event for this command to the Host. An Inquiry Result event will be created for each Bluetooth device which responds to the Inquiry message. In addition, multiple Bluetooth devices which response to the Inquiry message may be combined into the same event. An Inquiry Complete event is generated when each of the periodic Inquiry processes has completed. No Inquiry Complete event will be generated for the canceled Inquiry process.

4.5.4 Exit_Periodic_Inquiry_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Periodic_Inquiry_Mode	0x0004		Status

Description:

The Exit Periodic Inquiry Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inquiry Mode. If the local device is currently in an Inquiry process, the Inquiry process will be stopped directly and the Host Controller will no longer perform periodic inquiries until the Periodic Inquiry Mode command is reissued.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Exit Periodic Inquiry Mode command succeeded.
0x01-0xFF	Exit Periodic Inquiry Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

A Command Complete event for this command will occur when the local device is no longer in Periodic Inquiry Mode. No Inquiry Complete event will be generated for the canceled Inquiry process.

4.5.5 Create_Connection

Command	OCF	Command Parameters	Return Parameters
HCI_Create_Connection	0x0005	BD_ADDR, Packet_Type, Page_Scan_Repetition_Mode, Page_Scan_Mode, Clock_Offset, Allow_Role_Switch	

Description:

This command will cause the Link Manager to create a connection to the Bluetooth device with the BD_ADDR specified by the command parameters. This command causes the local Bluetooth device to begin the Page process to create a link level connection. The Link Manager will determine how the new ACL connection is established. This ACL connection is determined by the current state of the device, its piconet, and the state of the device to be connected. The Packet_Type command parameter specifies which packet types the Link Manager shall use for the ACL connection. The Link Manager must use only the packet type(s) specified by the Packet_Type command parameter for sending HCI ACL Data Packets. Multiple packet types may be specified for the Packet Type parameter by performing a bit-wise OR operation of the different packet types. The Link Manager may choose which packet type to be used from the list of acceptable packet types. The Page_Scan_Repetition_Mode and Page_Scan_Mode parameters specify the page scan modes supported by the remote device with the BD_ADDR. This is the information that was acquired during the inquiry process. The Clock_Offset parameter is the difference between its own clock and the clock of the remote device with BD_ADDR. Only bits 2 through 16 of the difference are used, and they are mapped to this parameter as bits 0 through 14 respectively. A Clock_Offset_Valid_Flag, located in bit 15 of the Clock_Offset parameter, is used to indicate if the Clock Offset is valid or not. A Connection handle for this connection is returned in the Connection Complete event (see below). The Allow_Role_Switch parameter specifies if the local device accepts or rejects the request of a master-slave role switch when the remote device requests it at the connection setup (in the Role parameter of the Accept_Connection_Request command) (before the local Host Controller returns a Connection Complete event). For a definition of the different packet types see the “Baseband Specification” on page33 .

Note: At least one packet type must be specified. The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device to be connected.

Packet_Type: *Size: 2 Bytes*

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	DM1
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

Page_Scan_Repetition_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode.
0x01	Optional Page Scan Mode I.
0x02	Optional Page Scan Mode II.
0x03	Optional Page Scan Mode III.
0x04 – 0xFF	Reserved.

Clock_Offset: *Size: 2 Bytes*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Clock_Offset_Valid_Flag Invalid Clock Offfset = 0 Valid Clock Offset = 1

Allow_Role_Switch: *Size: 1 Byte*

Value	Parameter Description
0x00	The local device will be a master, and will not accept a master-slave switch requested by the remote device at the connection setup.
0x01	The local device may be a master, or may become a slave after accepting a master-slave switch requested by the remote device at the connection setup.
0x02-0xFF	Reserved for future use.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Create Connection command, the Host Controller sends the Command Status event to the Host. In addition, when the LM determines the connection is established, the Host Controller, on both Bluetooth devices that form the connection, will send a Connection Complete event to each Host. The Connection Complete event contains the Connection Handle if this command is successful.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

4.5.6 Disconnect

Command	OCF	Command Parameters	Return Parameters
HCI_Disconnect	0x0006	Connection_Handle, Reason	

Description:

The Disconnection command is used to terminate an existing connection. The Connection_Handle command parameter indicates which connection is to be disconnected. The Reason command parameter indicates the reason for ending the connection. The remote Bluetooth device will receive the Reason command parameter in the Disconnection Complete event. All SCO connections on a physical link should be disconnected before the ACL connection on the same physical connection is disconnected.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for the connection being disconnected. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Reason: *Size: 1 Byte*

Value	Parameter Description
0x05, 0x13-0x15, 0x1A, 0x29	Authentication Failure error code (0x05), Other End Terminated Connection error codes (0x13-0x15), Unsupported Remote Feature error code (0x1A) and Pairing with Unit Key Not Supported error code (0x29), see Table 6.1 on page 766 for list of Error Codes.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Disconnect command, it sends the Command Status event to the Host. The Disconnection Complete event will occur at each Host when the termination of the connection has completed, and indicates that this command has been completed.
Note: No Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Disconnection Complete event will indicate that this command has been completed.

4.5.7 Add_SCO_Connection

Command	OCF	Command Parameters	Return Parameters
HCI_Add_SCO_Connection	0x0007	Connection_Handle, Packet_Type	

Description:

This command will cause the link manager to create a SCO connection using the ACL connection specified by the Connection_Handle command parameter. This command causes the local Bluetooth device to create a SCO connection. The Link Manager will determine how the new connection is established. This connection is determined by the current state of the device, its piconet, and the state of the device to be connected. The Packet_Type command parameter specifies which packet types the Link Manager should use for the connection. The Link Manager must only use the packet type(s) specified by the Packet_Type command parameter for sending HCI SCO Data Packets. Multiple packet types may be specified for the Packet_Type command parameter by performing a bitwise OR operation of the different packet types. The Link Manager may choose which packet type is to be used from the list of acceptable packet types. A Connection Handle for this connection is returned in the Connection Complete event (see below).
Note: An SCO connection can only be created when an ACL connection already exists and when it is not put in park mode. For a definition of the different packet types, see the “Baseband Specification” on pa g e33

Note: At least one packet type must be specified. The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Command Parameters:

Connection_Handle *Size 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for the ACL connection being used to create an SCO connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Packet_Type: *Size: 2 Bytes*

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Add_SCO_Connection command, it sends the Command Status event to the Host. In addition, when the LM determines the connection is established, the Host Controller, on both Bluetooth devices that form the connection, will send a Connection Complete event to each Host. The Connection Complete event contains the Connection Handle if this command is successful.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

4.5.8 Accept_Connection_Request

Command	OCF	Command Parameters	Return Parameters
HCI_Accept_Connection Request	0x0009	BD_ADDR, Role	

Description:

The Accept_Connection_Request command is used to accept a new incoming connection request. The Accept_Connection_Request command shall only be issued after a Connection Request event has occurred. The Connection Request event will return the BD_ADDR of the device which is requesting the connection. This command will cause the Link Manager to create a connection to the Bluetooth device, with the BD_ADDR specified by the command parameters. The Link Manager will determine how the new connection will be established. This will be determined by the current state of the device, its piconet, and the state of the device to be connected. The Role command parameter allows the Host to specify if the Link Manager shall perform a Master-Slave switch, and become the Master for this connection. Also, the decision to accept a connection must be completed before the connection accept timeout expires on the local Bluetooth Module.
Note: when accepting SCO connection request, the Role parameter is not used and will be ignored by the Host Controller.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device to be connected

Role: *Size: 1 Byte*

Value	Parameter Description
0x00	Become the Master for this connection. The LM will perform the Master/Slave switch.
0x01	Remain the Slave for this connection. The LM will **NOT** perform the Master/Slave switch.

Return Parameters:

None.

Event(s) generated (unless masked away):

The Accept_Connection_Request command will cause the Command Status event to be sent from the Host Controller when the Host Controller begins setting up the connection. In addition, when the Link Manager determines the connection is established, the Host Controllers on both Bluetooth devices that form the connection will send a Connection Complete event to each Host. The Connection Complete event contains the Connection Handle if this command is successful.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

4.5.9 Reject_Connection_Request

Command	OCF	Command Parameters	Return Parameters
HCI_Reject_Connection_Request	0x000A	BD_ADDR, Reason	

Description:

The Reject_Connection_Request command is used to decline a new incoming connection request. The Reject_Connection_Request command shall only be called after a Connection Request event has occurred. The Connection Request event will return the BD_ADDR of the device that is requesting the connection. The Reason command parameter will be returned to the connecting device in the Status parameter of the Connection Complete event returned to the Host of the connection device, to indicate why the connection was declined.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device to reject the connection from.

Reason: *Size: 1 Byte*

Value	Parameter Description
0x0D-0x0F	Host Reject Error Code. See Table 6.1 on page 766 for list of Error Codes and descriptions.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Reject_Connection_Request command, the Host Controller sends the Command Status event to the Host. A Connection Complete event will then be sent, both to the local Host and to the Host of the device attempting to make the connection. The Status parameter of the Connection Complete event, which is sent to the Host of the device attempting to make the connection, will contain the Reason command parameter from this command.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

4.5.10 Link_Key_Request_Reply

Command	OCF	Command Parameters	Return Parameters
HCI_Link_Key_Request_Reply	0x000B	BD_ADDR, Link_Key	Status, BD_ADDR

Description:

The Link_Key_Request_Reply command is used to reply to a Link Key Request event from the Host Controller, and specifies the Link Key stored on the Host to be used as the link key for the connection with the other Bluetooth Device specified by BD_ADDR. The Link Key Request event will be generated when the Host Controller needs a Link Key for a connection.

When the Host Controller generates a Link Key Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page183 .)

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of which the Link Key is for.

Link_Key: *Size: 16 Bytes*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Link_Key_Request_Reply command succeeded.
0x01-0xFF	Link_Key_Request_Reply command failed. See Table 6.1 on page 766 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of which the Link Key request reply has completed.

Event(s) generated (unless masked away):

The Link_Key_Request_Reply command will cause a Command Complete event to be generated.

4.5.11 Link_Key_Request_Negative_Reply

Command	OCF	Command Parameters	Return Parameters
HCI_Link_Key_Request_Negative_Reply	0x000C	BD_ADDR	Status, BD_ADDR

Description:

The Link_Key_Request_Negative_Reply command is used to reply to a Link Key Request event from the Host Controller if the Host does not have a stored Link Key for the connection with the other Bluetooth Device specified by BD_ADDR. The Link Key Request event will be generated when the Host Controller needs a Link Key for a connection.

When the Host Controller generates a Link Key Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page183 .)

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the Link Key is for.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Link_Key_Request_Negative_Reply command succeeded.
0x01-0xFF	Link_Key_Request_Negative_Reply command failed. See Table 6.1 on page 766 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the Link Key request negative reply has completed.

Event(s) generated (unless masked away):

The Link_Key_Request_Negative_Reply command will cause a Command Complete event to be generated.

4.5.12 PIN_Code_Request_Reply

Command	OCF	Command Parameters	Return Parameters
HCI_PIN_Code_Request_Reply	0x000D	BD_ADDR, PIN_Code_Length, PIN_Code	Status, BD_ADDR

Description:

The PIN_Code_Request_Reply command is used to reply to a PIN Code request event from the Host Controller, and specifies the PIN code to use for a connection. The PIN Code Request event will be generated when a connection with remote initiating device has requested pairing.

When the Host Controller generates a PIN Code Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page183 .)

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the PIN code is for.

PIN_Code_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	The PIN code length specifics the length, in bytes, of the PIN code to be used. Range: 0x01-0x10

PIN_Code: *Size: 16 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	PIN code for the device that is to be connected. The Host should insure that strong PIN Codes are used. PIN Codes can be up to a maximum of 128 bits. Note: the PIN_Code Parameter is a string parameter. Endianess does therefore not apply to the PIN_Code Parameter. The first byte of the PIN code should be transmitted first.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	PIN_Code_Request_Reply command succeeded.
0x01-0xFF	PIN_Code_Request_Reply command failed. See Table 6.1 on page 766 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the PIN Code request reply has completed.

Event(s) generated (unless masked away):

The PIN_Code_Request_Reply command will cause a Command Complete event to be generated.

4.5.13 PIN_Code_Request_Negative_Reply

Command	OCF	Command Parameters	Return Parameters
HCI_PIN_Code_ Request_Negative Reply	0x000E	BD_ADDR	Status, BD_ADDR

Description:

The PIN_Code_Request_Negative_Reply command is used to reply to a PIN Code request event from the Host Controller when the Host cannot specify a PIN code to use for a connection. This command will cause the pair request with remote device to fail.

When the Host Controller generates a PIN Code Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page183 .)

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which this command is responding to.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	PIN_Code_Request_Negative_Reply command succeeded.
0x01-0xFF	PIN_Code_Request_Negative_Reply command failed. See Table 6.1 on page 7660 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the PIN Code request negative reply has completed.

Event(s) generated (unless masked away):

The PIN_Code_Request_Negative_Reply command will cause a Command Complete event to be generated.

4.5.14 Change_Connection_Packet_Type

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Connection_ Packet_Type	0x000F	Connection_Handle, Packet_Type	

Description:

The Change_Connection_Packet_Type command is used to change which packet types can be used for a connection that is currently established. This allows current connections to be dynamically modified to support different types of user data. The Packet_Type command parameter specifies which packet types the Link Manager can use for the connection. The Link Manager must only use the packet type(s) specified by the Packet_Type command parameter for sending HCI Data Packets. The interpretation of the value for the Packet_Type command parameter will depend on the Link_Type command parameter returned in the Connection Complete event at the connection setup. Multiple packet types may be specified for the Packet_Type command parameter by bitwise OR operation of the different packet types. For a definition of the different packet types see the "Baseband Specification" on page33 .

Note: At least one packet type must be specified. The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to for transmitting and receiving voice or data. Returned from creating a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Packet_Type: *Size: 2 Bytes*

For ACL Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	DM1

Value	Parameter Description
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

For SCO Link Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Change Connection Packet Type command, the Host Controller sends the Command Status event to the Host. In addition, when the Link Manager determines the packet type has been changed for the connection, the Host Controller on the local device will send a Connection Packet Type Changed event to the Host. This will be done at the local side only.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Connection Packet Type Changed event will indicate that this command has been completed.

4.5.15 Authentication_Requested

Command	OCF	Command Parameters	Return Parameters
HCI_Authentication_ Requested	0x0011	Connection_Handle	

Description:

The Authentication_Requested command is used to try to authenticate the remote device associated with the specified Connection Handle. The Host must not issue the Authentication_Requested command with a Connection_Handle corresponding to an encrypted link. On an authentication failure, the Host Controller or Link Manager shall not automatically detach the link. The Host is responsible for issuing a Disconnect command to terminate the link if the action is appropriate.
Note**:** the Connection_Handle command parameter is used to identify the other Bluetooth device, which forms the connection. The Connection Handle should be a Connection Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to set up authentication for all Connection Handles with the same Bluetooth device end-point as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Authentication_Requested command, it sends the Command Status event to the Host. The Authentication Complete event will occur when the authentication has been completed for the connection and is indication that this command has been completed.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Authentication Complete event will indicate that this command has been completed.

Note: When the local or remote Host Controller does not have a link key for the specified Connection_Handle, it will request the link key from its Host, before the local Host finally receives the Authentication Complete event.

4.5.16 Set_Connection_Encryption

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connection_ Encryption	0x0013	Connection_Handle, Encryption_Enable	

Description:

The Set_Connection_Encryption command is used to enable and disable the link level encryption . Note**:** the Connection_Handle command parameter is used to identify the other Bluetooth device which forms the connection. The Connection Handle should be a Connection Handle for an ACL connection. While the encryption is being changed, all ACL traffic must be turned off for all Connection Handles associated with the remote device.

Command Parameters:

Connection_Handle: *Size 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to enable/disable the link layer encryption for all Connection Handles with the same Bluetooth device end-point as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Encryption_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	Turn Link Level Encryption OFF.
0x01	Turn Link Level Encryption ON.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Set_Connection_Encryption command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed enabling/disabling encryption for the connection, the Host Controller on the local Bluetooth device will send an Encryption Change event to the Host, and the Host Controller on the remote device will also generate an Encryption Change event.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Encryption Change event will indicate that this command has been completed.

4.5.17 Change_Connection_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Connection_ Link_Key	0x0015	Connection_Handle	

Description:

The Change_Connection_Link_Key command is used to force both devices of a connection associated with the connection handle to generate a new link key. The link key is used for authentication and encryption of connections.
Note: the Connection_Handle command parameter is used to identify the other Bluetooth device forming the connection. The Connection Handle should be a Connection Handle for an ACL connection. If the connection encryption is enabled, and the temporary link key is currently used, then the Bluetooth master device will automatically restart the encryption.

Command Parameters:

Connection_Handle: *Size 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Change_Connection_Link_Key command, the Host Controller sends the Command Status event to the Host. When the Link Manager has changed the Link Key for the connection, the Host Controller on the local Bluetooth device will send a Link Key Notification event and a Change Connection Link Key Complete event to the Host, and the Host Controller on the remote device will also generate a Link Key Notification event. The Link Key Notification event indicates that a new connection link key is valid for the connection.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Change Connection Link Key Complete event will indicate that this command has been completed.

4.5.18 Master_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Master_Link_Key	0x0017	Key_Flag	

Description:

The Master Link Key command is used to force the device that is master of the piconet to use the temporary link key of the master device, or the semi-permanent link keys. The temporary link key is used for encryption of broadcast messages within a piconet, and the semi-permanent link keys are used for private encrypted point-to-point communication. The Key_Flag command parameter is used to indicate which Link Key (temporary link key of the Master, or the semi-permanent link keys) shall be used.

Command Parameters:

Key_Flag: *Size: 1 Byte*

Value	Parameter Description
0x00	Use semi-permanent Link Keys.
0x01	Use Temporary Link Key.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Master_Link_Key command, the Host Controller sends the Command Status event to the Host. When the Link Manager has changed link key, the Host Controller on both the local and the remote device will send a Master Link Key Complete event to the Host. The Connection Handle on the master side will be a Connection Handle for one of the existing connections to a slave. On the slave side, the Connection Handle will be a Connection Handle to the initiating master.

The Master Link Key Complete event contains the status of this command. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Master Link Key Complete event will indicate that this command has been completed.

4.5.19 Remote_Name_Request

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_Name_Request	0x0019	BD_ADDR, Page_Scan_Repetition_Mode, Page_Scan_Mode, Clock_Offset	

Description:

The Remote_Name_Request command is used to obtain the user-friendly name of another Bluetooth device. The user-friendly name is used to enable the user to distinguish one Bluetooth device from another. The BD_ADDR command parameter is used to identify the device for which the user-friendly name is to be obtained. The Page_Scan_Repetition_Mode and Page_Scan_Mode command parameters specify the page scan modes supported by the remote device with the BD_ADDR. This is the information that was acquired during the inquiry process. The Clock_Offset parameter is the difference between its own clock and the clock of the remote device with BD_ADDR. Only bits 2 through 16 of the difference are used and they are mapped to this parameter as bits 0 through 14 respectively. A Clock_Offset_Valid_Flag, located in bit 15 of the Clock_Offset command parameter, is used to indicate if the Clock Offset is valid or not.
Note: if no connection exists between the local device and the device corresponding to the BD_ADDR, a temporary link layer connection will be established to obtain the name of the remote device.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the device whose name is requested.

Page_Scan_Repetition_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode.
0x01	Optional Page Scan Mode I.
0x02	Optional Page Scan Mode II.
0x03	Optional Page Scan Mode III.
0x04 – 0xFF	Reserved.

Clock_Offset: *Size: 2 Bytes*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Clock_Offset_Valid_Flag Invalid Clock Offfset = 0 Valid Clock Offset = 1

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Remote_Name_Request command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to obtain the remote name, the Host Controller on the local Bluetooth device will send a Remote Name Request Complete event to the Host. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Remote Name Request Complete event will indicate that this command has been completed.

4.5.20 Read_Remote_Supported_Features

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Remote_Supported_Features	0x001B	Connection_Handle	

Description:

This command requests a list of the supported features for the remote device identified by the Connection_Handle parameter. The Connection_Handle must be a Connection_Handle for an ACL connection. The Read Remote Supported Features Complete event will return a list of the LMP features. For details see "Link Manager Protocol" on page183 .

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's LMP-supported features list to get. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Read_Remote_Supported_Features command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to determine the remote features, the Host Controller on the local Bluetooth device will send a Read Remote Supported Features Complete event to the Host. The Read Remote Supported Features Complete event contains the status of this command, and parameters describing the supported features of the remote device. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Read Remote Supported Features Complete event will indicate that this command has been completed.

4.5.21 Read_Remote_Version_Information

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Remote_Version_ Information	0x001D	Connection_Handle	

Description:

This command will obtain the values for the version information for the remote Bluetooth device identified by the Connection_Handle parameter . The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's version information to get. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Read_Remote_Version_Information command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to determine the remote version information, the Host Controller on the local Bluetooth device will send a Read Remote Version Information Complete event to the Host. The Read Remote Version Information Complete event contains the status of this command, and parameters describing the version and subversion of the LMP used by the remote device.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the Read Remote Version Information Complete event will indicate that this command has been completed.

4.5.22 Read_Clock_Offset

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Clock_Offset	0x001F	Connection_Handle	

Description:

Both the System Clock and the clock offset to a remote device are used to determine what hopping frequency is used by a remote device for page scan. This command allows the Host to read clock offset to remote devices. The clock offset can be used to speed up the paging procedure when the local device tries to establish a connection to a remote device, for example, when the local Host has issued Create_Connection or Remote_Name_Request. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Clock Offset parameter is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the Read_Clock_Offset command, the Host Controller sends the Command Status event to the Host. If this command was requested at the master and the Link Manager has completed the LMP messages to obtain the Clock Offset information, the Host Controller on the local Bluetooth device will send a Read Clock Offset Complete event to the Host. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Read Clock Offset Complete event will indicate that this command has been completed. If the command is requested at the slave, the LM will immediately send a Command Status event and a Read Clock Offset Complete event to the Host, without an exchange of LMP PDU.

4.6 LINK POLICY COMMANDS

The Link Policy Commands provide methods for the Host to affect how the Link Manager manages the piconet. When Link Policy Commands are used, the LM still controls how Bluetooth piconets and scatternets are established and maintained, depending on adjustable policy parameters. These policy commands modify the Link Manager behavior that can result in changes to the link layer connections with Bluetooth remote devices.
Note: only one ACL connection can exist between two Bluetooth Devices, and therefore there can only be one ACL HCI Connection Handle for each physical link layer Connection. The Bluetooth Host Controller provides policy adjustment mechanisms to provide support for a number of different policies. This capability allows one Bluetooth module to be used to support many different usage models, and the same Bluetooth module can be incorporated in many different types of Bluetooth devices. For the Link Policy Commands, the OGF is defined as 0x02.

Command	Command Summary Description
Hold_Mode	The Hold_Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the hold mode.
Sniff_Mode	The Sniff_Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the sniff mode.
Exit_Sniff_Mode	The Exit_Sniff_Mode command is used to end the sniff mode for a connection handle which is currently in sniff mode.
Park_Mode	The Park_Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the Park mode.
Exit_Park_Mode	The Exit_Park_Mode command is used to switch the Bluetooth device from park mode back to active mode.
QoS_Setup	The QoS_Setup command is used to specify Quality of Service parameters for a connection handle.
Role_Discovery	The Role_Discovery command is used for a Bluetooth device to determine which role the device is performing for a particular Connection Handle.
Switch_Role	The Switch_Role command is used for a Bluetooth device switch the current role the device is performing for a particular connection with the specified Bluetooth device

Command	Command Summary Description
Read_Link_Policy_Settings	The Read_Link_Policy_Settings command will read the Link Policy settings for the specified Connection Handle. The Link Policy settings allow the Host to specify which Link Modes the LM can use for the specified Connection Handle.
Write_Link_Policy_Settings	The Write_Link_Policy_Settings command will write the Link Policy settings for the specified Connection Handle. The Link Policy settings allow the Host to specify which Link Modes the LM can use for the specified Connection Handle.

4.6.1 Hold_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Hold_Mode	0x0001	Connection_Handle, Hold_Mode_Max_Interval, Hold_Mode_Min_Interval	

Description:

The Hold_Mode command is used to alter the behavior of the Link Manager, and have it place the ACL baseband connection associated by the specified Connection Handle into the hold mode. The Hold_Mode_Max_Interval and Hold_Mode_Min_Interval command parameters specify the length of time the Host wants to put the connection into the hold mode. The local and remote devices will negotiate the length in the hold mode. The Hold_Mode_Max_ Interval parameter is used to specify the maximum length of the Hold interval for which the Host may actually enter into the hold mode after negotiation with the remote device. The Hold interval defines the amount of time between when the Hold Mode begins and when the Hold Mode is completed. The Hold_ Mode_Min_Interval parameter is used to specify the minimum length of the Hold interval for which the Host may actually enter into the hold mode after the negotiation with the remote device. Therefore the Hold_Mode_Min_Interval cannot be greater than the Hold_Mode_Max_Interval. The Host Controller will return the actual Hold interval in the Interval parameter of the Mode Change event, if the command is successful. This command enables the Host to support a low-power policy for itself or several other Bluetooth devices, and allows the devices to enter Inquiry Scan, Page Scan, and a number of other possible actions.
Note: the connection handle cannot be of the SCO link type
If the Host sends data to the Host Controller with a Connection_Handle corresponding to a connection in hold mode, the Host Controller will keep the data in its buffers until either the data can be transmitted (the hold mode has ended) or a flush, a flush timeout or a disconnection occurs. This is valid even if the Host has not yet been notified of the hold mode through a Mode Change event when it sends the data.
Note: the above is not valid for an HCI Data Packet sent from the Host to the Host Controller on the master side where the Connection_Handle is a Connection_Handle used for broadcast and the Broadcast_Flag is set to Active Broadcast or Piconet Broadcast. The broadcast data will then never be received by slaves in hold mode.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Hold_Mode_Max_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Maximum acceptable number of Baseband slots to wait in Hold Mode. Time Length of the Hold = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001-0xFFFF Time Range: 0.625ms - 40.9 sec

Hold_Mode_Min_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Minimum acceptable number of Baseband slots to wait in Hold Mode. Time Length of the Hold = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001-0xFFFF Time Range: 0.625 msec - 40.9 sec

Return Parameters:

None.

Event(s) generated (unless masked away):

The Host Controller sends the Command Status event for this command to the Host when it has received the Hold_Mode command. The Mode Change event will occur when the Hold Mode has started and the Mode Change event will occur again when the Hold Mode has completed for the specified connection handle. The Mode Change event signaling the end of the Hold Mode is an estimation of the hold mode ending if the event is for a remote Bluetooth device. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

4.6.2 Sniff_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Sniff_Mode	0x0003	Connection_Handle, Sniff_Max_Interval, Sniff_Min_Interval, Sniff_Attempt, Sniff_Timeout	

Description:

The Sniff Mode command is used to alter the behavior of the Link Manager and have it place the ACL baseband connection associated with the specified Connection Handle into the sniff mode. The Connection_Handle command parameter is used to identify which ACL link connection is to be placed in sniff mode. The Sniff_Max_Interval and Sniff_Min_Interval command parameters are used to specify the requested acceptable maximum and minimum periods in the Sniff Mode. The Sniff_Min_Interval cannot be greater than the Sniff_Max_Interval. The sniff interval defines the amount of time between each consecutive sniff period. The Host Controller will return the actual sniff interval in the Interval parameter of the Mode Change event, if the command is successful. For a description of the meaning of the Sniff_Attempt and Sniff_Timeout parameters, seeSection 10.8.2 on page 111 in the Baseband Specification. Sniff_Attempt is there called $N_{sniff\ attempt}$ and Sniff_Timeout is called $N_{sniff\ timeout}$. This command enables the Host to support a low-power policy for itself or several other Bluetooth devices, and allows the devices to enter Inquiry Scan, Page Scan, and a number of other possible actions.
Note: in addition, the connection handle cannot be one of SCO link type.
If the Host sends data to the Host Controller with a Connection_Handle corresponding to a connection in sniff mode, the Host Controller will keep the data in its buffers until either the data can be transmitted or a flush, a flush timeout or a disconnection occurs. This is valid even if the Host has not yet been notified of the sniff mode through a Mode Change event when it sends the data. Note that it is possible for the master to transmit data to a slave without exiting sniff mode (see description in section 10.8.2 on page 114 in "Baseband Specification").
Note: the above is not valid for an HCI Data Packet sent from the Host to the Host Controller on the master side where the Connection_Handle is a Connection_Handle used for broadcast and the Broadcast_Flag is set to Active Broadcast or Piconet Broadcast. In that case, the broadcast data will only be received by a slave in sniff mode if that slave happens to listen to the master when the broadcast is made.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Sniff_Max_Interval: *Size: 2 Byte*

Value	Parameter Description
N = 0xXXXX	Maximum acceptable number of Baseband slots between each sniff period. (Sniff_Max_Interval >= Sniff_Min_Interval) Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec - 40.9 Seconds

Sniff_Min_Interval: *Size: 2 Byte*

Value	Parameter Description
N = 0xXXXX	Minimum acceptable number of Baseband slots between each sniff period. (Sniff_Max_Interval >= Sniff_Min_Interval) Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec - 40.9 Seconds

Sniff_Attempt: *Size: 2 Byte*

Value	Parameter Description
N = 0xXXXX	Number of Baseband receive slots for sniff attempt. Length = (2 * N - 1)* 0.625 msec Range for N: 0x0001 – 0x7FFF Time Range: 0.625msec - 40.9 Seconds

Sniff_Timeout: *Size: 2 Byte*

Value	Parameter Description
N = 0xXXXX	Number of Baseband receive slots for sniff timeout. Length = (2 * N -1) * 0.625 msec if N > 0, Length = 0 msec if N = 0 Range for N: 0x0000 – 0x7FFF Time Range: 0 msec - 40.9 Seconds

Return Parameters:

None.

Event(s) generated (unless masked away):

The Host Controller sends the Command Status event for this command to the Host when it has received the Sniff_Mode command. The Mode Change event will occur when the Sniff Mode has started for the specified connection handle. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

4.6.3 Exit_Sniff_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Sniff_Mode	0x0004	Connection_Handle	

Description:

The Exit_Sniff_Mode command is used to end the sniff mode for a connection handle, which is currently in sniff mode. The Link Manager will determine and issue the appropriate LMP commands to remove the sniff mode for the associated Connection Handle.
Note: in addition, the connection handle cannot be one of SCO link type.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when Host Controller has received the Exit_Sniff_Mode command. The Mode Change event will occur when the Sniff Mode has ended for the specified connection handle.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed.

4.6.4 Park_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Park_Mode	0x0005	Connection_Handle, Beacon_Max_Interval, Beacon_Min_Interval	

Description:

The Park Mode command is used to alter the behavior of the Link Manager, and have the LM place the baseband connection associated by the specified Connection Handle into the Park mode. The Connection_Handle command parameter is used to identify which connection is to be placed in Park mode. The Connection_Handle must be a Connection_Handle for an ACL connection. The Beacon Interval command parameters specify the acceptable length of the interval between beacons. However, the remote device may request shorter interval. The Beacon_Max_Interval parameter specifies the acceptable longest length of the interval between beacons. The Beacon_Min_Interval parameter specifies the acceptable shortest length of the interval between beacons. Therefore, the Beacon Min Interval cannot be greater than the Beacon Max Interval. The Host Controller will return the actual Beacon interval in the Interval parameter of the Mode Change event, if the command is successful. This command enables the Host to support a low-power policy for itself or several other Bluetooth devices, allows the devices to enter Inquiry Scan, Page Scan, provides support for large number of Bluetooth Devices in a single piconet, and a number of other possible activities.
Note: when the Host issues the Park_Mode command, no Connection Handles for SCO connections are allowed to exist to the remote device that is identified by the Connection_Handle parameter. If one or more Connection Handles for SCO connections exist to that device, depending on the implementation, a Command Status event or a Mode Change event (following a Command Status event where Status=0x00) will be returned with the error code 0x0C "Command Disallowed".
If the Host sends data to the Host Controller with a Connection_Handle corresponding to a parked connection, the Host Controller will keep the data in its buffers until either the data can be transmitted (after unpark) or a flush, a flush timeout or a disconnection occurs. This is valid even if the Host has not yet been notified of the park mode through a Mode Change event when it sends the data.
Note: the above is not valid for an HCI Data Packet sent from the Host to the Host Controller on the master side where the Connection_Handle is a Connection_Handle used for Piconet Broadcast and the Broadcast_Flag is set to Piconet Broadcast. In that case, slaves in park mode will also receive the broadcast data. (If the Broadcast_Flag is set to Active Broadcast, the broadcast data will usually not be received by slaves in park mode.)
It is possible for the Host Controller to do an automatic unpark to transmit data

and then park the connection again depending on the value of the Link_Policy_Settings parameter (see Write_Link_Policy_Settings) and depending on whether the implementation supports this or not (optional feature). The optional feature of automatic unpark/park can also be used for link supervision. Whether Mode Change events are returned or not at automatic unpark/park if this is implemented, is vendor specific. This could be controlled by a vendor specific HCI command.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Beacon_Max_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Maximum acceptable number of Baseband slots between consecutive beacons. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec - 40.9 Seconds

Beacon_Min_Interval *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Minimum acceptable number of Baseband slots between consecutive beacons Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec - 40.9 Seconds

Return Parameters:

None.

Event(s) generated (unless masked away):

The Host Controller sends the Command Status event for this command to the Host when it has received the Park_Mode command. The Mode Change event will occur when the Park Mode has started for the specified connection handle. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

4.6.5 Exit_Park_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Park_Mode	0x0006	Connection_Handle	

Description:

The Exit_Park_Mode command is used to switch the Bluetooth device from park mode back to active mode. This command may only be issued when the device associated with the specified Connection Handle is in Park Mode. The Connection_Handle must be a Connection_Handle for an ACL connection. This function does not complete immediately.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when the Host Controller has received the Exit_Park_Mode command. The Mode Change event will occur when the Park Mode has ended for the specified connection handle. Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed.

4.6.6 QoS_Setup

Command	OCF	Command Parameters	Return Parameters
HCI_QoS_Setup	0x0007	Connection_Handle, Flags, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation	

Description:

The QoS_Setup command is used to specify Quality of Service parameters for a connection handle. The Connection_Handle must be a Connection_Handle for an ACL connection. These QoS parameter are the same parameters as L2CAP QoS. For more detail see “Logical Link Control and Adaptation Protocol Specification” on page253 . This allows the Link Manager to have all of the information about what the Host is requesting for each connection. The LM will determine if the QoS parameters can be met. Bluetooth devices that are both slaves and masters can use this command. When a device is a slave, this command will trigger an LMP request to the master to provide the slave with the specified QoS as determined by the LM. When a device is a master, this command is used to request a slave device to accept the specified QoS as determined by the LM of the master. The Connection_Handle command parameter is used to identify for which connection the QoS request is requested.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection for the QoS Setup. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flags: *Size: 1 Byte*

Value	Parameter Description
0x00 – 0xFF	Reserved for Future Use.

Service_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	No Traffic.
0x01	Best Effort.
0x02	Guaranteed.
0x03-0xFF	Reserved for Future Use.

Token_Rate: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Token Rate in bytes per second.

Peak_Bandwidth: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Peak Bandwidth in bytes per second.

Latency: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Latency in microseconds.

Delay_Variation: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Delay Variation in microseconds.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Host Controller receives the QoS_Setup command, the Host Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to establish the requested QoS parameters, the Host Controller on the local Bluetooth device will send a QoS Setup Complete event to the Host, and the event may also be generated on the remote side if there was LMP negotiation. The values of the parameters of the QoS Setup Complete event may, however, be different on the initiating and the remote side. The QoS Setup Complete event returned by the Host Controller on the local side contains the status of this command, and returned QoS parameters describing the supported QoS for the connection.
Note: No Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, the QoS Setup Complete event will indicate that this command has been completed.

4.6.7 Role_Discovery

Command	OCF	Command Parameters	Return Parameters
HCI_Role_Discovery	0x0009	Connection_Handle	Status, Connection_Handle, Current_Role

Description:

The Role_Discovery command is used for a Bluetooth device to determine which role the device is performing for a particular Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Role_Discovery command succeeded,
0x01-0xFF	Role_Discovery command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection the Role_Discovery command was issued on. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Current_Role: *Size: 1 Byte*

Value	Parameter Description
0x00	Current Role is Master for this Connection Handle
0x01	Current Role is Slave for this Connection Handle.

Event(s) generated (unless masked away):

When the Role_Discovery command has completed, a Command Complete event will be generated.

4.6.8 Switch_Role

Command	OCF	Command Parameters	Return Parameters
HCI_Switch_Role	0x000B	BD_ADDR, Role	

Description:

The Switch_Role command is used for a Bluetooth device to switch the current role the device is performing for a particular connection with another specified Bluetooth device. The BD_ADDR command parameter indicates for which connection the role switch is to be performed. The Role indicates the requested new role that the local device performs.
Note: the BD_ADDR command parameter must specify a Bluetooth device for which a connection already exists.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the connected device with which a role switch is to be performed.

Role: *Size: 1 Byte*

Value	Parameter Description
0x00	Change own Role to Master for this BD_ADDR.
0x01	Change own Role to Slave for this BD_ADDR.

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when the Host Controller has received the Switch_Role command. When the role switch is performed, a Role Change event will occur to indicate that the roles have been changed, and will be communicated to both Hosts.
Note: no Command Complete event will be sent by the Host Controller to indicate that this command has been completed. Instead, only the Role Change event will indicate that this command has been completed.

4.6.9 Read_Link_Policy_Settings

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Policy_Settings	0x000C	Connection_Handle	Status, Connection_Handle Link_Policy_Settings

Description:

This command will read the Link Policy setting for the specified Connection Handle. The Link_Policy_Settings parameter determines the behavior of the local Link Manager when it receives a request from a remote device or it determines itself to change the master-slave role or to enter the hold, sniff, or park mode. The local Link Manager will automatically accept or reject such a request from the remote device, and may even autonomously request itself, depending on the value of the Link_Policy_Settings parameter for the corresponding Connection_Handle. When the value of the Link_Policy_Settings parameter is changed for a certain Connection_Handle, the new value will only be used for requests from a remote device or from the local Link Manager itself made after this command has been completed. The Connection_Handle must be a Connection_Handle for an ACL connection. By enabling each mode individually, the Host can choose any combination needed to support various modes of operation. Multiple LM policies may be specified for the Link_Policy_Settings parameter by performing a bitwise OR operation of the different activity types.

Note: The local device may be forced into hold mode (regardless of whether the local device is master or slave) by the remote device regardless of the value of the Link_Policy_Settings parameter. The forcing of hold mode can however only be done once the connection has already been placed into hold mode through an LMP request (the Link_Policy_Settings determine if requests from a remote device should be accepted or rejected). The forcing of hold mode can after that be done as long as the connection lasts regardless of the setting for hold mode in the Link_Policy_Settings parameter.

Note that the previous description implies that if the implementation in the remote device is a "polite" implementation that does not force another device into hold mode via LMP PDUs, then the Link_Policy_Settings will never be overruled.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Link_Policy_Settings command succeeded.
0x01-0xFF	Read_Link_Policy_Settings command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Policy_Settings *Size: 2 Bytes*

Value	Parameter Description
0x0000	Disable All LM Modes.
0x0001	Enable Master Slave Switch.
0x0002	Enable Hold Mode.
0x0004	Enable Sniff Mode.
0x0008	Enable Park Mode.
0x0010 – 0x8000	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Link_Policy_Settings command has completed, a Command Complete event will be generated.

4.6.10 Write_Link_Policy_Settings

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Link_Policy_Settings	0x000D	Connection_Handle, Link_Policy_Settings	Status, Connection_Handle

Description:

This command will write the Link Policy setting for the specified Connection Handle. The Link_Policy_Settings parameter determines the behavior of the local Link Manager when it receives a request from a remote device or it determines itself to change the master-slave role or to enter the hold, sniff, or park mode. The local Link Manager will automatically accept or reject such a request from the remote device, and may even autonomously request itself, depending on the value of the Link_Policy_Settings parameter for the corresponding Connection_Handle. When the value of the Link_Policy_Settings parameter is changed for a certain Connection_Handle, the new value will only be used for requests from a remote device or from the local Link Manager itself made after this command has been completed. The Connection_Handle must be a Connection_Handle for an ACL connection. By enabling each mode individually, the Host can choose any combination needed to support various modes of operation. Multiple LM policies may be specified for the Link_Policy_Settings parameter by performing a bitwise OR operation of the different activity types.

Note: The local device may be forced into hold mode (regardless of whether the local device is master or slave) by the remote device regardless of the value of the Link_Policy_Settings parameter. The forcing of hold mode can however only be done once the connection has already been placed into hold mode through an LMP request (the Link_Policy_Settings determine if requests from a remote device should be accepted or rejected). The forcing of hold mode can after that be done as long as the connection lasts regardless of the setting for hold mode in the Link_Policy_Settings parameter.

Note that the previous description implies that if the implementation in the remote device is a "polite" implementation that does not force another device into hold mode via LMP PDUs, then the Link_Policy_Settings will never be overruled.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Policy_Settings *Size: 2 Bytes*

Value	Parameter Description
0x0000	Disable All LM Modes **Default**.
0x0001	Enable Master Slave Switch.
0x0002	Enable Hold Mode.
0x0004	Enable Sniff Mode.
0x0008	Enable Park Mode.
0x0010 – 0x8000	Reserved for Future Use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Link_Policy_Settings command succeeded.
0x01-0xFF	Write_Link_Policy_Settings command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Link_Policy_Settings command has completed, a Command Complete event will be generated.

4.7 HOST CONTROLLER & BASEBAND COMMANDS

The Host Controller & Baseband Commands provide access and control to various capabilities of the Bluetooth hardware. These parameters provide control of Bluetooth devices and of the capabilities of the Host Controller, Link Manager, and Baseband. The host device can use these commands to modify the behavior of the local device. For the HCI Control and Baseband Commands, the OGF is defined as 0x03

Command	Command Summary Description
Set_Event_Mask	The Set_Event_Mask command is used to control which events are generated by the HCI for the Host.
Reset	The Reset command will reset the Bluetooth Host Controller, Link Manager, and the radio module.
Set_Event_Filter	The Set_Event_Filter command is used by the Host to specify different event filters. The Host may issue this command multiple times to request various conditions for the same type of event filter and for different types of event filters.
Flush	The Flush command is used to discard all data that is currently pending for transmission in the Host Controller for the specified connection handle, even if there currently are chunks of data that belong to more than one L2CAP packet in the Host Controller.
Read_PIN_Type	The Read_PIN_Type command is used for the Host to read the value that is specified to indicate whether the Host supports variable PIN or only fixed PINs.
Write_PIN_Type	The Write_PIN_Type command is used for the Host to specify whether the Host supports variable PIN or only fixed PINs.
Create_New_Unit_Key	The Create_New_Unit_Key command is used to create a new unit key.
Read_Stored_Link_Key	The Read_Stored_Link_Key command provides the ability to read one or more link keys stored in the Bluetooth Host Controller.
Write_Stored_Link_Key	The Write_Stored_Link_Key command provides the ability to write one or more link keys to be stored in the Bluetooth Host Controller.

Command	Command Summary Description
Delete_Stored_Link_Key	The Delete_Stored_Link_Key command provides the ability to remove one or more of the link keys stored in the Bluetooth Host Controller.
Change_Local_Name	The Change_Local_Name command provides the ability to modify the user-friendly name for the Bluetooth device.
Read_Local_Name	The Read_Local_Name command provides the ability to read the stored user-friendly name for the Bluetooth device.
Read_Connection_Accept_Timeout	The Read_Connection_Accept_Timeout command will read the value for the Connection_Accept_Timeout configuration parameter, which allows the Bluetooth hardware to automatically deny a connection request after a specified period has occurred, and to refuse a new connection.
Write_Connection_Accept_Timeout	The Write_Connection_Accept_Timeout will write the value for the Connection_Accept_Timeout configuration parameter, which allows the Bluetooth hardware to automatically deny a connection request after a specified period has occurred, and to refuse a new connection.
Read_Page_Timeout	The Read_Page_Timeout command will read the value for the Page_Reply_Timeout configuration parameter, which allows the Bluetooth hardware to define the amount of time a connection request will wait for the remote device to respond before the local device returns a connection failure.
Write_Page_Timeout	The Write_Page_Timeout command will write the value for the Page_Reply_Timeout configuration parameter, which allows the Bluetooth hardware to define the amount of time a connection request will wait for the remote device to respond before the local device returns a connection failure.
Read_Scan_Enable	The Read_Scan_Enable command will read the value for the Scan_Enable configuration parameter, which controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices.

Command	Command Summary Description
Write_Scan_Enable	The Write_Scan_Enable command will write the value for the Scan_Enable configuration parameter, which controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices.
Read_Page_Scan_Activity	The Read_Page_Scan_Activity command will read the values for the Page_Scan_Interval and Page_Scan_Window configuration parameters. Page_Scan_Interval defines the amount of time between consecutive page scans. Page_Scan_Window defines the duration of the page scan.
Write_Page_Scan_Activity	The Write_Page_Scan_Activity command will write the value for Page_Scan_Interval and_ Page_Scan_Window configuration parameters. Page_Scan_Interval defines the amount of time between consecutive page scans. Page_Scan_Window defines the duration of the page scan.
Read_Inquiry_Scan_Activity	The Read_Inquiry_Scan_Activity command will read the value for Inquiry_Scan_Interval and Inquiry_Scan_Window configuration parameters. Inquiry_Scan_Interval defines the amount of time between consecutive inquiry scans. Inquiry_Scan_Window defines the amount of time for the duration of the inquiry scan.
Write_Inquiry_Scan_Activity	The Write_Inquiry_Scan_Activity command will write the value for Inquiry_Scan_Interval and Inquiry_Scan_Window configuration parameters. Inquiry_Scan_Interval defines the amount of time between consecutive inquiry scans. Inquiry_Scan_Window defines the amount of time for the duration of the inquiry scan.
Read_Authentication_Enable	The Read_Authentication_Enable command will read the value for the Authentication_ Enable parameter, which controls whether the Bluetooth device will require authentication for each connection with other Bluetooth devices.
Write_Authentication_Enable	The Write_Authentication_Enable command will write the value for the Authentication_ Enable parameter, which controls whether the Bluetooth device will require authentication for each connection with other Bluetooth devices.
Read_Encryption_Mode	The Read_Encryption_Mode command will read the value for the Encryption_Mode parameter, which controls whether the Bluetooth device will require encryption for each connection with other Bluetooth devices.

Command	Command Summary Description
Write_Encryption_Mode	The Write_Encryption_Mode command will write the value for the Encryption_Mode parameter, which controls whether the Bluetooth device will require encryption for each connection with other Bluetooth devices.
Read_Class_of_Device	The Read_Class_of_Device command will read the value for the Class_of_Device parameter, which is used to indicate its capabilities to other devices.
Write_Class_of_Device	The Write_Class_of_Device command will write the value for the Class_of_Device parameter, which is used to indicate its capabilities to other devices.
Read_Voice_Setting	The Read_Voice_Setting command will read the values for the Voice_Setting parameter, which controls all the various settings for the voice connections.
Write_Voice_Setting	The Write_Voice_Setting command will write the values for the Voice_Setting parameter, which controls all the various settings for the voice connections.
Read_Automatic_Flush_Timeout	The Read_Automatic_Flush_Timeout will read the value for the Flush_Timeout parameter for the specified connection handle. The Flush_Timeout parameter is only used for ACL connections.
Write_Automatic_Flush_Timeout	The Write_Automatic_Flush_Timeout will write the value for the Flush_Timeout parameter for the specified connection handle. The Flush_Timeout parameter is only used for ACL connections.
Read_Num_Broadcast_Retransmissions	The Read_Num_Broadcast_Retransmissions command will read the parameter value for the Number of Broadcast Retransmissions for the device. Broadcast packets are not acknowledged and are unreliable. This parameter is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times.
Write_Num_Broadcast_Retransmissions	The Write_Num_Broadcast_Retransmissions command will write the parameter value for the Number of Broadcast Retransmissions for the device. Broadcast packets are not acknowledged and are unreliable. This parameter is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times.

Command	Command Summary Description
Read_Hold_Mode_Activity	The Read_Hold_Mode_Activity command will read the value for the Hold_Mode_Activity parameter. This value is used to determine what activity the device should do when it is in hold mode.
Write_Hold_Mode_Activity	The Write_Hold_Mode_Activity command will write the value for the Hold_Mode_Activity parameter. This value is used to determine what activity the device should do when it is in hold mode.
Read_Transmit_Power_Level	The Read_Transmit_Power_Level command will read the values for the Transmit_Power_ Level parameter for the specified Connection Handle.
Read_SCO_Flow_Control_Enable	The Read_SCO_Flow_Control_Enable command provides the ability to read the SCO_Flow_Control_Enable setting. By using this setting, the Host can decide if the Host Controller will send Number Of Completed Packets events for SCO Connection Handles.
Write_SCO_Flow_Control_Enable	The Write_SCO_Flow_Control_Enable command provides the ability to write the SCO_Flow_Control_Enable setting. By using this setting, the Host can decide if the Host Controller will send Number Of Completed Packets events for SCO Connection Handles.
Set_Host_Controller_To_Host_Flow_Control	The Set_Host_Controller_To_Host_Flow_Control command is used by the Host to turn flow control on or off in the direction from the Host Controller to the Host.
Host_Buffer_Size	The Host_Buffer_Size command is used by the Host to notify the Host Controller about its buffer sizes for ACL and SCO data. The Host Controller will segment the data to be transmitted from the Host Controller to the Host, so that data contained in HCI Data Packets will not exceed these sizes.
Host_Number_Of_Completed_Packets	The Host_Number_Of_Completed_Packets command is used by the Host to indicate to the Host Controller when the Host is ready to receive more HCI packets for any connection handle.
Read_Link_Supervision_Timeout	The Read_Link_Supervision_Timeout command will read the value for the Link_Supervision_Timeout parameter for the device. This parameter is used by the master or slave Bluetooth device to monitor link loss.

Command	Command Summary Description
Write_Link_Supervision_Timeout	The Write_Link_Supervision_Timeout command will write the value for the Link_Supervision_ Timeout parameter for the device. This parameter is used by the master or slave Bluetooth device to monitor link loss.
Read_Number_Of_Supported_IAC	The Read_Number_Of_Supported_IAC command will read the value for the number of Inquiry Access Codes (IAC) that the local Bluetooth device can simultaneously listen for during an Inquiry Scan.
Read_Current_IAC_LAP	The Read_Current_IAC_LAP command will read the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans.
Write_Current_IAC_LAP	The Write_Current_IAC_LAP will write the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans.
Read_Page_Scan_Period_Mode	The Read_Page_Scan_Period_Mode command is used to read the mandatory Page_Scan_Period_Mode of the local Bluetooth device.
Write_Page_Scan_Period_Mode	The Write_Page_Scan_Period_Mode command is used to write the mandatory Page_Scan_Period_Mode of the local Bluetooth device.
Read_Page_Scan_Mode	The Read_Page_Scan_Mode command is used to read the default Page_Scan_Mode of the local Bluetooth device.
Write_Page_Scan_Mode	The Write_Page_Scan_Mode command is used to write the default Page_Scan_Mode of the local Bluetooth device.

4.7.1 Set_Event_Mask

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Event_Mask	0x0001	Event_Mask	Status

Description:

The Set_Event_Mask command is used to control which events are generated by the HCI for the Host. If the bit in the Event_Mask is set to a one, then the event associated with that bit will be enabled. The Host has to deal with each event that occurs by the Bluetooth devices. The event mask allows the Host to control how much it is interrupted.
Note: the Command Complete event, Command Status event and Number Of Completed Packets event cannot be masked. These events always occur. The Event_Mask is a bit mask of all of the events specified in Table 5.1 on page 724.

Command Parameters:

Event_Mask: *Size: 8 Bytes*

Value	Parameter Description
0x0000000000000000	No events specified
0x0000000000000001	Inquiry Complete event
0x0000000000000002	Inquiry Result event
0x0000000000000004	Connection Complete event
0x0000000000000008	Connection Request event
0x0000000000000010	Disconnection Complete event
0x0000000000000020	Authentication Complete event
0x0000000000000040	Remote Name Request Complete event
0x0000000000000080	Encryption Change event
0x0000000000000100	Change Connection Link Key Complete event
0x0000000000000200	Master Link Key Complete event
0x0000000000000400	Read Remote Supported Features Complete event
0x0000000000000800	Read Remote Version Information Complete event
0x0000000000001000	QoS Setup Complete event
0x0000000000002000	Command Complete event
0x0000000000004000	Command Status event
0x0000000000008000	Hardware Error event
0x0000000000010000	Flush Occurred event
0x0000000000020000	Role Change event

Value	Parameter Description
0x0000000000040000	Number Of Completed Packets event
0x0000000000080000	Mode Change event
0x0000000000100000	Return Link Keys event
0x0000000000200000	PIN Code Request event
0x0000000000400000	Link Key Request event
0x0000000000800000	Link Key Notificationevent
0x0000000001000000	Loopback Command event
0x0000000002000000	Data Buffer Overflow event
0x0000000004000000	Max Slots Change event
0x0000000008000000	Read Clock Offset Complete event
0x0000000010000000	Connection Packet Type Changed event
0x0000000020000000	QoS Violation event
0x0000000040000000	Page Scan Mode Change event
0x0000000080000000	Page Scan Repetition Mode Change event
0x0000000100000000 to 0x8000000000000000	Reserved for future use
0x00000000FFFFFFFF	**Default** (All events enabled)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Set_Event_Mask command succeeded.
0x01-0xFF	Set_Event_Mask command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Set_Event_Mask command has completed, a Command Complete event will be generated.

4.7.2 Reset

Command	OCF	Command Parameters	Return Parameters
HCI_Reset	0x0003		Status

Description:

The Reset command will reset the Host Controller and the Link Manager. The reset command should not affect the used HCI transport layer since the HCI transport layers have reset mechanisms of their own. After the reset is completed, the current operational state will be lost, the Bluetooth device will enter standby mode and the Host Controller will automatically revert to the default values for the parameters for which default values are defined in the specification.

Note: the Host is not allowed to send additional HCI commands before the Command Complete event related to the Reset command has been received.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Reset command succeeded, was received and will be executed.
0x01-0xFF	Reset command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the reset has been performed, a Command Complete event will be generated.

4.7.3 Set_Event_Filter

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Event_Filter	0x0005	Filter_Type, Filter_Condition_Type, Condition	Status

Description:

The Set_Event_Filter command is used by the Host to specify different event filters. The Host may issue this command multiple times to request various conditions for the same type of event filter and for different types of event filters. The event filters are used by the Host to specify items of interest, which allow the Host Controller to send only events which interest the Host. Only some of the events have event filters. By default (before this command has been issued after power-on or Reset) no filters are set, and the Auto_Accept_Flag is off (incoming connections are not automatically accepted). An event filter is added each time this command is sent from the Host and the Filter_Condition_Type is not equal to 0x00. (The old event filters will not be overwritten). To clear all event filters, the Filter_Type = 0x00 is used. The Auto_Accept_Flag will then be set to off. To clear event filters for only a certain Filter_Type, the Filter_Condition_Type = 0x00 is used.

The Inquiry Result filter allows the Host Controller to filter out Inquiry Result events. The Inquiry Result filter allows the Host to specify that the Host Controller only sends Inquiry Results to the Host if the Inquiry Result event meets one of the specified conditions set by the Host. For the Inquiry Result filter, the Host can specify one or more of the following Filter Condition Types:

1. A new device responded to the Inquiry process
2. A device with a specific Class of Device responded to the Inquiry process
3. A device with a specific BD_ADDR responded to the Inquiry process

The Inquiry Result filter is used in conjunction with the Inquiry and Periodic Inquiry command.

The Connection Setup filter allows the Host to specify that the Host Controller only sends a Connection Complete or Connection Request event to the Host if the event meets one of the specified conditions set by the Host. For the Connection Setup filter, the Host can specify one or more of the following Filter Condition Types:

1. Allow Connections from all devices
2. Allow Connections from a device with a specific Class of Device
3. Allow Connections from a device with a specific BD_ADDR

For each of these conditions, an Auto_Accept_Flag parameter allows the Host to specify what action should be done when the condition is met. The Auto_ Accept_Flag allows the Host to specify if the incoming connection should be auto accepted (in which case the Host Controller will send the Connection Complete event to the Host when the connection is completed) or if the Host should make the decision (in which case the Host Controller will send the Connection Request event to the Host, to elicit a decision on the connection).

The Connection Setup filter is used in conjunction with the Read/Write_ Scan_Enable commands. If the local device is in the process of a page scan, and is paged by another device which meets one on the conditions set by the Host, and the Auto_Accept_Flag is off for this device, then a Connection Request event will be sent to the Host by the Host Controller. A Connection Complete event will be sent later on after the Host has responded to the incoming connection attempt. In this same example, if the Auto_Accept_Flag is on, then a Connection Complete event will be sent to the Host by the Host Controller. (No Connection Request event will be sent in that case.)

The Host Controller will store these filters in volatile memory until the Host clears the event filters using the Set_Event_Filter command or until the Reset command is issued. The number of event filters the Host Controller can store is implementation dependent. If the Host tries to set more filters than the Host Controller can store, the Host Controller will return the "Memory Full" error code and the filter will not be installed.
Note: the Clear All Filters has no Filter Condition Types or Conditions.
Note: In the condition that a connection is auto accepted, a Link Key Request event and possibly also a PIN Code Request event and a Link Key Notification event could be sent to the Host by the Host Controller before the Connection Complete event is sent.

If there is a contradiction between event filters, the latest set event filter will override older ones. An example is an incoming connection attempt where more than one Connection Setup filter matches the incoming connection attempt, but the Auto-Accept_Flag has different values in the different filters.

Command Parameters:

Filter_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	Clear All Filters (Note: In this case, the Filter_Condition_type and Condition parameters should not be given, they should have a length of 0 bytes. Filter_Type should be the only parameter.)
0x01	Inquiry Result.
0x02	Connection Setup.
0x03-0xFF	Reserved for Future Use.

Filter Condition Types: For each Filter Type one or more Filter Condition types exists.

Inquiry_Result_Filter_Condition_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	A new device responded to the Inquiry process. (Note: A device may be reported to the Host in an Inquiry Result event more than once during an inquiry or inquiry period depending on the implementation, see description in Section 4.5.1 on page 561 and Section 4.5.3 on page 564)
0x01	A device with a specific Class of Device responded to the Inquiry process.
0x02	A device with a specific BD_ADDR responded to the Inquiry process.
0x03-0xFF	Reserved for Future Use

Connection_Setup_Filter_Condition_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	Allow Connections from all devices.
0x01	Allow Connections from a device with a specific Class of Device.
0x02	Allow Connections from a device with a specific BD_ADDR.
0x03-0xFF	Reserved for Future Use.

Condition: For each Filter Condition Type defined for the Inquiry Result Filter and the Connection Setup Filter, zero or more Condition parameters are required – depending on the filter condition type and filter type.

Condition for Inquiry_Result_Filter_Condition_Type = 0x00

Condition: *Size: 0 Byte*

Value	Parameter Description
	The Condition parameter is not used.

Condition for Inquiry_Result_Filter_Condition_Type = 0x01

Condition:

Size: 6 Bytes

Class_of_Device: *Size: 3 Bytes*

Value	Parameter Description
0x000000	Default, Return All Devices.
0xXXXXXX	*Class of Device* of Interest.

Class_of_Device_Mask: *Size: 3 Bytes*

Value	Parameter Description
0xXXXXXX	Bit Mask used to determine which bits of the Class of Device parameter are 'don't care'. Zero-value bits in the mask indicate the 'don't care' bits of the Class of Device.

Condition for Inquiry_Result_Filter_Condition_Type = 0x02

Condition:

Size: 6 Bytes

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of Interest

Condition for Connection_Setup_Filter_Condition_Type = 0x00

Condition:

Size: 1 Byte

Auto_Accept_Flag: *Size:1 Byte*

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). **Note: When auto accepting an incoming SCO connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02**.
0x04 – 0xFF	Reserved for future use.

Condition for Connection_Setup_Filter_Condition_Type = 0x01

Condition:

Size: 7 Bytes

Class_of_Device: *Size: 3 Bytes*

Value	Parameter Description
0x000000	Default, Return All Devices.
0xXXXXXX	*Class of Device* of Interest.

Class_of_Device_Mask: *Size: 3 Bytes*

Value	Parameter Description
0xXXXXXX	Bit Mask used to determine which bits of the Class of Device parameter are 'don't care'. Zero-value bits in the mask indicate the 'don't care' bits of the Class of Device.

Auto_Accept_Flag: *Size: 1 Byte*

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). **Note: When auto accepting an incoming SCO connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02**.
0x04 – 0xFF	Reserved for future use.

Condition for Connection_Setup_Filter_Condition_Type = 0x02

Condition:

Size: 7 Bytes

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of Interest.

Auto_Accept_Flag: *Size: 1 Byte*

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). **Note: When auto accepting an incoming SCO connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02**.
0x04 – 0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Set_Event_Filter command succeeded.
0x01-0xFF	Set_Event_Filter command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

A Command Complete event for this command will occur when the Host Controller has enabled the filtering of events. When one of the conditions are met, a specific event will occur.

4.7.4 Flush

Command	OCF	Command Parameters	Return Parameters
HCI_Flush	0x0008	Connection_Handle	Status, Connection_Handle

Description:

The Flush command is used to discard all data that is currently pending for transmission in the Host Controller for the specified connection handle, even if there currently are chunks of data that belong to more than one L2CAP packet in the Host Controller. After this, all data that is sent to the Host Controller for the same connection handle will be discarded by the Host Controller until an HCI Data Packet with the start Packet_Boundary_Flag (0x02) is received. When this happens, a new transmission attempt can be made. This command will allow higher-level software to control how long the baseband should try to retransmit a baseband packet for a connection handle before all data that is currently pending for transmission in the Host Controller should be flushed. Note that the Flush command is used for ACL connections ONLY. In addition to the Flush command, the automatic flush timers (see section 4.7.31 on page 667) can be used to automatically flush the L2CAP packet that is currently being transmitted after the specified flush timer has expired.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection to flush. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Flush command succeeded.
0x01-0xFF	Flush command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection the flush command was issued on. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

The Flush Occurred event will occur once the flush is completed. A Flush Occurred event could be from an automatic Flush or could be cause by the Host issuing the Flush command. When the Flush command has completed, a Command Complete event will be generated, to indicate that the Host caused the Flush.

4.7.5 Read_PIN_Type

Command	OCF	Command Parameters	Return Parameters
HCI_Read_PIN_Type	0x0009		Status, PIN_Type

Description:

The Read_PIN_Type command is used for the Host to read whether the Link Manager assumes that the Host supports variable PIN codes only a fixed PIN code. The Bluetooth hardware uses the PIN-type information during pairing.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_PIN_Type command succeeded.
0x01-0xFF	Read_PIN_Type command failed. See Table 6.1 on page 766 for list of Error Codes.

PIN_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	Variable PIN.
0x01	Fixed PIN.

Event(s) generated (unless masked away):

When the Read_PIN_Type command has completed, a Command Complete event will be generated.

4.7.6 Write_PIN_Type

Command	OCF	Command Parameters	Return Parameters
HCI_Write_PIN_Type	0x000A	PIN_Type	Status

Description:

The Write_PIN_Type command is used for the Host to write to the Host Controller whether the Host supports variable PIN codes or only a fixed PIN code. The Bluetooth hardware uses the PIN-type information during pairing.

Command Parameters:

PIN_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	Variable PIN.
0x01	Fixed PIN.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write PIN Type command succeeded.
0x01-0xFF	Write PIN Type command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_PIN_Type command has completed, a Command Complete event will be generated.

4.7.7 Create_New_Unit_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Create_New_Unit_Key	0x000B		Status

Description:

The Create_New_Unit_Key command is used to create a new unit key. The Bluetooth hardware will generate a random seed that will be used to generate the new unit key. All new connection will use the new unit key, but the old unit key will still be used for all current connections.
Note: this command will not have any effect for a device which doesn't use unit keys (i.e. a device which uses only combination keys).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Create New Unit Key command succeeded.
0x01-0xFF	Create New Unit Key command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Create_New_Unit_Key command has completed, a Command Complete event will be generated.

4.7.8 Read_Stored_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Stored_ Link_Key	0x000D	BD_ADDR, Read_All_Flag	Status, Max_Num_Keys, Num_Keys_Read

Description:

The Read_Stored_Link_Key command provides the ability to read one or more link keys stored in the Bluetooth Host Controller. The Bluetooth Host Controller can store a limited number of link keys for other Bluetooth devices. Link keys are shared between two Bluetooth devices, and are used for all security transactions between the two devices. A Host device may have additional storage capabilities, which can be used to save additional link keys to be reloaded to the Bluetooth Host Controller when needed. The Read_All_Flag parameter is used to indicate if all of the stored Link Keys should be returned. If Read_All_Flag indicates that all Link Keys are to be returned, then the BD_ADDR command parameter must be ignored The BD_ADDR command parameter is used to identify which link key to read. The stored Link Keys are returned by one or more Return Link Keys events.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the stored link key to be read.

Read_All_Flag: *Size: 1 Byte*

Value	Parameter Description
0x00	Return Link Key for specified BD_ADDR.
0x01	Return all stored Link Keys.
0x02-0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Stored_Link_Key command succeeded.
0x01-0xFF	Read_Stored_Link_Key command failed. See Table 6.1 on page 766 for list of Error Codes.

Max_Num_Keys: *Size: 2 Byte*

Value	Parameter Description
0xXXXX	Total Number of Link Keys that the Host Controller can store. Range: 0x0000 – 0xFFFF

Num_Keys_Read: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Number of Link Keys Read. Range: 0x0000 – 0xFFFF

Event(s) generated (unless masked away):

Zero or more instances of the Return Link Keys event will occur after the command is issued. When there are no link keys stored, no Return Link Keys events will be returned. When there are link keys stored, the number of link keys returned in each Return Link Keys event is implementation specific. When the Read Stored Link Key command has completed a Command Complete event will be generated.

4.7.9 Write_Stored_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Stored_Link_Key	0x0011	Num_Keys_To_Write, *BD_ADDR*[i], Link_Key[i]	Status, Num_Keys_Written

Description:

The Write_Stored_Link_Key command provides the ability to write one or more link keys to be stored in the Bluetooth Host Controller. The Bluetooth Host Controller can store a limited number of link keys for other Bluetooth devices. If no additional space is available in the Bluetooth Host Controller then no additional link keys will be stored. If space is limited and if all the link keys to be stored will not fit in the limited space, then the order of the list of link keys without any error will determine which link keys are stored. Link keys at the beginning of the list will be stored first. The Num_Keys_Written parameter will return the number of link keys that were successfully stored. If no additional space is available, then the Host must delete one or more stored link keys before any additional link keys are stored. The link key replacement algorithm is implemented by the Host and not the Host Controller. Link keys are shared between two Bluetooth devices and are used for all security transactions between the two devices. A Host device may have additional storage capabilities, which can be used to save additional link keys to be reloaded to the Bluetooth Host Controller when needed.
Note: Link Keys are only stored by issuing this command.

Command Parameters:

Num_Keys_To_Write: *Size: 1 Byte*

Value	Parameter Description
0xXX	Number of Link Keys to Write. Range: 0x01 - 0x0B

BD_ADDR [i]: *Size: 6 Bytes * Num_Keys_To_Write*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the associated Link Key.

Link_Key[i]: *Size: 16 Bytes * Num_Keys_To_Write*

Value	Parameter Description
0xXXXXXXXXXXXXXXXXXXXXXXXXXXXX	Link Key for the associated BD_ADDR.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Stored_Link_Key command succeeded.
0x01-0xFF	Write_Stored_Link_Key command failed. See Table 6.1 on page 766 for list of Error Codes.

Num_Keys_Written: *Size: 1 Bytes*

Value	Parameter Description
0xXX	Number of Link Keys successfully written. Range: 0x00 – 0x0B

Event(s) generated (unless masked away):

When the Write_Stored_Link_Key command has completed, a Command Complete event will be generated.

4.7.10 Delete_Stored_Link_Key

Command	OCF	Command Parameters	Return Parameters
HCI_Delete_Stored_ Link_Key	0x0012	BD_ADDR, Delete_All_Flag	Status, Num_Keys_Deleted

Description:

The Delete_Stored_Link_Key command provides the ability to remove one or more of the link keys stored in the Bluetooth Host Controller. The Bluetooth Host Controller can store a limited number of link keys for other Bluetooth devices. Link keys are shared between two Bluetooth devices and are used for all security transactions between the two devices. The Delete_All_Flag parameter is used to indicate if all of the stored Link Keys should be deleted. If the Delete_All_Flag indicates that all Link Keys are to be deleted, then the BD_ADDR command parameter must be ignored This command provides the ability to negate all security agreements between two devices. The BD_ADDR command parameter is used to identify which link key to delete. If a link key is currently in use for a connection, then the link key will be deleted when all of the connections are disconnected.

Command Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the link key to be deleted.

Delete_All_Flag: *Size: 1 Byte*

Value	Parameter Description
0x00	Delete only the Link Key for specified BD_ADDR.
0x01	Delete all stored Link Keys.
0x02-0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Delete_Stored_Link_Key command succeeded.
0x01-0xFF	Delete_Stored_Link_Key command failed. See Table 6.1 on page 766 for list of Error Codes.

Num_Keys_Deleted: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Number of Link Keys Deleted

Event(s) generated (unless masked away):

When the Delete_Stored_Link_Key command has completed, a Command Complete event will be generated.

4.7.11 Change_Local_Name

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Local_Name	0x0013	Name	Status

Description:

The Change_Local_Name command provides the ability to modify the user-friendly name for the Bluetooth device. A Bluetooth device may send a request to get the user-friendly name of another Bluetooth device. The user-friendly name provides the user with the ability to distinguish one Bluetooth device from another. The Name command parameter is a UTF-8 encoded string with up to 248 bytes in length. The Name command parameter should be null-terminated (0x00) if the UTF-8 encoded string is less than 248 bytes.
Note: the Name Parameter is a string parameter. Endianess does therefore not apply to the Name Parameter. The first byte of the name should be transmitted first.

Command Parameters:

Name: *Size: 248 Bytes*

Value	Parameter Description
	A UTF-8 encoded User-Friendly Descriptive Name for the device. If the name contained in the parameter is shorter than 248 bytes, the end of the name is indicated by a NULL byte (0x00), and the following bytes (to fill up 248 bytes, which is the length of the parameter) do not have valid values.
	Null terminated Zero length String. **Default.**

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Change_Local_Name command succeeded.
0x01-0xFF	Change_Local_Name command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Change_Local_Name command has completed, a Command Complete event will be generated.

4.7.12 Read_Local_Name

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Name	0x0014		Status, Name

Description:

The Read_Local_Name command provides the ability to read the stored user-friendly name for the Bluetooth device. The user-friendly name provides the user the ability to distinguish one Bluetooth device from another. The Name return parameter is a UTF-8 encoded string with up to 248 bytes in length. The Name return parameter will be null terminated (0x00) if the UTF-8 encoded string is less than 248 bytes.
Note: the Name Parameter is a string parameter. Endianess does therefore not apply to the Name Parameter. The first byte of the name is received first.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Local_Name command succeeded
0x01-0xFF	Read_Local_Name command failed see Table 6.1 on page 767 for list of Error Codes

Name: *Size: 248 Bytes*

Value	Parameter Description
	A UTF-8 encoded User Friendly Descriptive Name for the device. If the name contained in the parameter is shorter than 248 bytes, the end of the name is indicated by a NULL byte (0x00), and the following bytes (to fill up 248 bytes, which is the length of the parameter) do not have valid values.

Event(s) generated (unless masked away):

When the Read_Local_Name command has completed a Command Complete event will be generated.

4.7.13 Read_Connection_Accept_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Connection_ Accept_Timeout	0x0015		Status, Conn_Accept_Timeout

Description:

This command will read the value for the Connection_Accept_Timeout configuration parameter. The Connection_Accept_Timeout configuration parameter allows the Bluetooth hardware to automatically deny a connection request after a specified time period has occurred and the new connection is not accepted. The parameter defines the time duration from when the Host Controller sends a Connection Request event until the Host Controller will automatically reject an incoming connection.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Connection_Accept_Timeout command succeeded.
0x01-0xFF	Read_Connection_Accept_Timeout command failed. See Table 6.1 on page 766 for list of Error Codes.

Conn_Accept_Timeout: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Connection Accept Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xB540 Time Range: 0.625 msec -29 seconds

Event(s) generated (unless masked away):

When the Read_Connection_Timeout command has completed, a Command Complete event will be generated.

4.7.14 Write_Connection_Accept_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Connection_Accept_Timeout	0x0016	Conn_Accept_Timeout	Status

Description:

This command will write the value for the Connection_Accept_Timeout configuration parameter. The Connection_Accept_Timeout configuration parameter allows the Bluetooth hardware to automatically deny a connection request after a specified time interval has occurred and the new connection is not accepted. The parameter defines the time duration from when the Host Controller sends a Connection Request event until the Host Controller will automatically reject an incoming connection.

Command Parameters:

Conn_Accept_Timeout: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Connection Accept Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xB540 Time Range: 0.625 msec - 29 seconds Default: N = 0x1FA0 Time = 5 Sec

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Connection_Accept_Timeout command succeeded.
0x01-0xFF	Write_Connection_Accept_Timeout command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Connection_Accept_Timeout command has completed, a Command Complete event will be generated.

4.7.15 Read_Page_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Timeout	0x0017		Status, Page_Timeout

Description:

This command will read the value for the Page_Timeout configuration parameter. The Page_Timeout configuration parameter defines the maximum time the local Link Manager will wait for a baseband page response from the remote device at a locally initiated connection attempt. If this time expires and the remote device has not responded to the page at baseband level, the connection attempt will be considered to have failed.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Page_Timeout command succeeded.
0x01-0xFF	Read_Page_Timeout command failed. See Table 6.1 on page 766 for list of Error Codes.

Page_Timeout: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Page Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec -40.9 Seconds

Event(s) generated (unless masked away):

When the Read_Page_Timeout command has completed, a Command Complete event will be generated.

4.7.16 Write_Page_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Timeout	0x0018	Page_Timeout	Status

Description:

This command will write the value for the Page_Timeout configuration parameter. The Page_Timeout configuration parameter defines the maximum time the local Link Manager will wait for a baseband page response from the remote device at a locally initiated connection attempt. If this time expires and the remote device has not responded to the page at baseband level, the connection attempt will be considered to have failed.

Command Parameters:

Page_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0	Illegal Page Timeout. Must be larger than 0.
N = 0xXXXX	Page Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec -40.9 Seconds **Default: N = 0x2000 Time = 5.12 Sec**

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Page_Timeout command succeeded.
0x01-0xFF	Write_Page_Timeout command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Timeout command has completed, a Command Complete event will be generated.

4.7.17 Read_Scan_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Scan_Enable	0x0019		Status, Scan_Enable

Description:

This command will read the value for the Scan_Enable parameter. The Scan_Enable parameter controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices. If Page_Scan is enabled, then the device will enter page scan mode based on the value of the Page_Scan_Interval and Page_Scan_Window parameters. If Inquiry_Scan is enabled, then the device will enter Inquiry Scan mode based on the value of the Inquiry_Scan_Interval and Inquiry_Scan_Window parameters.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Scan_Enable command succeeded.
0x01-0xFF	Read_Scan_Enable command failed. See Table 6.1 on page 766 for list of Error Codes.

Scan_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	No Scans enabled.
0x01	Inquiry Scan enabled. Page Scan disabled.
0x02	Inquiry Scan disabled. Page Scan enabled.
0x03	Inquiry Scan enabled. Page Scan enabled.
0x04-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Scan_Enable command has completed, a Command Complete event will be generated.

4.7.18 Write_Scan_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Scan_Enable	0x001A	Scan_Enable	Status

Description:

This command will write the value for the Scan_Enable parameter. The Scan_Enable parameter controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices. If Page_Scan is enabled, then the device will enter page scan mode based on the value of the Page_Scan_Interval and Page_Scan_Window parameters. If Inquiry_Scan is enabled, then the device will enter Inquiry Scan mode based on the value of the Inquiry_Scan_Interval and Inquiry_Scan_Window parameters.

Command Parameters:

Scan_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	No Scans enabled. **Default.**
0x01	Inquiry Scan enabled. Page Scan disabled.
0x02	Inquiry Scan disabled. Page Scan enabled.
0x03	Inquiry Scan enabled. Page Scan enabled.
0x04-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Scan_Enable command succeeded.
0x01-0xFF	Write_Scan_Enable command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Scan_Enable command has completed, a Command Complete event will be generated.

4.7.19 Read_Page_Scan_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_ Activity	0x001B		Status, Page_Scan_Interval, Page_Scan_Window

Description:

This command will read the value for Page_Scan_Activity configuration parameters. The Page_Scan_Interval configuration parameter defines the amount of time between consecutive page scans. This time interval is defined from when the Host Controller started its last page scan until it begins the next page scan. The Page_Scan_Window configuration parameter defines the amount of time for the duration of the page scan. The Page_Scan_Window can only be less than or equal to the Page_Scan_Interval.
Note: Page Scan is only performed when Page_Scan is enabled (see 4.7.17 and 4.7.18).

A changed Page_Scan_Interval could change the local Page_Scan_ Repetition_Mode (see "Baseband Specification" on pa ge33, Keyword: SR-Mode).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Page_Scan_Activity command succeeded.
0x01-0xFF	Read_Page_Scan_Activity command failed. See Table 6.1 on page 766 for list of Error Codes.

Page_Scan_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec – 2560 msec

Page_Scan_Window: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec – 2560 msec

Event(s) generated (unless masked away):

When the Read_Page_Scan_Activity command has completed, a Command Complete event will be generated.

4.7.20 Write_Page_Scan_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_ Activity	0x001C	Page_Scan_Interval, Page_Scan_Window	Status

Description:

This command will write the value for Page_Scan_Activity configuration parameter. The Page_Scan_Interval configuration parameter defines the amount of time between consecutive page scans. This is defined as the time interval from when the Host Controller started its last page scan until it begins the next page scan. The Page_Scan_Window configuration parameter defines the amount of time for the duration of the page scan.The Page_Scan_Window can only be less than or equal to the Page_Scan_Interval.
Note: Page Scan is only performed when Page_Scan is enabled (see 4.7.17 and 4.7.18). A changed Page_Scan_Interval could change the local Page_ Scan_Repetition_Mode (see "Baseband Specification" on page33 , Keyword: SR-Mode).

Command Parameters:

Page_Scan_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec –2560 msec Default: N = 0x0800 Time = 1.28 Sec

Page_Scan_Window: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec – 2560 msec Default: N = 0x0012 Time = 11.25 msec

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Page_Scan_Activity command succeeded.
0x01-0xFF	Write_Page_Scan_Activity command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Activity command has completed, a Command Complete event will be generated.

4.7.21 Read_Inquiry_Scan_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Read_ Inquiry_Scan_Activity	0x001D		Status, Inquiry_Scan_Interval, Inquiry_Scan_Window

Description:

This command will read the value for Inquiry_Scan_Activity configuration parameter. The Inquiry_Scan_Interval configuration parameter defines the amount of time between consecutive inquiry scans. This is defined as the time interval from when the Host Controller started its last inquiry scan until it begins the next inquiry scan.
The Inquiry_Scan_Window configuration parameter defines the amount of time for the duration of the inquiry scan. The Inquiry_Scan_Window can only be less than or equal to the Inquiry_Scan_Interval.
Note: Inquiry Scan is only performed when Inquiry_Scan is enabled see 4.7.17 and 4.7.18).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Inquiry_Scan_Activity command succeeded.
0x01-0xFF	Read_Inquiry_Scan_Activity command failed. See Table 6.1 on page 766 for list of Error Codes.

Inquiry_Scan_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 – 2560 msec

Inquiry_Scan_Window: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 0.625 msec – 2560 msec

Event(s) generated (unless masked away):

When the Read_Inquiry_Scan_Activity command has completed, a Command Complete event will be generated.

4.7.22 Write_Inquiry_Scan_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_ Scan_Activity	0x001E	Inquiry_Scan_Interval, Inquiry_Scan_Window	Status

Description:

This command will write the value for Inquiry_Scan_Activity configuration parameter. The Inquiry_Scan_Interval configuration parameter defines the amount of time between consecutive inquiry scans. This is defined as the time interval from when the Host Controller started its last inquiry scan until it begins the next inquiry scan.
The Inquiry_Scan_Window configuration parameter defines the amount of time for the duration of the inquiry scan. The Inquiry_Scan_Window can only be less than or equal to the Inquiry_Scan_Interval.
Note: Inquiry Scan is only performed when Inquiry_Scan is enabled (see 4.7.17 and 4.7.18).

Command Parameters:

Inquiry_Scan_Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 – 2560 msec Default: N = 0x0800 Time = 1.28 Sec

Inquiry_Scan_Window: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Size: 2 Bytes Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec – 2560 msec Default: N = 0x0012 Time = 11.25 msec

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Inquiry_Scan_Activity command succeeded.
0x01-0xFF	Write_Inquiry_Scan_Activity command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Inquiry_Scan_Activity command has completed, a Command Complete event will be generated.

4.7.23 Read_Authentication_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Read_ Authentication_Enable	0x001F		Status, Authentication_Enable

Description:

This command will read the value for the Authentication_Enable parameter. The Authentication_Enable parameter controls if the local device requires to authenticate the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled will try to authenticate the other device.

Note: Changing this parameter does not affect existing connections.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Authentication_Enable command succeeded.
0x01-0xFF	Read_Authentication_Enable command failed.See Table 6.1 on page 766 for list of Error Codes.

Authentication_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	Authentication disabled.
0x01	Authentication enabled for all connections.
0x02-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Authentication_Enable command has completed, a Command Complete event will be generated.

4.7.24 Write_Authentication_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Write_ Authentication_Enable	0x0020	Authentication_Enable	Status

Description:

This command will write the value for the Authentication_Enable parameter. The Authentication_Enable parameter controls if the local device requires to authenticate the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled will try to authenticate the other device.

Note: Changing this parameter does not affect existing connections.

Command Parameters:

Authentication_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	Authentication disabled. **Default.**
0x01	Authentication enabled for all connection.
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write Authentication_Enable command succeeded.
0x01-0xFF	Write Authentication_Enable command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Authentication_Enable command has completed, a Command Complete event will be generated.

4.7.25 Read_Encryption_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Encryption_Mode	0x0021		Status, Encryption_Mode

Description:

This command will read the value for the Encryption_Mode parameter. The Encryption_Mode parameter controls if the local device requires encryption to the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled and Encryption_Mode parameter enabled will try to encrypt the connection to the other device.
Note: Changing this parameter does not affect existing connections.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Encryption_Mode command succeeded.
0x01-0xFF	Read_Encryption_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Encryption_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Encryption disabled.
0x01	Encryption only for point-to-point packets.
0x02	Encryption for both point-to-point and broadcast packets.
0x03-0xFF	Reserved.

Event(s) generated (unless masked away):

When the Read_Encryption_Mode command has completed, a Command Complete event will be generated.

4.7.26 Write_Encryption_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Encryption_Mode	0x0022	Encryption_Mode	Status

Description:

This command will write the value for the Encryption_Mode parameter. The Encryption_Mode parameter controls if the local device requires encryption to the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled and Encryption_Mode parameter enabled will try to encrypt the connection to the other device.
Note: Changing this parameter does not affect existing connections.
A temporary link key must be used when both broadcast and point-to-point traffic shall be encrypted.

The Host must not specify the Encryption_Mode parameter with more encryption capability than its local device currently supports, although the parameter is used to request the encryption capability to the remote device. Note that the Host must not request the command with the Encryption_Mode parameter set to either 0x01 or 0x02, when the local device does not support encryption. Also note that the Host must not request the command with the parameter set to 0x02, when the local device does not support broadcast encryption.

Note that the actual Encryption_Mode to be returned in an event for a new connection (or in a Connection Complete event) will only support a part of the capability, when the local device requests more encryption capability than the current remote device supports. For example, 0x00 will always be returned in the event when the remote device supports no encryption, and either 0x00 or 0x01 will be returned when it supports only point-to-point encryption.

Command Parameters:

Encryption_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Encryption disabled. **Default.**
0x01	Encryption only for point-to-point packets.
0x02	Encryption for both point-to-point and broadcast packets.
0x03-0xFF	Reserved.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Encryption_Mode command succeeded.
0x01-0xFF	Write_Encryption_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Encryption_Mode command has completed, a Command Complete event will be generated.

4.7.27 Read_Class_of_Device

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Class_of_Device	0x0023		Status, Class_of_Device

Description:

This command will read the value for the Class_of_Device parameter. The Class_of_Device parameter is used to indicate the capabilities of the local device to other devices.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Class_of_Device command succeeded.
0x01-0xFF	Read_Class_of_Device command failed. See Table 6.1 on page 766 for list of Error Codes.

Class_of_Device: *Size: 3 Bytes*

Value	Parameter Description
0xXXXXXX	Class of Device for the device.

Event(s) generated (unless masked away):

When the Read_Class_of_Device command has completed, a Command Complete event will be generated.

4.7.28 Write_Class_of_Device

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Class_of_Device	0x0024	Class_of_Device	Status

Description:

This command will write the value for the Class_of_Device parameter. The Class_of_Device parameter is used to indicate the capabilities of the local device to other devices.

Command Parameters:

Class_of_Device: *Size: 3 Bytes*

Value	Parameter Description
0xXXXXXX	Class of Device for the device.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Class_of_Device command succeeded.
0x01-0xFF	Write_Class_of_Device command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Class_of_Device command has completed, a Command Complete event will be generated.

4.7.29 Read_Voice_Setting

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Voice_Setting	0x0025		Status, Voice_Setting

Description:

This command will read the values for the Voice_Setting parameter. The Voice_Setting parameter controls all the various settings for voice connections. These settings apply to all voice connections, and **cannot** be set for individual voice connections. The Voice_Setting parameter controls the configuration for voice connections: Input Coding, Air coding format, input data format, Input sample size, and linear PCM parameter.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Voice_Setting command succeeded.
0x01-0xFF	Read_Voice_Setting command failed. See Table 6.1 on page 766 for list of Error Codes.

Voice_Setting: *Size: 2 Bytes (10 Bits meaningful)*

Value	Parameter Description
00XXXXXXXX	Input Coding: Linear
01XXXXXXXX	Input Coding: μ-law Input Coding
10XXXXXXXX	Input Coding: A-law Input Coding
11XXXXXXXX	Reserved for Future Use
XX00XXXXXX	Input Data Format: 1's complement
XX01XXXXXX	Input Data Format: 2's complement
XX10XXXXXX	Input Data Format: Sign-Magnitude
XX11XXXXXX	Reserved for Future Use
XXXX0XXXXX	Input Sample Size: 8-bit (only for Liner PCM)
XXXX1XXXXX	Input Sample Size: 16-bit (only for Liner PCM)

Value	Parameter Description
XXXXXnnnXX	Linear_PCM_Bit_Pos: # bit positions that MSB of sample is away from starting at MSB (only for Liner PCM).
XXXXXXXX00	Air Coding Format: CVSD
XXXXXXXX01	Air Coding Format: μ-law
XXXXXXXX10	Air Coding Format: A-law
XXXXXXXX11	Reserved

Event(s) generated (unless masked away):

When the Read_Voice_Setting command has completed, a Command Complete event will be generated.

4.7.30 Write_Voice_Setting

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Voice_Setting	0x0026	Voice_Setting	Status

Description:

This command will write the values for the Voice_Setting parameter. The Voice_Setting parameter controls all the various settings for the voice connections. These settings apply to all voice connections and **cannot** be set for individual voice connections. The Voice_Setting parameter controls the configuration for voice connections: Input Coding, Air coding format, input data format, Input sample size, and linear PCM parameter.

Command Parameters:

Voice_Setting: *Size: 2 Bytes (10 Bits meaningful)*

Value	Parameter Description
00XXXXXXXX	Input Coding: Linear
01XXXXXXXX	Input Coding: μ-law Input Coding
10XXXXXXXX	Input Coding: A-law Input Coding
11XXXXXXXX	Reserved for Future Use
XX00XXXXXX	Input Data Format: 1's complement
XX01XXXXXX	Input Data Format: 2's complement
XX10XXXXXX	Input Data Format: Sign-Magnitude
XX11XXXXXX	Reserved for Future Use
XXXX0XXXXX	Input Sample Size: 8 bit (only for Liner PCM)
XXXX1XXXXX	Input Sample Size: 16 bit (only for Liner PCM)
XXXXXnnnXX	Linear_PCM_Bit_Pos: # bit positions that MSB of sample is away from starting at MSB (only for Liner PCM)
XXXXXXXX00	Air Coding Format: CVSD
XXXXXXXX01	Air Coding Format: μ-law
XXXXXXXX10	Air Coding Format: A-law
XXXXXXXX11	Reserved
00011000XX	Default Condition, (X means that there is no default value for the corresponding bit. The manufacturer may use any value.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Voice_Setting command succeeded.
0x01-0xFF	Write_Voice_Setting command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Voice_Setting command has completed, a Command Complete event will be generated.

4.7.31 Read_Automatic_Flush_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Automatic_Flush_Timeout	0x0027	Connection_Handle	Status, Connection_Handle, Flush_Timeout

Description:

This command will read the value for the Flush_Timeout parameter for the specified connection handle. The Flush_Timeout parameter is used for ACL connections ONLY. The Flush_Timeout parameter defines the amount of time before all chunks of the L2CAP packet, of which a baseband packet is currently being transmitted, are automatically flushed by the Host Controller. The timeout period starts when a transmission attempt is made for the first baseband packet of an L2CAP packet. This allows ACL packets to be automatically flushed without the Host device issuing a Flush command. The Read_Automatic_Flush_Timeout command provides support for isochronous data, such as video. When the L2CAP packet that is currently being transmitted is automatically 'flushed', the Failed Contact Counter is incremented by one. The first chunk of the next L2CAP packet to be transmitted for the specified connection handle may already be stored in the Host Controller. In that case, the transmission of the first baseband packet containing data from that L2CAP packet can begin immediately. If the next L2CAP packet is not stored in the Host Controller, all data that is sent to the Host Controller after the flush for the same connection handle will be discarded by the Host Controller until an HCI Data Packet having the start Packet_Boundary_Flag (0x02) is received. When this happens, a new transmission attempt will be made.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout to read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Automatic_Flush_Timeout command succeeded.
0x01-0xFF	Read_Automatic_Flush_Timeout command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flush_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0	Timeout = ∞; No Automatic Flush
N = 0xXXXX	Flush Timeout = N * 0.625 msec Size: 11 bits Range: 0x0001 – 0x07FF

Event(s) generated (unless masked away):

When the Read_Automatic_Flush_Timeout command has completed, a Command Complete event will be generated.

4.7.32 Write_Automatic_Flush_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Automatic_Flush_Timeout	0x0028	Connection_Handle, Flush_Timeout	Status, Connection_Handle

Description:

This command will write the value for the Flush_Timeout parameter for the specified connection handle. The Flush_Timeout parameter is used for ACL connections ONLY. The Flush_Timeout parameter defines the amount of time before all chunks of the L2CAP packet, of which a baseband packet is currently being transmitted, are automatically flushed by the Host Controller. The timeout period starts when a transmission attempt is made for the first baseband packet of an L2CAP packet. This allows ACL packets to be automatically flushed without the Host device issuing a Flush command. The Write_Automatic_Flush_Timeout command provides support for isochronous data, such as video. When the L2CAP packet that is currently being transmitted is automatically 'flushed', the Failed Contact Counter is incremented by one. The first chunk of the next L2CAP packet to be transmitted for the specified connection handle may already be stored in the Host Controller. In that case, the transmission of the first baseband packet containing data from that L2CAP packet can begin immediately. If the next L2CAP packet is not stored in the Host Controller, all data that is sent to the Host Controller after the flush for the same connection handle will be discarded by the Host Controller until an HCI Data Packet having the start Packet_Boundary_Flag (0x02) is received. When this happens, a new transmission attempt will be made.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout to write to. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flush_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0	Timeout = ∞; No Automatic Flush. **Default.**
N = 0xXXXX	Flush Timeout = N * 0.625 msec Size: 11 bits Range: 0x0001 – 0x07FF

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Automatic_Flush_Timeout command succeeded.
0x01-0xFF	Write_Automatic_Flush_Timeout command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout has been written. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Automatic_Flush_Timeout command has completed, a Command Complete event will be generated.

4.7.33 Read_Num_Broadcast_Retransmissions

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Num_Broadcast_ Retransmissions	0x0029		Status, Num_Broadcast_Retran

Description:

This command will read the device's parameter value for the Number of Broadcast Retransmissions. Broadcast packets are not acknowledged and are unreliable. The Number of Broadcast Retransmissions parameter is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times. This parameter defines the number of times the device will retransmit a broadcast data packet. This parameter should be adjusted as the link quality measurement changes.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Num_Broadcast_Retransmissions command succeeded.
0x01-0xFF	Read_Num_Broadcast_Retransmissions command failed. See Table 6.1 on page 766 for list of Error Codes.

Num_Broadcast_Retran: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Number of Broadcast Retransmissions = N Range 0x00-0xFF

Event(s) generated (unless masked away):

When the Read_Num_Broadcast_Retransmission command has completed, a Command Complete event will be generated.

4.7.34 Write_Num_Broadcast_Retransmissions

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Num_Broadcast_ Retransmissions	0x002A	Num_Broadcast_Retran	Status

Description:

This command will write the device's parameter value for the Number of Broadcast Retransmissions. Broadcast packets are not acknowledged and are unreliable. The Number of Broadcast Retransmissions parameter is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times. This parameter defines the number of times the device will retransmit a broadcast data packet. This parameter should be adjusted as link quality measurement change.

Command Parameters:

Num_Broadcast_Retran: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Number of Broadcast Retransmissions = N Range 0x00-0xFF Default: N = 0x01

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Num_Broadcast_Retransmissions command succeeded.
0x01-0xFF	Write_Num_Broadcast_Retransmissions command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Num_Broadcast_Retransmissions command has completed, a Command Complete event will be generated.

4.7.35 Read_Hold_Mode_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Hold_Mode_Activity	0x002B		Status, Hold_Mode_Activity

Description:

This command will read the value for the Hold_Mode_Activity parameter. The Hold_Mode_Activity value is used to determine what activities should be suspended when the device is in hold mode. After the hold period has expired, the device will return to the previous mode of operation. Multiple hold mode activities may be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types. If no activities are suspended, then all of the current Periodic Inquiry, Inquiry Scan, and Page Scan settings remain valid during the Hold Mode. If the Hold_Mode_Activity parameter is set to Suspend Page Scan, Suspend Inquiry Scan, and Suspend Periodic Inquiries, then the device can enter a low-power state during the Hold Mode period, and all activities are suspended. Suspending multiple activities can be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types.The Hold Mode Activity is only valid if all connections are in Hold Mode.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Hold_Mode_Activity command succeeded.
0x01-0xFF	Read_Hold_Mode_Activity command failed. See Table 6.1 on page 766 for list of Error Codes.

Hold_Mode_Activity: *Size: 1 Byte*

Value	Parameter Description
0x00	Maintain current Power State.
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
0x08-0xFF	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Hold_Mode_Activity command has completed, a Command Complete event will be generated.

4.7.36 Write_Hold_Mode_Activity

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Hold_Mode_Activity	0x002C	Hold_Mode_Activity	Status

Description:

This command will write the value for the Hold_Mode_Activity parameter. The Hold_Mode_Activity value is used to determine what activities should be suspended when the device is in hold mode. After the hold period has expired, the device will return to the previous mode of operation. Multiple hold mode activities may be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types. If no activities are suspended, then all of the current Periodic Inquiry, Inquiry Scan, and Page Scan settings remain valid during the Hold Mode. If the Hold_Mode_Activity parameter is set to Suspend Page Scan, Suspend Inquiry Scan, and Suspend Periodic Inquiries, then the device can enter a low power state during the Hold Mode period and all activities are suspended. Suspending multiple activities can be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types. The Hold Mode Activity is only valid if all connections are in Hold Mode.

Command Parameters:

Hold_Mode_Activity: *Size: 1 Byte*

Value	Parameter Description
0x00	Maintain current Power State. **Default.**
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
0x08-0xFF	Reserved for Future Use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Hold_Mode_Activity command succeeded.
0x01-0xFF	Write_Hold_Mode_Activity command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Hold_Mode_Activity command has completed, a Command Complete event will be generated.

4.7.37 Read_Transmit_Power_Level

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Transmit_ Power_Level	0x002D	Connection_Handle, Type	Status, Connection_Handle, Transmit_Power_Level

Description:

This command will read the values for the Transmit_Power_Level parameter for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Transmit Power Level setting to read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Type: *Size: 1 Byte*

Value	Parameter Description
0x00	Read Current Transmit Power Level.
0x01	Read Maximum Transmit Power Level.
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Transmit_Power_Level command succeeded.
0x01-0xFF	Read_Transmit_Power_Level command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Transmit Power Level setting is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Transmit_Power_Level: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Size: 1 Byte (signed integer) Range: -30 ≤ N ≤ 20 Units: dBm

Event(s) generated (unless masked away):

When the Read_Transmit_Power_Level command has completed, a Command Complete event will be generated.

4.7.38 Read_SCO_Flow_Control_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Read_SCO_ Flow_Control_Enable	0x002E		Status, SCO_Flow_Control_Enable

Description:

The Read_SCO_Flow_Control_Enable command provides the ability to read the SCO_Flow_Control_Enable setting. By using this setting, the Host can decide if the Host Controller will send Number Of Completed Packets events for SCO Connection Handles. This setting allows the Host to enable and disable SCO flow control.
Note: the SCO_Flow_Control_Enable setting can only be changed if no connections exist.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_SCO_Flow_Control_Enable command succeeded
0x01-0xFF	Read_SCO_Flow_Control_Enable command failed see Table 6.1 on page 767 for list of Error Codes

SCO_Flow_Control_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO Flow Control is disabled. No Number of Completed Packets events will be sent from the Host Controller for SCO Connection Handles.
0x01	SCO Flow Control is enabled. Number of Completed Packets events will be sent from the Host Controller for SCO Connection Handles.

Event(s) generated (unless masked away):

When the Read_SCO_Flow_Control_Enable command has completed a Command Complete event will be generated.

4.7.39 Write_SCO_Flow_Control_Enable

Command	OCF	Command Parameters	Return Parameters
HCI_Write_SCO_Flow_Control_Enable	0x002F	SCO_Flow_Control_Enable	Status

Description:

The Write_SCO_Flow_Control_Enable command provides the ability to write the SCO_Flow_Control_Enable setting. By using this setting, the Host can decide if the Host Controller will send Number Of Completed Packets events for SCO Connection Handles. This setting allows the Host to enable and disable SCO flow control.
Note: the SCO_Flow_Control_Enable setting can only be changed if no connections exist.

Command Parameters:

SCO_Flow_Control_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO Flow Control is disabled. No Number of Completed Packets events will be sent from the Host Controller for SCO Connection Handles. **Default**
0x01	SCO Flow Control is enabled. Number of Completed Packets events will be sent from the Host Controller for SCO Connection Handles.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_SCO_Flow_Control_Enable command succeeded
0x01-0xFF	Write_SCO_Flow_Control_Enable command failed see Table 6.1 on page 767 for list of Error Codes

Event(s) generated (unless masked away):

When the Write_SCO_Flow_Control_Enable command has completed a Command Complete event will be generated.

4.7.40 Set_Host_Controller_To_Host_Flow_Control

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Host_Controller_To_Host_Flow_Control	0x0031	Flow_Control_Enable	Status

Description:

This command is used by the Host to turn flow control on or off for data and/or voice sent in the direction from the Host Controller to the Host. If flow control is turned off, the Host should not send the Host_Number_Of_Completed_Packets command. That command will be ignored by the Host Controller if it is sent by the Host and flow control is off. If flow control is turned on for HCI ACL Data Packets and off for HCI SCO Data Packets, Host_Number_Of_Completed_Packets commands sent by the Host should only contain Connection Handles for ACL connections. If flow control is turned off for HCI ACL Data Packets and on for HCI SCO Data Packets, Host_Number_Of_Completed_Packets commands sent by the Host should only contain Connection Handles for SCO connections. If flow control is turned on for HCI ACL Data Packets and HCI SCO Data Packets, the Host will send Host_Number_Of_Completed_Packets commands both for ACL connections and SCO connections.Note: The Flow_Control_Enable setting must only be changed if no connections exist.

Command Parameters:

Flow_Control_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	Flow control **off** in direction from Host Controller to Host. **Default.**
0x01	Flow control **on** for HCI ACL Data Packets and **off** for HCI SCO Data Packets in direction from Host Controller to Host.
0x02	Flow control **off** for HCI ACL Data Packets and **on** for HCI SCO Data Packets in direction from Host Controller to Host.
0x03	Flow control **on** both for HCI ACL Data Packets and HCI SCO Data Packets in direction from Host Controller to Host.
0x04-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Set_Host_Controller_To_Host_Flow_Control command succeeded.

0x01-0xFF	Set_Host_Controller_To_Host_Flow_Control command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Set_Host_Controller_To_Host_Flow_Control command has completed, a Command Complete event will be generated.

4.7.41 Host_Buffer_Size

Command	OCF	Command Parameters	Return Parameters
HCI_Host_Buffer_Size	0x0033	Host_ACL_Data_Packet_Length, Host_SCO_Data_Packet_Length, Host_Total_Num_ACL_Data_Packets, Host_Total_Num_SCO_Data_Packets	Status

Description:

The Host_Buffer_Size command is used by the Host to notify the Host Controller about the maximum size of the data portion of HCI ACL and SCO Data Packets sent from the Host Controller to the Host. The Host Controller will segment the data to be transmitted from the Host Controller to the Host according to these sizes, so that the HCI Data Packets will contain data with up to these sizes. The Host_Buffer_Size command also notifies the Host Controller about the total number of HCI ACL and SCO Data Packets that can be stored in the data buffers of the Host. If flow control from the Host Controller to the Host is turned off, and the Host_Buffer_Size command has not been issued by the Host, this means that the Host Controller will send HCI Data Packets to the Host with any lengths the Host Controller wants to use, and it is assumed that the data buffer sizes of the Host are unlimited. If flow control from the Host controller to the Host is turned on, the Host_Buffer_Size command must after a power-on or a reset always be sent by the Host before the first Host_Number_Of_Completed_Packets command is sent.

(The Set_Host_Controller_To_Host_Flow_Control command is used to turn flow control on or off.) The Host_ACL_Data_Packet_Length command parameter will be used to determine the size of the L2CAP segments contained in ACL Data Packets, which are transferred from the Host Controller to the Host. The Host_SCO_Data_Packet_Length command parameter is used to determine the maximum size of HCI SCO Data Packets. Both the Host and the Host Controller must support command and event packets, where the data portion (excluding header) contained in the packets is 255 bytes in size.

The Host_Total_Num_ACL_Data_Packets command parameter contains the total number of HCI ACL Data Packets that can be stored in the data buffers of the Host. The Host Controller will determine how the buffers are to be divided between different Connection Handles. The Host_Total_Num_SCO_Data_Packets command parameter gives the same information for HCI SCO Data Packets.
Note: the Host_ACL_Data_Packet_Length and Host_SCO_Data_Packet_Length command parameters do not include the length of the HCI Data Packet header.

Command Parameters:

Host_ACL_Data_Packet_Length: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Maximum length (in bytes) of the data portion of each HCI ACL Data Packet that the Host is able to accept.

Host_SCO_Data_Packet_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Maximum length (in bytes) of the data portion of each HCI SCO Data Packet that the Host is able to accept.

Host_Total_Num_ACL_Data_Packets: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Total number of HCI ACL Data Packets that can be stored in the data buffers of the Host.

Host_Total_Num_SCO_Data_Packets: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Total number of HCI SCO Data Packets that can be stored in the data buffers of the Host.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Host_Buffer_Size command succeeded.
0x01-0xFF	Host_Buffer_Size command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Host_Buffer_Size command has completed, a Command Complete event will be generated.

4.7.42 Host_Number_Of_Completed_Packets

Command	OCF	Command Parameters	Return Parameters
HCI_Host_Number_Of_ Completed_Packets	0x0035	Number_Of_Handles, Connection_ Handle[i], Host_Num_Of_Completed_Packets [i]	

Description:

The Host_Number_Of_Completed_Packets command is used by the Host to indicate to the Host Controller the number of HCI Data Packets that have been completed for each Connection Handle since the previous Host_Number_Of_ Completed_Packetscommand was sent to the Host Controller. This means that the corresponding buffer space has been freed in the Host. Based on this information, and the Host_Total_Num_ACL_Data_Packets and Host_Total_Num_ SCO_Data_Packets command parameters of the Host_Buffer_Size command, the Host Controller can determine for which Connection Handles the following HCI Data Packets should be sent to the Host. The command should only be issued by the Host if flow control in the direction from the Host Controller to the Host is on and there is at least one connection, or if the Host Controller is in local loopback mode. Otherwise, the command will be ignored by the Host Controller. While the Host has HCI Data Packets in its buffers, it must keep sending the Host_Number_Of_Completed_Packets command to the Host Controller at least periodically, until it finally reports that all buffer space in the Host used by ACL Data Packets has been freed. The rate with which this command is sent is manufacturer specific.

(The Set_Host_Controller_To_Host_Flow_Control command is used to turn flow control on or off.) If flow control from the Host controller to the Host is turned on, the Host_Buffer_Size command must after a power-on or a reset always be sent by the Host before the first Host_Number_Of_Completed_Packets command is sent.
Note: the Host_Number_Of_Completed_Packets command is a special command in the sense that no event is normally generated after the command has completed. The command may be sent at any time by the Host when there is at least one connection, or if the Host Controller is in local loopback mode independent of other commands. The normal flow control for commands is not used for the Host_Number_Of_Completed_Packets command.

Command Parameters:

Number_Of_Handles: *Size: 1 Byte*

Value	Parameter Description
0xXX	The number of Connection Handles and Host_Num_Of_Completed_Packets parameters pairs contained in this command. Range: 0-255

*Connection_Handle[i]: Size: Number_Of_Handles*2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

*Host_Num_Of_Completed_Packets [i]: Size: Number_Of_Handles * 2 Bytes*

Value	Parameter Description
N = 0xXXXX	The number of HCI Data Packets that have been completed for the associated Connection Handle since the previous time the event was returned. Range for N: 0x0000-0xFFFF

Return Parameters:

None.

Event(s) generated (unless masked away):

Normally, no event is generated after the Host_Number_Of_Completed_ Packets command has completed. However, if the Host_Number_Of_ Completed_Packets command contains one or more invalid parameters, the Host Controller will return a Command Complete event with a failure status indicating the Invalid HCI Command Parameters error code. The Host may send the Host_Number_Of_Completed_Packets command at any time when there is at least one connection, or if the Host Controller is in local loopback mode. The normal flow control for commands is not used for this command.

4.7.43 Read_Link_Supervision_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Supervision_Timeout	0x0036	Connection_Handle	Status, Connection_Handle, Link_Supervision_Timeout

Description:

This command will read the value for the Link_Supervision_Timeout parameter for the device. The Link_Supervision_Timeout parameter is used by the master or slave Bluetooth device to monitor link loss. If, for any reason, no Baseband packets are received from that Connection Handle for a duration longer than the Link_Supervision_Timeout, the connection is disconnected. The same timeout value is used for both SCO and ACL connections for the device specified by the Connection Handle.
Note: the Connection_Handle used for this command must be the ACL connection to the appropriate device. This command will set the Link_Supervision_Timeout values for other SCO Connection_Handle to that device.

Note: Setting the Link_Supervision_Timeout to No Link_Supervision_Timeout (0x0000) will disable the Link_Supervision_Timeout check for the specified Connection Handle. This makes it unnecessary for the master of the piconet to unpark and then park each Bluetooth Device every ~40 seconds. By using the No Link_Supervision_Timeout setting, the scalability of the Park mode is not limited.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value is to be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Link_Supervision_Timeout command succeeded.
0x01-0xFF	Read_Link_Supervision_Timeout command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value was read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Supervision_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xXXXX	Measured in Number of Baseband slots Link_Supervision_Timeout = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625ms - 40.9 sec

Event(s) generated (unless masked away):

When the Read_Link_Supervision_Timeout command has completed, a Command Complete event will be generated.

4.7.44 Write_Link_Supervision_Timeout

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Link_Supervision_Timeout	0x0037	Connection_Handle, Link_Supervision_Timeout	Status, Connection_Handle

Description:

This command will write the value for the Link_Supervision_Timeout parameter for the device. The Link_Supervision_Timeout parameter is used by the master or slave Bluetooth device to monitor link loss. If, for any reason, no Baseband packets are received from that Connection_Handle for a duration longer than the Link_Supervision_Timeout, the connection is disconnected. The same timeout value is used for both SCO and ACL connections for the device specified by the Connection_Handle.
Note: the Connection_Handle used for this command must be the ACL connection to the appropriate device. This command will set the Link_Supervision_Timeout values for other SCO Connection_Handle to that device.
Note: Setting the Link_Supervision_Timeout parameter to No Link_Supervision_Timeout (0x0000) will disable the Link_Supervision_Timeout check for the specified Connection Handle. This makes it unnecessary for the master of the piconet to unpark and then park each Bluetooth Device every ~40 seconds. By using the No Link_Supervision_Timeout setting, the scalability of the Park mode is not limited.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value is to be written. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Supervision_Timeout: *Size: 2 Bytes*

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xXXXX	Measured in Number of Baseband slots Link_Supervision_Timeout = N*0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625ms – 40.9 sec **Default**: N = 0x7D00 Link_Supervision_Timeout = 20 sec

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Link_Supervision_Timeout command succeeded.
0x01-0xFF	Write_Link_Supervision_Timeout command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value was written. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Link_Supervision_Timeout command has completed, a Command Complete event will be generated.

4.7.45 Read_Number_Of_Supported_IAC

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Number_Of_Supported_IAC	0x0038		Status, Num_Support_IAC

Description:

This command will read the value for the number of Inquiry Access Codes (IAC) that the local Bluetooth device can simultaneous listen for during an Inquiry Scan. All Bluetooth devices are required to support at least one IAC, the General Inquiry Access Code (the GIAC). Some Bluetooth devices support additional IACs.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Number_Of_Supported_IAC command succeeded.
0x01-0xFF	Read_Number_Of_Supported_IAC command failed. See Table 6.1 on page 766 for list of Error Codes.

Num_Support_IAC *Size: 1 Byte*

Value	Parameter Description
0xXX	Specifies the number of Supported IAC that the local Bluetooth device can simultaneous listen for during an Inquiry Scan. Range: 0x01-0x40

Event(s) generated (unless masked away):

When the Read_Number_Of_Supported_IAC command has completed, a Command Complete event will be generated.

4.7.46 Read_Current_IAC_LAP

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Current_IAC_LAP	0x0039		Status, Num_Current_IAC, IAC_LAP[i]

Description:

This command reads the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans. All Bluetooth devices are required to support at least one IAC, the General Inquiry Access Code (the GIAC). Some Bluetooth devices support additional IACs.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Current_IAC_LAP command succeeded.
0x01-0xFF	Read_Current_IAC_LAP command failed. See Table 6.1 on page 766 for list of Error Codes.

Num_Current_IAC *Size: 1 Byte*

Value	Parameter Description
0xXX	Specifies the number of IACs which are currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x01-0x40

IAC_LAP[i] *Size: 3 Bytes * Num_Current_IAC*

Value	Parameter Description
0xXXXXXX	LAPs used to create the IAC which is currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x9E8B00-0x9E8B3F

Event(s) generated (unless masked away):

When the Read_Current_IAC_LAP command has completed, a Command Complete event will be generated.

4.7.47 Write_Current_IAC_LAP

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Current_IAC_LAP	0x003A	Num_Current_IAC, IAC_LAP[i]	Status

Description:

This command writes the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans. All Bluetooth devices are required to support at least one IAC, the General Inquiry Access Code (the GIAC). Some Bluetooth devices support additional IACs.)
Note: this command writes over the current IACs used by the Bluetooth device. If the value of the NumCurrentIAC is more than the number of supported IACs, then only the first, X Inquiry Access Codes (where X equals the number of supported IACs) will be stored without any error.

Command Parameters:

Num_Current_IAC *Size: 1 Byte*

Value	Parameter Description
0xXX	Specifies the number of IACs which are currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x01-0x40

IAC_LAP[i] *Size: 3 Bytes * Num_Current_IAC*

Value	Parameter Description
0xXXXXXX	LAP(s) used to create IAC which is currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x9E8B00-0x9E8B3F. The GIAC is the default IAC to be used. If additional IACs are supported, additional default IAC will be determined by the manufacturer.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Current_IAC_LAP command succeeded.
0x01-0xFF	Write_Current_IAC_LAP command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Current_IAC_LAP command has completed, a Command Complete event will be generated.

4.7.48 Read_Page_Scan_Period_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Period_Mode	0x003B		Status, Page_Scan_Period_Mode

Description:

This command is used to read the mandatory Page_Scan_Period_Mode of the local Bluetooth device. Every time an inquiry response message is sent, the Bluetooth device will start a timer (T_mandatory_pscan), the value of which is dependent on the Page_Scan_Period_Mode. As long as this timer has not expired, the Bluetooth device will use the Page_Scan_Period_Mode for all following page scans.
Note: the timer T_mandatory_pscan will be reset at each new inquiry response. For details see the "Baseband Specification" on page 33. (Keyword: SP-Mode, FHS-Packet, T_mandatory_pscan, Inquiry-Response).

After transmitting one or more inquiry response (FHS) packets as a result of the inquiry scan process, the local Bluetooth device is allowed to enter the page scan state using mandatory page scan mode regardless of the setting of the Scan_Enable parameter.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Page_Scan_Period_Mode command succeeded.
0x01-0xFF	Read_Page_Scan_Period_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Page_Scan_Period_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	P0
0x01	P1
0x02	P2
0x03-0xFF	Reserved.

Event(s) generated (unless masked away):

When the Read_Page_Scan_Period_Mode command has completed, a Command Complete event will be generated.

4.7.49 Write_Page_Scan_Period_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Period_Mode	0x003C	Page_Scan_Period_Mode	Status

Description:

This command is used to write the mandatory Page_Scan_Period_Mode of the local Bluetooth device. Every time an inquiry response message is sent, the Bluetooth device will start a timer (T_mandatory_pscan), the value of which is dependent on the Page_Scan_Period_Mode. As long as this timer has not expired, the Bluetooth device will use the Page_Scan_Period_Mode for all following page scans.
Note: the timer T_mandatory_pscan will be reset at each new inquiry response. For details see the "Baseband Specification" on page 33. (Keyword: SP-Mode, FHS-Packet, T_mandatory_pscan, Inquiry-Response).

After transmitting one or more inquiry response (FHS) packets as a result of the inquiry scan process, the local Bluetooth device is allowed to enter the page scan state using mandatory page scan mode regardless of the setting of the Scan_Enable parameter.

Command Parameters:

Page_Scan_Period_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	P0. **Default.**
0x01	P1
0x02	P2
0x03-0xFF	Reserved.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Page_Scan_Period_Mode command succeeded.
0x01-0xFF	Write_Page_Scan_Period_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Period_Mode command has completed, a Command Complete event will be generated.

4.7.50 Read_Page_Scan_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Mode	0x003D		Status, Page_Scan_Mode

Description:

This command is used to read the default page scan mode of the local Bluetooth device. The Page_Scan_Mode parameter indicates the page scan mode that is used for default page scan. Currently one mandatory page scan mode and three optional page scan modes are defined. Following an inquiry response, if the Baseband timer T_mandatory_pscan has not expired, the mandatory page scan mode must be applied. For details see the "Baseband Specification" on pa g e33(Keyword: Page-Scan-Mode, FHS-Packet, T_mandatory_pscan)

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Page_Scan_Mode command succeeded.
0x01-0xFF	Read_Page_Scan_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode
0x01	Optional Page Scan Mode I
0x02	Optional Page Scan Mode II
0x03	Optional Page Scan Mode III
0x04-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Page_Scan_Mode command has completed, a Command Complete event will be generated.

4.7.51 Write_Page_Scan_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Mode	0x003E	Page_Scan_Mode	Status

Description:

This command is used to write the default page scan mode of the local Bluetooth device. The Page_Scan_Mode parameter indicates the page scan mode that is used for the default page scan. Currently, one mandatory page scan mode and three optional page scan modes are defined. Following an inquiry response, if the Baseband timer T_mandatory_pscan has not expired, the mandatory page scan mode must be applied. For details see the "Baseband Specification" on page 38 (Keyword: Page-Scan-Mode, FHS-Packet, T_mandatory_pscan).

Command Parameters:

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode. **Default**.
0x01	Optional Page Scan Mode I
0x02	Optional Page Scan Mode II
0x03	Optional Page Scan Mode III
0x04-0xFF	Reserved.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Page_Scan_Mode command succeeded.
0x01-0xFF	Write_Page_Scan_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Mode command has completed, a Command Complete event will be generated.

4.8 INFORMATIONAL PARAMETERS

The Informational Parameters are fixed by the manufacturer of the Bluetooth hardware. These parameters provide information about the Bluetooth device and the capabilities of the Host Controller, Link Manager, and Baseband. The host device cannot modify any of these parameters. For Informational Parameters Commands, the OGF is defined as 0x04

Command	Command Summary Description
Read_Local_Version_Information	The Read_Local_Version_Information command will read the values for the version information for the local Bluetooth device.
Read_Local_Supported_Features	The Read_Local_Supported_Features command requests a list of the supported features for the local device.
Read_Buffer_Size	The Read_Buffer_Size command returns the size of the HCI buffers. These buffers are used by the Host Controller to buffer data that is to be transmitted.
Read_Country_Code	The Read_Country_Code command will read the value for the Country Code status parameter. The Country Code defines which range of frequency band of the ISM 2.4 GHz band will be used by the device.
Read_BD_ADDR	The Read_BD_ADDR command will read the value for the BD_ADDR parameter. The BD_ADDR is a 48-bit unique identifier for a Bluetooth device.

4.8.1 Read_Local_Version_Information

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Version_ Information	0x0001		Status, HCI Version, HCI Revision, LMP Version, Manufacturer_Name, LMP Subversion

Description:

This command will read the values for the version information for the local Bluetooth device. The version information consists of two parameters: the version and revision parameters.

The version parameter defines the major hardware version of the Bluetooth hardware. The version parameter only changes when new versions of the Bluetooth hardware are produced for new Bluetooth SIG specifications. The version parameter is controlled by the SIG.

The revision parameter should be controlled by the manufacturer and should be changed as needed. The Manufacturer_Name parameter indicates the manufacturer of the local Bluetooth module as specified by the LMP definition of this parameter. The subversion parameter should be controlled by the manufacturer and should be changed as needed. The subversion parameter defines the various revisions that each version of the Bluetooth hardware will go through as design processes change and errors are fixed. This allows the software to determine what Bluetooth hardware is being used, and to work around various bugs in the hardware if necessary.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Local_Version_Information command succeeded.
0x01-0xFF	Read_Local_Version_Information command failed. See Table 6.1 on page 766 for list of Error Codes.

HCI_Version: *Size: 1 Byte*

Value	Parameter Description
0x00	Bluetooth HCI Specification 1.0B.
0x01	Bluetooth HCI Specification 1.1.
0x02-0xfFF	Reserved for future use.

HCI_Revision: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Revision of the Current HCI in the Bluetooth hardware.

LMP_Version: *Size: 1 Byte*

Value	Parameter Description
0xXX	Version of the Current LMP in the Bluetooth Hardware, see Table 5.2 on page 239 in the Link Manager Protocol for assigned values (VersNr).

Manufacturer_Name: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Manufacturer Name of the Bluetooth Hardware, see Table 5.2 on page 239 in the Link Manager Protocol for assigned values (CompId).

LMP_Subversion: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Subversion of the Current LMP in the Bluetooth Hardware, see Table 5.2 on page 239 in the Link Manager Protocol for assigned values (SubVersNr).

Event(s) generated (unless masked away):

When the Read_Local_Version_Information command has completed, a Command Complete event will be generated.

4.8.2 Read_Local_Supported_Features

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Supported_Features	0x0003		Status, LMP_Features

Description:

This command requests a list of the supported features for the local device. This command will return a list of the LMP features. For details see "Link Manager Protocol" on page183 .

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Local_Supported_Features command succeeded.
0x01-0xFF	Read_Local_Supported_Features command failed. See Table 6.1 on page 766 for list of Error Codes.

LMP_Features: *Size: 8 Bytes*

Value	Parameter Description
0xXXXXXXXX XXXXXXXX	Bit Mask List of LMP features. For details see "Link Manager Protocol" on page 183

Event(s) generated (unless masked away):

When the Read_Local_Supported_Features command has completed, a Command Complete event will be generated.

4.8.3 Read_Buffer_Size

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Buffer_Size	0x0005		Status, HC_ACL_Data_Packet_Length, HC_SCO_Data_Packet_Length, HC_ Total_Num_ACL_Data_Packets, HC_Total_Num_SCO_Data_Packets

Description:

The Read_Buffer_Size command is used to read the maximum size of the data portion of HCI ACL and SCO Data Packets sent from the Host to the Host Controller. The Host will segment the data to be transmitted from the Host to the Host Controller according to these sizes, so that the HCI Data Packets will contain data with up to these sizes. The Read_Buffer_Size command also returns the total number of HCI ACL and SCO Data Packets that can be stored in the data buffers of the Host Controller. The Read_Buffer_Size command must be issued by the Host before it sends any data to the Host Controller.

The HC_ACL_Data_Packet_Length return parameter will be used to determine the size of the L2CAP segments contained in ACL Data Packets, which are transferred from the Host to the Host Controller to be broken up into baseband packets by the Link Manager. The HC_SCO_Data_Packet_Length return parameter is used to determine the maximum size of HCI SCO Data Packets. Both the Host and the Host Controller must support command and event packets, where the data portion (excluding header) contained in the packets is 255 bytes in size. The HC_Total_Num_ACL_Data_Packets return parameter contains the total number of HCI ACL Data Packets that can be stored in the data buffers of the Host Controller. The Host will determine how the buffers are to be divided between different Connection Handles. The
HC_Total_Num_SCO_Data_
Packets return parameter gives the same information but for HCI SCO Data Packets.
Note: the HC_ACL_Data_Packet_Length and HC_SCO_Data_Packet_Length return parameters do not include the length of the HCI Data Packet header.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Buffer_Size command succeeded.
0x01-0xFF	Read_Buffer_Size command failed. See Table 6.1 on page 766 for list of Error Codes.

HC_ACL_Data_Packet_Length: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Maximum length (in bytes) of the data portion of each HCI ACL Data Packet that the Host Controller is able to accept.

HC_SCO_Data_Packet_Length: *Size: 1 Byte*

Value	Parameter Description
0xXX	Maximum length (in bytes) of the data portion of each HCI SCO Data Packet that the Host Controller is able to accept.

HC_Total_Num_ACL_Data_Packets: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Total number of HCI ACL Data Packets that can be stored in the data buffers of the Host Controller.

HC_Total_Num_SCO_Data_Packets: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Total number of HCI SCO Data Packets that can be stored in the data buffers of the Host Controller.

Event(s) generated (unless masked away):

When the Read_Buffer_Size command has completed, a Command Complete event will be generated.

4.8.4 Read_Country_Code

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Country_Code	0x0007		Status, Country_Code

Description:

This command will read the value for the Country_Code return parameter. The Country_Code defines which range of frequency band of the ISM 2.4 GHz band will be used by the device. Each country has local regulatory bodies regulating which ISM 2.4 GHz frequency ranges can be used.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Country_Code command succeeded.
0x01-0xFF	Read_Country_Code command failed. See Table 6.1 on page 766 for list of Error Codes.

Country_Code: *Size: 1 Byte*

Value	Parameter Description
0x00	North America & Europe* and Japan
0x01	France
0x04-FF	Reserved for Future Use.

*. Except France

Event(s) generated (unless masked away):

When the Read_Country_Code command has completed, a Command Complete event will be generated.

4.8.5 Read_BD_ADDR

Command	OCF	Command Parameters	Return Parameters
HCI_Read_BD_ADDR	0x0009		Status, BD_ADDR

Description:

This command will read the value for the BD_ADDR parameter.
The BD_ADDR is a 48-bit unique identifier for a Bluetooth device. See the "Baseband Specification" on page33 for details of how BD_ADDR is used.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_BD_ADDR command succeeded.
0x01-0xFF	Read_BD_ADDR command failed. See Table 6.1 on page 766 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device

Event(s) generated (unless masked away):

When the Read_BD_ADDR command has completed, a Command Complete event will be generated.

4.9 STATUS PARAMETERS

The Host Controller modifies all status parameters. These parameters provide information about the current state of the Host Controller, Link Manager, and Baseband. The host device cannot modify any of these parameters other than to reset certain specific parameters. For the Status and baseband, the OGF is defined as 0x05

Command	Command Summary Description
Read_Failed_Contact_Counter	The Read_Failed_Contact_Counter will read the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Failed_Contact_Counter records the number of consecutive incidents in which either the slave or master didn't respond after the flush timeout had expired, and the L2CAP packet that was currently being transmitted was automatically 'flushed'.
Reset_Failed_Contact_Counter	The Reset_Failed_Contact_Counter will reset the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Failed_Contact_Counter records the number of consecutive incidents in which either the slave or master didn't respond after the flush timeout had expired and the L2CAP packet that was currently being transmitted was automatically 'flushed'.
Get_Link_Quality	The Get_Link_Quality command will read the value for the Link_Quality for the specified Connection Handle.
Read_RSSI	The Read_RSSI command will read the value for the Received Signal Strength Indication (RSSI) for a connection handle to another Bluetooth device.

4.9.1 Read_Failed_Contact_Counter

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Failed_ Contact_Counter	0x0001	Connection_Handle	Status, Connection_Handle, Failed_Contact_Counter

Description:

This command will read the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Connection_Handle must be a Connection_Handle for an ACL connection. The Failed_Contact_Counter records the number of consecutive incidents in which either the slave or master didn't respond after the flush timeout had expired, and the L2CAP packet that was currently being transmitted was automatically 'flushed'. When this occurs, the Failed_Contact_Counter is incremented by 1. The Failed_Contact_ Counter for a connection is reset to zero on the following conditions:

1. When a new connection is established
2. When the Failed_Contact_Counter is > zero and an L2CAP packet is acknowledged for that connection
3. When the Reset_Failed_Contact_Counter command has been issued

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter should be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Failed_Contact_Counter command succeeded.
0x01-0xFF	Read_Failed_Contact_Counter command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Failed_Contact_Counter: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Number of consecutive failed contacts for a connection corresponding to the connection handle.

Event(s) generated (unless masked away):

When the Read_Failed_Contact_Counter command has completed, a Command Complete event will be generated.

4.9.2 Reset_Failed_Contact_Counter

Command	OCF	Command Parameters	Return Parameters
HCI_Reset_Failed_ Contact_Counter	0x0002	Connection_Handle	Status, Connection_Handle

Description:

This command will reset the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Connection_Handle must be a Connection_Handle for an ACL connection. The Failed_Contact_Counter records the number of consecutive incidents in which either the slave or master didn't respond after the flush timeout had expired, and the L2CAP packet that was currently being transmitted was automatically 'flushed'. When this occurs, the Failed_Contact_Counter is incremented by 1.The Failed_Contact_ Counter for a connection is reset to zero on the following conditions:

1. When a new connection is established
2. When the Failed_Contact_Counter is > zero and an L2CAP packet is acknowledged for that connection
3. When the Reset_Failed_Contact_Counter command has been issued

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter should be reset. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Reset_Failed_Contact_Counter command succeeded.
0x01-0xFF	Reset_Failed_Contact_Counter command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter has been reset. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Reset_Failed_Contact_Counter command has completed, a Command
Complete event will be generated.

4.9.3 Get_Link_Quality

Command	OCF	Command Parameters	Return Parameters
HCI_Get_Link_Quality	0x0003	Connection_Handle	Status, Connection_Handle, Link_Quality

Description:

This command will return the value for the Link_Quality for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection. This command will return a Link_Quality value from 0-255, which represents the quality of the link between two Bluetooth devices. The higher the value, the better the link quality is. Each Bluetooth module vendor will determine how to measure the link quality.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the connection for which link quality parameters are to be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Get_Link_Quality command succeeded.
0x01-0xFF	Get_Link_Quality command failed.See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the connection for which the link quality parameter has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Quality: *Size: 1 Byte*

Value	Parameter Description
0xXX	The current quality of the Link connection between the local device and the remote device specified by the Connection Handle Range: 0x00 – 0xFF The higher the value, the better the link quality is.

Event(s) generated (unless masked away):

When the Get_Link_Quality command has completed, a Command Complete event will be generated.

4.9.4 Read_RSSI

Command	OCF	Command Parameters	Return Parameters
HCI_Read_RSSI	0x0005	Connection_Handle	Status, Connection_Handle,RSSI

Description:

This command will read the value for the difference between the measured Received Signal Strength Indication (RSSI) and the limits of the Golden Receive Power Range (see Radio Specification Section 4.7 on page 27) for a connection handle to another Bluetooth device. The Connection_Handle must be a Connection_Handle for an ACL connection. Any positive RSSI value returned by the Host Controller indicates how many dB the RSSI is above the upper limit, any negative value indicates how many dB the RSSI is below the lower limit. The value zero indicates that the RSSI is inside the Golden Receive Power Range.
Note: how accurate the dB values will be depends on the Bluetooth hardware. The only requirements for the hardware are that the Bluetooth device is able to tell whether the RSSI is inside, above or below the Golden Device Power Range.

Command Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the RSSI is to be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_RSSI command succeeded.
0x01-0xFF	Read_RSSI command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the RSSI has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

RSSI: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	Size: 1 Byte (signed integer) Range: -128 ≤ N ≤ 127 Units: dB

Event(s) generated (unless masked away):

When the Read_RSSI command has completed, a Command Complete event will be generated.

4.10 TESTING COMMANDS

The Testing commands are used to provide the ability to test various functionalities of the Bluetooth hardware. These commands provide the ability to arrange various conditions for testing. For the Testing Commands, the OGF is defined as 0x06

Command	Command Summary Description
Read_Loopback_Mode	The Read_Loopback_Mode will read the value for the setting of the Host Controllers Loopback Mode. The setting of the Loopback Mode will determine the path of information.
Write_Loopback_Mode	The Write_Loopback_Mode will write the value for the setting of the Host Controllers Loopback Mode. The setting of the Loopback Mode will determine the path of information.
Enable_Device_Under_Test_Mode	The Enable_Device_Under_Test_Mode command will allow the local Bluetooth module to enter test mode via LMP test commands. The Host issues this command when it wants the local device to be the DUT for the Testing scenarios as described in the Bluetooth Test Mode document.

4.10.1 Read_Loopback_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Loopback_Mode	0x0001		Status, Loopback_Mode

Description:

This command will read the value for the setting of the Host Controller's Loopback Mode. The setting of the Loopback Mode will determine the path of information. In Non-testing Mode operation, the Loopback Mode is set to Non-testing Mode and the path of the information is as specified by the Bluetooth specifications. In Local Loopback Mode, every Data Packet (ACL and SCO) and Command Packet that is sent from the Host to the Host Controller is sent back with no modifications by the Host Controller, as shown in Fig. 4.5 on page 718.

When the Bluetooth Host Controller enters Local Loopback Mode, it shall respond with four Connection Complete events, one for an ACL channel and three for SCO channels, so that the Host gets connection handles to use when sending ACL and SCO data. When in Local Loopback Mode the Host Controller loops back commands and data to the Host. The Loopback Command event is used to loop back commands that the Host sends to the Host Controller.

There are some commands that are not looped back in Local Loopback Mode: Reset, Set_Host_Controller_To_Host_Flow_Control, Host_Buffer_Size, Host_Number_Of_Completed_Packets, Read_Buffer_Size, Read_Loopback_Mode and Write_Loopback_Mode.These commands should be executed in the way they are normally executed. The commands Reset and Write_Loopback_Mode can be used to exit local loopback mode. If Write_Loopback_Mode is used to exit Local Loopback Mode, four Disconnection Complete events should be sent to the Host, corresponding to the Connection Complete events that were sent when entering Local Loopback Mode. Furthermore, no connections are allowed in Local Loopback mode. If there is a connection and there is an attempt to set the device to Local Loopback Mode, the attempt will be refused. When the device is in Local Loopback Mode, the Host Controller will refuse incoming connection attempts. This allows the Host Controller Transport Layer to be tested without any other variables.

If a device is set to Remote Loopback Mode, it will send back all data (ACL and SCO) that comes over the air, and it will only allow a maximum of one ACL connection and three SCO connections – and these should be all to the same remote device. If there already are connections to more than one remote device and there is an attempt to set the local device to Remote Loopback Mode, the attempt will be refused. See Fig. 4.6 on pag e718 where the rightmost device is set to Remote Loopback Mode and the leftmost device is set to

Non-testing Mode. This allows the Bluetooth Air link to be tested without any other variables.

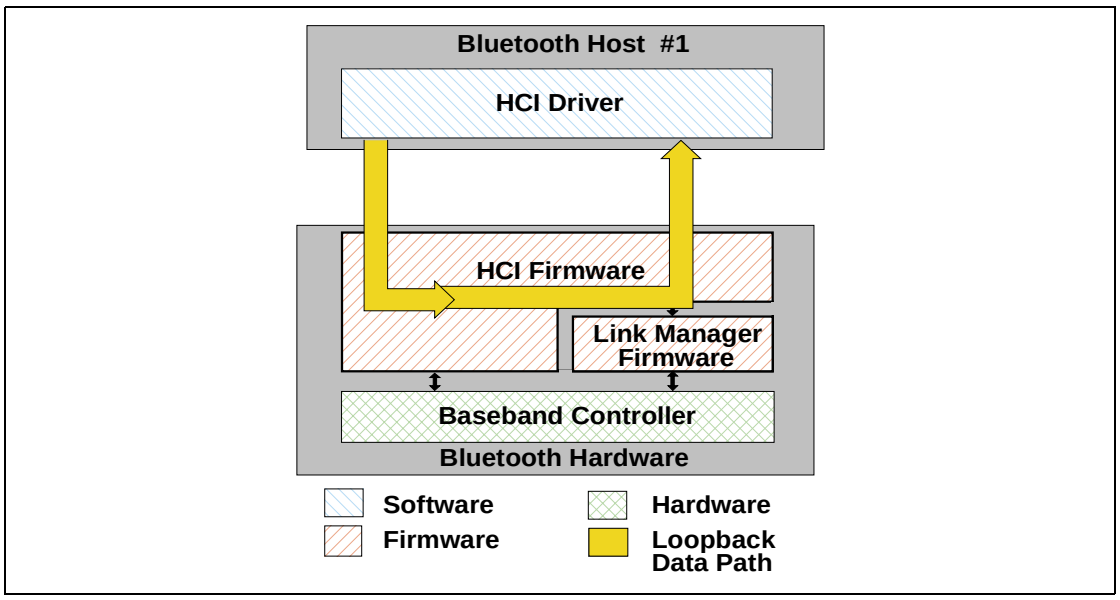


Figure 4.5: Local Loopback Mode

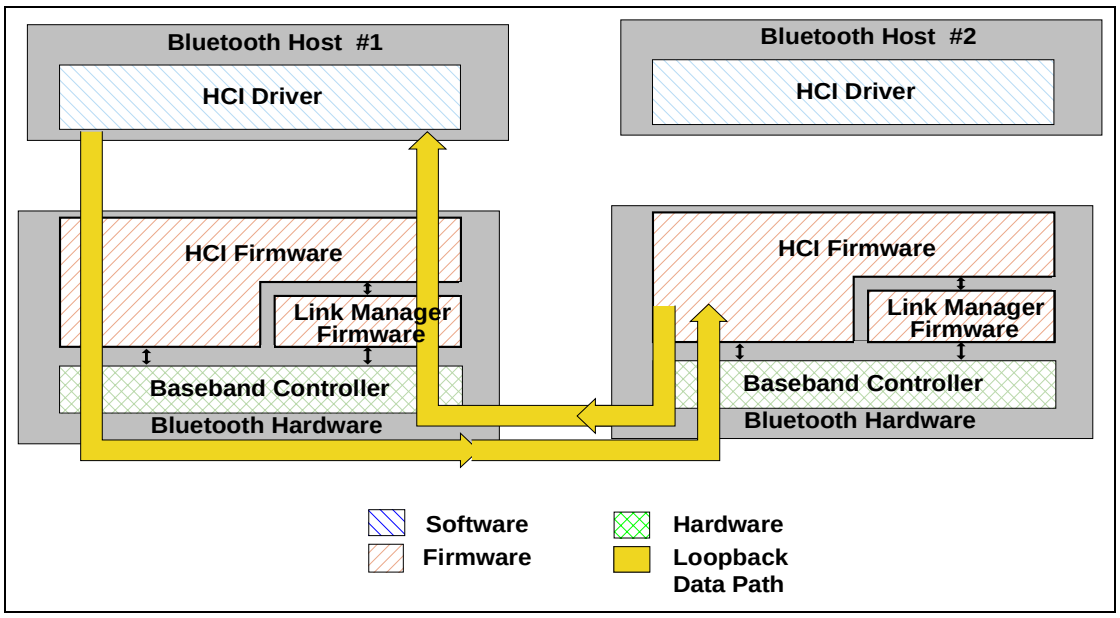


Figure 4.6: Remote Loopback Mode

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Loopback_Mode command succeeded.
0x01-0xFF	Read_Loopback_Mode command failed. See Table 2 on page 260 for list of Error Codes.

Loopback_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	No Loopback mode enabled. **Default.**
0x01	Enable Local Loopback.
0x02	Enable Remote Loopback.
0x03-0xFF	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Loopback_Mode command has completed, a Command Complete event will be generated.

4.10.2 Write_Loopback_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Loopback_Mode	0x0002	Loopback_Mode	Status

Description:

This command will write the value for the setting of the Host Controller's Loopback Mode. The setting of the Loopback Mode will determine the path of information. In Non-testing Mode operation, the Loopback Mode is set to Non-testing Mode and the path of the information as specified by the Bluetooth specifications. In Local Loopback Mode, every Data Packet (ACL and SCO) and Command Packet that is sent from the Host to the Host Controller is sent back with no modifications by the Host Controller, as shown in Fig. 4.7 on page 721.

When the Bluetooth Host Controller enters Local Loopback Mode, it shall respond with four Connection Complete events, one for an ACL channel and three for SCO channels, so that the Host gets connection handles to use when sending ACL and SCO data. When in Local Loopback Mode, the Host Controller loops back commands and data to the Host. The Loopback Command event is used to loop back commands that the Host sends to the Host Controller.

There are some commands that are not looped back in Local Loopback Mode: Reset, Set_Host_Controller_To_Host_Flow_Control, Host_Buffer_Size, Host_Number_Of_Completed_Packets, Read_Buffer_Size, Read_Loopback_Mode and Write_Loopback_Mode. These commands should be executed in the way they are normally executed. The commands Reset and Write_Loopback_Mode can be used to exit local loopback mode.

If Write_Loopback_Mode is used to exit Local Loopback Mode, four Disconnection Complete events should be sent to the Host corresponding to the Connection Complete events that were sent when entering Local Loopback Mode. Furthermore, no connections are allowed in Local Loopback mode. If there is a connection, and there is an attempt to set the device to Local Loopback Mode, the attempt will be refused. When the device is in Local Loopback Mode, the Host Controller will refuse incoming connection attempts. This allows the Host Controller Transport Layer to be tested without any other variables.

If a device is set to Remote Loopback Mode, it will send back all data (ACL and SCO) that comes over the air. It will only allow a maximum of one ACL connection and three SCO connections, and these should all be to the same remote device. If there already are connections to more than one remote device and there is an attempt to set the local device to Remote Loopback Mode, the attempt will be refused.

See Fig. 4.8 on page721 , where the rightmost device is set to Remote Loopback Mode and the leftmost device is set to Non-testing Mode. This allows the Bluetooth Air link to be tested without any other variables.

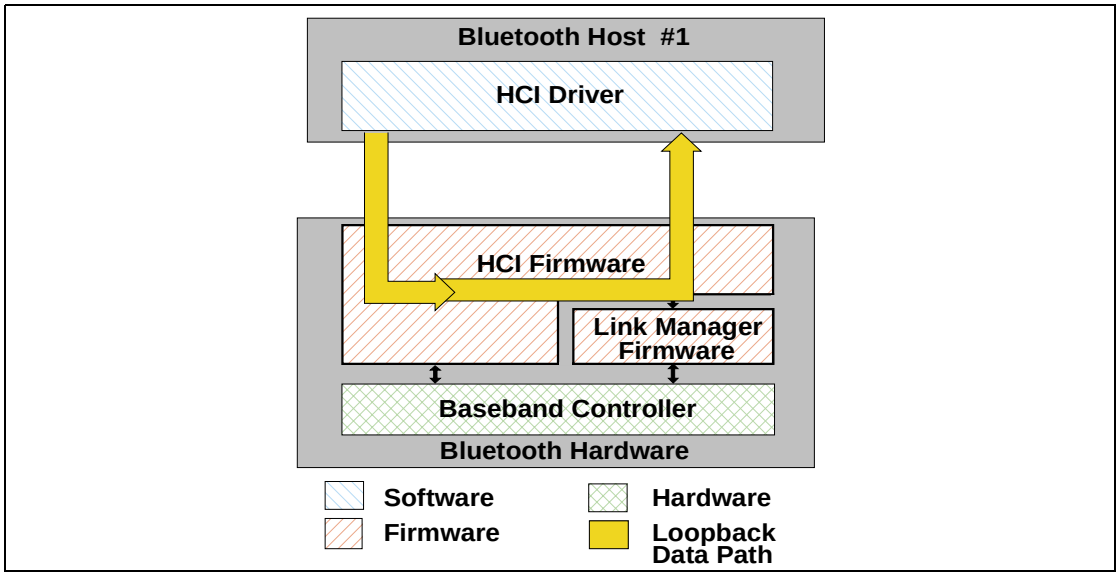


Figure 4.7: Local Loopback Mode

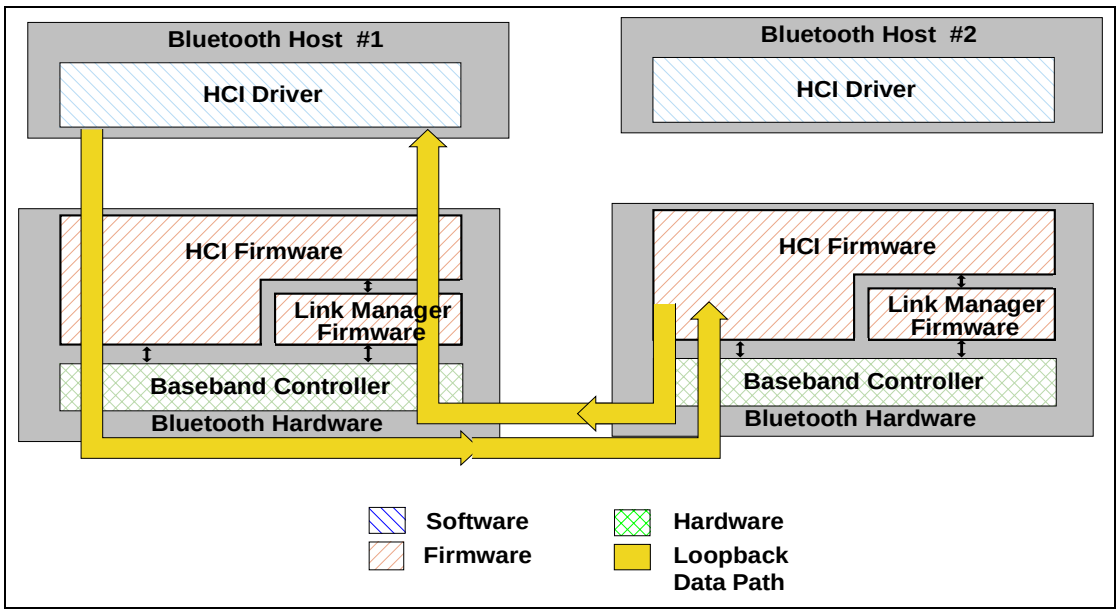


Figure 4.8: Remote Loopback Mode

Command Parameters:

Loopback_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	No Loopback mode enabled. **Default.**
0x01	Enable Local Loopback.
0x02	Enable Remote Loopback.
0x03-0xFF	Reserved for Future Use.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Write_Loopback_Mode command succeeded.
0x01-0xFF	Write_Loopback_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Loopback_Mode command has completed, a Command Complete event will be generated.

4.10.3 Enable_Device_Under_Test_Mode

Command	OCF	Command Parameters	Return Parameters
HCI_Enable_Device_Under_ Test_Mode	0x0003		Status

Description:

The Enable_Device_Under_Test_Mode command will allow the local Bluetooth module to enter test mode via LMP test commands. For details see "Link Manager Protocol" on page 183. The Host issues this command when it wants the local device to be the DUT for the Testing scenarios as described in the "Bluetooth Test Mode" on page827 . When the Host Controller receives this command, it will complete the command with a Command Complete event. The Host Controller functions as normal until the remote tester issues the LMP test command to place the local device into Device Under Test mode. To disable and exit the Device Under Test Mode, the Host can issue the HCI_Reset command. This command prevents remote Bluetooth devices from causing the local Bluetooth device to enter test mode without first issuing this command.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Enter_Device_Under_Test_Mode command succeeded.
0x01-0xFF	Enter_Device_Under_Test_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Event(s) generated (unless masked away):

When the Enter_Device_Under_Test_Mode command has completed, a Command Complete event will be generated.

5 EVENTS

5.1 EVENT

In addition to the events listed below, event code 0xFF is reserved for the event code used for vendor-specific debug events, and event code 0xFE is reserved for Bluetooth Logo Testing.

Event	Event Summary Description
Inquiry Complete event	The Inquiry Complete event indicates that the Inquiry is finished.
Inquiry Result event	The Inquiry Result event indicates that a Bluetooth device or multiple Bluetooth devices have responded so far during the current Inquiry process.
Connection Complete event	The Connection Complete event indicates to both of the Hosts forming the connection that a new connection has been established.
Connection Request event	The Connection Request event is used to indicate that a new incoming connection is trying to be established.
Disconnection Complete event	The Disconnection Complete event occurs when a connection has been terminated.
Authentication Complete event	The Authentication Complete event occurs when authentication has been completed for the specified connection.
Remote Name Request Complete event	The Remote Name Request Complete event is used to indicate a remote name request has been completed. The Remote_Name event parameter is a UTF-8 encoded string with up to 248 bytes in length.
Encryption Change event	The Encryption Change event is used to indicate that the change in the encryption has been completed for the Connection Handle specified by the Connection_Handle event parameter.
Change Connection Link Key Complete event	The Change Connection Link Key Complete event is used to indicate that the change in the Link Key for the Connection Handle specified by the Connection_Handle event parameter had been completed.
Master Link Key Complete event	The Master Link Key Complete event is used to indicate that the change in the temporary Link Key or in the semi-permanent link keys on the Bluetooth master side has been completed.
Read Remote Supported Features Complete event	The Read Remote Supported Features Complete event is used to indicate the completion of the process of the Link Manager obtaining the supported features of the remote Bluetooth device specified by the Connection_Handle event parameter.

Table 5.1: List of Supported Events

Event	Event Summary Description
Read Remote Version Information Complete event	The Read Remote Version Information Complete event is used to indicate the completion of the process of the Link Manager obtaining the version information of the remote Bluetooth device specified by the Connection_Handle event parameter.
QoS Setup Complete event	The QoS Setup Complete event is used to indicate the completion of the process of the Link Manager setting up QoS with the remote Bluetooth device specified by the Connection_Handle event parameter.
Command Complete event	The Command Complete event is used by the Host Controller to pass the return status of a command and the other event parameters for each HCI Command.
Command Status event	The Command Status event is used to indicate that the command described by the Command_Opcode parameter has been received and the Host Controller is currently performing the task for this command.
Hardware Error event	The Error event is used to indicate some type of hardware failure for the Bluetooth device.
Flush Occurred event	The Flush Occurred event is used to indicate that, for the specified Connection Handle, the current user data to be transmitted has been removed.
Role Change event	The Role Change event is used to indicate that the current Bluetooth role related to the particular connection has been changed.
Number Of Completed Packets event	The Number Of Completed Packets event is used by the Host Controller to indicate to the Host how many HCI Data Packets have been completed for each Connection Handle since the previous Number Of Completed Packets event was sent.
Mode Change event	The Mode Change event is used to indicate when the device associated with the Connection Handle changes between Active, Hold, Sniff and Park mode.
Return Link Keys event	The Return Link Keys event is used to return stored link keys after a Read_Stored_Link_Key command is used.
PIN Code Request event	The PIN Code Request event is used to indicate that a PIN code is required to create a new link key for a connection.
Link Key Request event	The Link Key Request event is used to indicate that a Link Key is required for the connection with the device specified in BD_ADDR.
Link Key Notification event	The Link Key Notification event is used to indicate to the Host that a new Link Key has been created for the connection with the device specified in BD_ADDR.
Loopback Command event	The Loopback Command event is used to loop back most commands that the Host sends to the Host Controller.

Table 5.1: List of Supported Events

Event	Event Summary Description
Data Buffer Overflow event	The Data Buffer Overflow event is used to indicate that the Host Controller's data buffers have overflowed, because the Host has sent more packets than allowed.
Max Slots Change event	This event is used to notify the Host about the LMP_Max_Slots parameter when the value of this parameter changes.
Read Clock Offset Complete event	The Read Clock Offset Complete event is used to indicate the completion of the process of the LM obtaining the Clock offset information.
Connection Packet Type Changed event	The Connection Packet Type Changed event is used to indicate the completion of the process of the Link Manager changing the Packet Types used for the specified Connection_Handle.
QoS Violation event	The QoS Violation event is used to indicate the Link Manager is unable to provide the current QoS requirement for the Connection Handle.
Page Scan Mode Change event	The Page Scan Mode Change event indicates that the connected remote Bluetooth device with the specified Connection_Handle has successfully changed the Page_Scan_Mode.
Page Scan Repetition Mode Change event	The Page Scan Repetition Mode Change event indicates that the connected remote Bluetooth device with the specified Connection_Handle has successfully changed the Page_Scan_Repetition_Mode (SR).

Table 5.1: List of Supported Events

5.2 POSSIBLE EVENTS

The events provide a method to return parameters and data associated for each event.

5.2.1 Inquiry Complete event

Event	Event Code	Event Parameters
Inquiry Complete	0x01	Status

Description:

The Inquiry Complete event indicates that the Inquiry is finished. This event contains a status parameter, which is used to indicate if the Inquiry completed successfully or if the Inquiry was not completed.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Inquiry command completed successfully.
0x01-0xFF	Inquiry command failed. See Table 6.1 on page 766 for list of Error Codes.

5.2.2 Inquiry Result event

Event	Event Code	Event Parameters
Inquiry Result	0x02	Num_Responses, BD_ADDR[i], Page_Scan_Repetition_Mode[i], Page_Scan_Period_Mode[i], Page_Scan_Mode[i], Class_of_Device[i] Clock_Offset[i]

Description:

The Inquiry Result event indicates that a Bluetooth device or multiple Bluetooth devices have responded so far during the current Inquiry process. This event will be sent from the Host Controller to the Host as soon as an Inquiry Response from a remote device is received if the remote device supports only mandatory paging scheme. The Host Controller may queue these Inquiry Responses and send multiple Bluetooth devices information in one Inquiry Result event. The event can be used to return one or more Inquiry responses in one event. This event contains the BD_ADDR, Page_Scan_Repetition_Mode, Page_Scan_Period_Mode, Page_Scan_Mode, Clock_Offset, and the Class of Device for each Bluetooth device that responded to the latest inquiry.

Event Parameters:

Num_Responses: *Size: 1 Byte*

Value	Parameter Description
0xXX	Number of responses from the Inquiry.

BD_ADDR[i]: *Size: 6 Bytes * Num_Responses*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for each device which responded.

Page_Scan_Repetition_Mode[i]: *Size: 1 Byte* Num_Responses*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved

Page_Scan_Period_Mode[i]: *Size: 1 Byte* Num_Responses*

Value	Parameter Description
0x00	P0
0x01	P1
0x02	P2
0x03 – 0xFF	Reserved

Page_Scan_Mode[i]: *Size: 1 Byte* Num_Responses*

Value	Parameter Description
0x00	Mandatory Page Scan Mode
0x01	Optional Page Scan Mode I
0x02	Optional Page Scan Mode II
0x03	Optional Page Scan Mode III
0x04 – 0xFF	Reserved

Class_of_Device[i]: *Size: 3 Bytes * Num_Responses*

Value	Parameter Description
0xXXXXXX	Class of Device for the device

Clock_Offset[i]: *Size: 2 Bytes * Num_Responses*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Reserved

5.2.3 Connection Complete event

Event	Event Code	Event Parameters
Connection Complete	0x03	Status, Connection_Handle, BD_ADDR, Link_Type, Encryption_Mode

Description:

The Connection Complete event indicates to both of the Hosts forming the connection that a new connection has been established. This event also indicates to the Host, which issued the Create Connection, Add_SCO_Connection, or Accept_Connection_Request or Reject_Connection_Request command and then received a Command Status event, if the issued command failed or was successful.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Connection successfully completed.
0x01-0xFF	Connection failed to Complete. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection between to Bluetooth devices. The Connection Handle is used as an identifier for transmitting and receiving voice or data. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the other connected Device forming the connection.

Link_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO connection (Voice Channels).
0x01	ACL connection (Data Channels).
0x02-0xFF	Reserved for Future Use.

Encryption_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Encryption disabled.
0x01	Encryption only for point-to-point packets.
0x02	Encryption for both point-to-point and broadcast packets.
0x03-0xFF	Reserved.

5.2.4 Connection Request event

Event	Event Code	Event Parameters
Connection Request	0x04	BD_ADDR, Class_of_Device, Link_Type

Description:

The Connection Request event is used to indicate that a new incoming connection is trying to be established. The connection may either be accepted or rejected. If this event is masked away and there is an incoming connection attempt and the Host Controller is not set to auto-accept this connection attempt, the Host Controller will automatically refuse the connection attempt. When the Host receives this event, it should respond with either an Accept_Connection_Request or Reject_Connection_Request command before the timer Conn_Accept_Timeout expires.

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the device that requests the connection.

Class_of_Device: *Size: 3 Bytes*

Value	Parameter Description
0xXXXXXX	Class of Device for the device, which request the connection.

Link_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO connection requested (Voice Channels).
0x01	ACL connection requested (Data Channels).
0x02-0xFF	Reserved for Future Use.

5.2.5 Disconnection Complete event

Event	Event Code	Event Parameters
Disconnection Complete	0x05	Status, Connection_Handle, Reason

Description:

The Disconnection Complete event occurs when a connection is terminated. The status parameter indicates if the disconnection was successful or not. The reason parameter indicates the reason for the disconnection if the disconnection was successful. If the disconnection was not successful, the value of the reason parameter can be ignored by the Host. For example, this can be the case if the Host has issued the Disconnect command and there was a parameter error, or the command was not presently allowed, or a connection handle that didn't correspond to a connection was given.
Note: When a physical link fails, one Disconnection Complete event will be returned for each logical channel on the physical link with the corresponding Connection handle as a parameter.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Disconnection has occurred.
0x01-0xFF	Disconnection failed to complete. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which was disconnected. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Reason: *Size: 1 Byte*

Value	Parameter Description
0xXX	Reason for disconnection. See Table 6.1 on page 766 for list of Error Codes.

5.2.6 Authentication Complete event

Event	Event Code	Event Parameters
Authentication Complete	0x06	Status, Connection_Handle

Description:

The Authentication Complete event occurs when authentication has been completed for the specified connection. The Connection_Handle must be a Connection_Handle for an ACL connection.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Authentication Request successfully completed.
0x01-0xFF	Authentication Request failed to complete. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which Authentication has been performed. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

5.2.7 Remote Name Request Complete event

Event	Event Code	Event Parameters
Remote Name Request Complete	0x07	Status, BD_ADDR, Remote_Name

Description:

The Remote Name Request Complete event is used to indicate that a remote name request has been completed. The Remote_Name event parameter is a UTF-8 encoded string with up to 248 bytes in length. The Remote_Name event parameter will be null-terminated (0x00) if the UTF-8 encoded string is less than 248 bytes. The BD_ADDR event parameter is used to identify which device the user-friendly name was obtained from.
Note: the Remote_Name Parameter is a string parameter. Endianess does therefore not apply to the Remote_Name Parameter. The first byte of the name is received first.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Remote_Name_Request command succeeded.
0x01-0xFF	Remote_Name_Request command failed. See Table 6.1 on page 766 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the device whose name was requested.

Remote_Name: *Size: 248 Bytes*

Value	Parameter Description
Name[248]	A UTF-8 encoded user-friendly descriptive name for the remote device. If the name contained in the parameter is shorter than 248 bytes, the end of the name is indicated by a NULL byte (0x00), and the following bytes (to fill up 248 bytes, which is the length of the parameter) do not have valid values.

5.2.8 Encryption Change event

Event	Event Code	Event Parameters
Encryption Change	0x08	Status, Connection_Handle, Encryption_Enable

Description:

The Encryption Change event is used to indicate that the change in the encryption has been completed for the Connection Handle specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The Encryption_Enable event parameter specifies the new Encryption Enable for the Connection Handle specified by the Connection_Handle event parameter. This event will occur on both devices to notify both Hosts when Encryption has changed for the specified Connection Handle between two devices.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Encryption Change has occurred.
0x01-0xFF	Encryption Change failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which the link layer encryption has been enabled/disabled for all Connection Handles with the same Bluetooth device endpoint as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Encryption_Enable: *Size: 1 Byte*

Value	Parameter Description
0x00	Link Level Encryption is OFF.
0x01	Link Level Encryption is ON.

5.2.9 Change Connection Link Key Complete event

Event	Event Code	Event Parameters
Change Connection Link Key Complete	0x09	Status, Connection_Handle

Description:

The Change Connection Link Key Complete event is used to indicate that the change in the Link Key for the Connection Handle specified by the Connection_Handle event parameter has been completed.
The Connection_Handle will be a Connection_Handle for an ACL connection.
The Change Connection Link Key Complete event is sent only to the Host which issued the Change_Connection_Link_Key command.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Change_Connection_Link_Key command succeeded.
0x01-0xFF	Change_Connection_Link_Key command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which the Link Key has been change for all Connection Handles with the same Bluetooth device end point as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

5.2.10 Master Link Key Complete event

Event	Event Code	Event Parameters
Master Link Key Complete	0x0A	Status, Connection_Handle, Key_Flag

Description:

The Master Link Key Complete event is used to indicate that the Link Key managed by the master of the piconet has been changed.The Connection_Handle will be a Connection_Handle for an ACL connection. The link key used for the connection will be the temporary link key of the master device or the semi-permanent link key indicated by the Key_Flag. The Key_Flag event parameter is used to indicate which Link Key (temporary link key of the Master, or the semi-permanent link keys) is now being used in the piconet.
Note: for a master, the change from a semi-permanent Link Key to temporary Link Key will affect all Connection Handles related to the piconet. For a slave, this change affects only this particular connection handle. A temporary link key must be used when both broadcast and point-to-point traffic shall be encrypted.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Master_Link_Key command succeeded.
0x01-0xFF	Master_Link_Key command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which the Link Key has been changed for all devices in the same piconet. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Key_Flag: *Size: 1 Byte*

Value	Parameter Description
0x00	Using Semi-permanent Link Key.
0x01	Using Temporary Link Key.

5.2.11 Read Remote Supported Features Complete event

Event	Event Code	Event Parameters
Read Remote Supported Features Complete	0x0B	Status, Connection_Handle, LMP_Features

Description:

The Read Remote Supported Features Complete event is used to indicate the completion of the process of the Link Manager obtaining the supported features of the remote Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The event parameters include a list of LMP features. For details see "Link Manager Protocol" on pa g e183

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Remote_Supported_Features command succeeded.
0x01-0xFF	Read_Remote_Supported_Features command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which is used for the Read_Remote_Supported_Features command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

LMP_Features: *Size: 8 Bytes*

Value	Parameter Description
0xXXXXXXXX XXXXXXXX	Bit Mask List of LMP features. See "Link Manager Protocol" on page 183.

5.2.12 Read Remote Version Information Complete event

Event	Event Code	Event Parameters
Read Remote Version Information Complete	0x0C	Status, Connection_Handle, LMP_Version, Manufacturer_Name, LMP_Subversion

Description:

The Read Remote Version Information Complete event is used to indicate the completion of the process of the Link Manager obtaining the version information of the remote Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The LMP_Version event parameter defines the major hardware version of the Bluetooth hardware. This event parameter only changes when new versions of the Bluetooth hardware are produced for new Bluetooth SIG specifications; it is controlled by the SIG. The Manufacturer_Name event parameter indicates the manufacturer of the remote Bluetooth module. The LMP_Subversion event parameter should be controlled by the manufacturer and should be changed as needed. The LMP_Subversion event parameter defines the various revisions that each version of the Bluetooth hardware will go through as design processes change and errors are fixed. This allows the software to determine what Bluetooth hardware is being used and, if necessary, to work around various bugs in the hardware.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Remote_Version_Information command succeeded.
0x01-0xFF	Read_Remote_Version_Information command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which is used for the Read_Remote_Version_Information command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

LMP_Version: *Size: 1 Byte*

Value	Parameter Description
0xXX	Version of the Current LMP in the remote Bluetooth Hardware, see Table 5.2 on page 239 in the Link Manager Protocol for assigned values (VersNr).

Manufacturer_Name: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Manufacturer Name of the remote Bluetooth Hardware, see Table 5.2 on page 239 in the Link Manager Protocol for assigned values (CompId).

LMP_Subversion: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Subversion of the Current LMP in the remote Bluetooth Hardware, see Table 5.2 on page 239 in the Link Manager Protocol for assigned values (SubVersNr).

5.2.13 QoS Setup Complete event

Event	Event Code	Event Parameters
QoS Setup Complete	0x0D	Status, Connection_Handle, Flags, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation

Description:

The QoS Setup Complete event is used to indicate the completion of the process of the Link Manager setting up QoS with the remote Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. For more detail see "Logical Link Control and Adaptation Protocol Specification" on pag e253

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	QoS_Setup command succeeded.
0x01-0xFF	QoS_Setup command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which is used for the QoS_Setup command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flags: *Size: 1 Byte*

Value	Parameter Description
0x00 – 0xFF	Reserved for Future Use.

Service_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	No Traffic Available.
0x01	Best Effort Available.
0x02	Guaranteed Available.
0x03-0xFF	Reserved for Future Use.

Token_Rate: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Available Token Rate, in bytes per second.

Peak_Bandwidth: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Available Peak Bandwidth, in bytes per second.

Latency: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Available Latency, in microseconds.

Delay_Variation: *Size: 4 Bytes*

Value	Parameter Description
0xXXXXXXXX	Available Delay Variation, in microseconds.

5.2.14 Command Complete event

Event	Event Code	Event Parameters
Command Complete	0x0E	Num_HCI_Command_Packets, Command_Opcode, Return_Parameters

Description:

The Command Complete event is used by the Host Controller for most commands to transmit return status of a command and the other event parameters that are specified for the issued HCI command.

The Num_HCI_Command_Packets event parameter allows the Host Controller to indicate the number of HCI command packets the Host can send to the Host Controller. If the Host Controller requires the Host to stop sending commands, the Num_HCI_Command_Packets event parameter will be set to zero. To indicate to the Host that the Host Controller is ready to receive HCI command packets, the Host Controller generates a Command Complete event with the Command_Opcode 0x0000, and the Num_HCI_Command_Packets event parameter is set to 1 or more. Command_Opcode, 0x0000 is a NOP (No OPeration), and can be used to change the number of outstanding HCI command packets that the Host can send before waiting. See each command for the parameters that are returned by this event.

Event Parameters:

Num_HCI_Command_Packets: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	The Number of HCI command packets which are allowed to be sent to the Host Controller from the Host. Range for N: 0 – 255

Command_Opcode: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Opcode of the command which caused this event.

Return_Parameter(s): *Size: Depends on Command*

Value	Parameter Description
0xXX	This is the return parameter(s) for the command specified in the Command_Opcode event parameter. See each command's definition for the list of return parameters associated with that command.

5.2.15 Command Status event

Event	Event Code	Event Parameters
Command Status	0x0F	Status, Num_HCI_Command_Packets, Command_Opcode

Description:

The Command Status event is used to indicate that the command described by the Command_Opcode parameter has been received, and that the Host Controller is currently performing the task for this command. This event is needed to provide mechanisms for asynchronous operation, which makes it possible to prevent the Host from waiting for a command to finish. If the command can not begin to execute (a parameter error may have occurred, or the command may currently not be allowed), the Status event parameter will contain the corresponding error code, and no complete event will follow since the command was not started. The Num_HCI_Command_Packets event parameter allows the Host Controller to indicate the number of HCI command packets the Host can send to the Host Controller. If the Host Controller requires the Host to stop sending commands, the Num_HCI_Command_Packets event parameter will be set to zero. To indicate to the Host that the Host Controller is ready to receive HCI command packets, the Host Controller generates a Command Status event with Status 0x00 and Command_Opcode 0x0000, and the Num_HCI_Command_Packets event parameter is set to 1 or more. Command_Opcode, 0x0000 is a NOP (No OPeration) and can be used to change the number of outstanding HCI command packets that the Host can send before waiting.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Command currently in pending.
0x01-0xFF	Command failed. See Table 6.1 on page 766 for list of Error Codes.

Num_HCI_Command_Packets: *Size: 1 Byte*

Value	Parameter Description
N = 0xXX	The Number of HCI command packets which are allowed to be sent to the Host Controller from the Host. Range for N: 0 – 255

Command_Opcode: *Size: 2 Bytes*

Value	Parameter Description
0xXXXX	Opcode of the command which caused this event and is pending completion.

5.2.16 Hardware Error event

Event	Event Code	Event Parameters
Hardware Error	0x10	Hardware_Code

Description:

The Hardware Error event is used to indicate some type of hardware failure for the Bluetooth device. This event is used to notify the Host that a hardware failure has occurred in the Bluetooth module.

Event Parameters:

Hardware_Code: *Size: 1 Byte*

Value	Parameter Description
0x00	These Hardware_Codes will be implementation-specific, and will be assigned to indicate various hardware problems.

5.2.17 Flush Occurred event

Event	Event Code	Event Parameters
Flush Occurred	0x11	Connection_Handle

Description:

The Flush Occurred event is used to indicate that, for the specified Connection Handle, the current user data to be transmitted has been removed. The Connection_Handle will be a Connection_Handle for an ACL connection. This could result from the flush command, or be due to the automatic flush. Multiple blocks of an L2CAP packet could have been pending in the Host Controller. If one baseband packet part of an L2CAP packet is flushed, then the rest of the HCI data packets for the L2CAP packet must also be flushed.

Event Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which was flushed. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

5.2.18 Role Change event

Event	Event Code	Event Parameters
Role Change	0x12	Status, BD_ADDR, New_Role

Description:

The Role Change event is used to indicate that the current Bluetooth role related to the particular connection has changed. This event only occurs when both the remote and local Bluetooth devices have completed their role change for the Bluetooth device associated with the BD_ADDR event parameter. This event allows both affected Hosts to be notified when the Role has been changed.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Role change has occurred.
0x01-0xFF	Role change failed. See Table 6.1 on page 766 for list of Error Codes.

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device for which a role change has completed.

New_Role: *Size: 1 Byte*

Value	Parameter Description
0x00	Currently the Master for specified BD_ADDR.
0x01	Currently the Slave for specified BD_ADDR.

5.2.19 Number Of Completed Packets event

Event	Event Code	Event Parameters
Number Of Completed Packets	0x13	Number_of_Handles, Connection_Handle[i], HC_Num_Of_Completed_Packets[i]

Description:

The Number Of Completed Packets event is used by the Host Controller to indicate to the Host how many HCI Data Packets have been completed (transmitted or flushed) for each Connection Handle since the previous Number Of Completed Packets event was sent to the Host. This means that the corresponding buffer space has been freed in the Host Controller. Based on this information, and the HC_Total_Num_ACL_Data_Packets and HC_Total_
Num_SCO_Data_Packets return parameter of the Read_Buffer_Size command, the Host can determine for which Connection Handles the following HCI Data Packets should be sent to the Host Controller.The Number Of Completed Packets event must not be sent before the corresponding Connection Complete event. While the Host Controller has HCI data packets in its buffer, it must keep sending the Number Of Completed Packets event to the Host at least periodically, until it finally reports that all the pending ACL Data Packets have been transmitted or flushed. The rate with which this event is sent is manufacturer specific.
Note that Number Of Completed Packets events will not report on SCO connection handles if SCO Flow Control is disabled. (See Read/Write_SCO_Flow_Control_Enable on page 678 and page 679.)

Event Parameters:

Number_of_Handles: *Size: 1 Byte*

Value	Parameter Description
0xXX	The number of Connection Handles and Num_HCI_Data_Packets parameters pairs contained in this event. Range: 0-255

*Connection_Handle[i]:Size: Number_of_Handles * 2 Bytes(12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

*HC_Num_Of_Completed_Packets [i]: Size: Number_of_Handles * 2 Bytes*

Value	Parameter Description
N = 0xXXXX	The number of HCI Data Packets that have been completed (transmitted or flushed) for the associated Connection Handle since the previous time the event was returned. Range for N: 0x0000-0xFFFF

5.2.20 Mode Change event

Event	Event Code	Event Parameters
Mode Change	0x14	Status, Connection_Handle, Current_Mode, Interval

Description:

The Mode Change event is used to indicate when the device associated with the Connection Handle changes between Active, Hold, Sniff and Park mode. The Connection_Handle will be a Connection_Handle for an ACL connection. The Connection_Handle event parameter is used to indicate which connection the Mode Change event is for. The Current_Mode event parameter is used to indicate which state the connection is currently in. The Interval parameter is used to specify a time amount specific to each state. Each Host Controller that is associated with the Connection Handle which has changed Modes will send the Mode Change event to its Host.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	A Mode Change has occurred.
0x01-0xFF	Hold_Mode, Sniff_Mode, Exit_Sniff_Mode, Park_Mode, or Exit_Park_Mode command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes(12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Current_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Active Mode.
0x01	Hold Mode.
0x02	Sniff Mode.
0x03	Park Mode.
0x04-0xFF	Reserved for future use.

Interval: *Size: 2 Bytes*

Value	Parameter Description
N = 0xXXXX	Hold: Number of Baseband slots to wait in Hold Mode. Hold Interval = N * 0.625 msec (1 Baseband slot) Range for N: 0x0000-0xFFFF Time Range: 0-40.9 sec Sniff: Number of Baseband slots between sniff intervals. Time between sniff intervals = 0.625 msec (1 Baseband slot) Range for N: 0x0000-0xFFFF Time Range: 0-40.9 sec Park: Number of Baseband slots between consecutive beacons. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0000-0xFFFF Time Range: 0-40.9 Seconds

5.2.21 Return Link Keys event

Event	Event Code	Event Parameters
Return Link Keys	0x15	Num_Keys, BD_ADDR [i], Link_Key[i]

Description:

The Return Link Keys event is used by the Host Controller to send the Host one or more stored Link Keys. Zero or more instances of this event will occur after the Read_Stored_Link_Key command. When there are no link keys stored, no Return Link Keys events will be returned. When there are link keys stored, the number of link keys returned in each Return Link Keys event is implementation specific.

Event Parameters:

Num_Keys: *Size: 1 Byte*

Value	Parameter Description
0xXX	Number of Link Keys contained in this event. Range: 0x01 – 0x0B

BD_ADDR [i]: *Size: 6 Bytes * Num_Keys*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the associated Link Key.

Link_Key[i]: *Size:16 Bytes * Num_Keys*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

5.2.22 PIN Code Request event

Event	Event Code	Event Parameters
PIN Code Request	0x16	BD_ADDR

Description:

The PIN Code Request event is used to indicate that a PIN code is required to create a new linkkey. The Host must respond using either the PIN Code Request Reply or the PIN Code Request Negative Reply command, depending on whether the Host can provide the Host Controller with a PIN code or not. Note: If the PIN Code Request event is masked away, then the Host Controller will assume that the Host has no PIN Code.

When the Host Controller generates a PIN Code Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page183 .)

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which a new link key is being created for.

5.2.23 Link Key Request event

Event	Event Code	Event Parameters
Link Key Request	0x17	BD_ADDR

Description:

The Link Key Request event is used to indicate that a Link Key is required for the connection with the device specified in BD_ADDR. If the Host has the requested stored Link Key, then the Host will pass the requested Key to the Host Controller using the Link_Key_Request_Reply Command. If the Host does not have the requested stored Link Key, then the Host will use the Link_Key_Request_Negative_Reply Command to indicate to the Host Controller that the Host does not have the requested key.
Note: If the Link Key Request event is masked away, then the Host Controller will assume that the Host has no additional link keys.

When the Host Controller generates a Link Key Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See "Link Manager Protocol" on page183 .)

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which a stored link key is being requested.

5.2.24 Link Key Notification event

Event	Event Code	Event Parameters
Link Key Notification	0x18	BD_ADDR, Link_Key,Key_Type

Description:

The Link Key Notification event is used to indicate to the Host that a new Link Key has been created for the connection with the device specified in BD_ADDR. The Host can save this new Link Key in its own storage for future use. Also, the Host can decided to store the Link Key in the Host Controller's Link Key Storage by using the Write_Stored_Link_Key command. The Key_Type event parameter informs the Host about which key type (combination key, local unit key or remote unit key) that has been used during pairing. If pairing with unit key is not supported, the Host can for instance discard the key or disconnect the link.

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device for which the new link key has been generated.

Link_Key: *Size: 16 Bytes*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

Key_Type: *Size: 1 Bytes*

Value	Parameter Description
0x00	Combination Key
0x01	Local Unit Key
0x02	Remote Unit Key
0x03-0xFF	Reserved

5.2.25 Loopback Command event

Event	Event Code	Event Parameters
Loopback Command	0x19	HCI_Command_Packet

Description:

When in Local Loopback mode, the Host Controller loops back commands and data to the Host. The Loopback Command event is used to loop back all commands that the Host sends to the Host Controller with some exceptions. See Section 4.10.1, “Read_Loopback_Mode,” on page717 for a description of which commands that are not looped back. The HCI_Command_Packet event parameter contains the entire HCI Command Packet including the header. Note: the event packet is limited to a maximum of 255 bytes in the payload; since an HCI Command Packet has 3 bytes of header data, only the first 252 bytes of the command parameters will be returned.

Event Parameters:

HCI_Command_Packet: *Size: Depends on Command*

Value	Parameter Description
0xXXXXXX	HCI Command Packet, including header.

5.2.26 Data Buffer Overflow event

Event	Event Code	Event Parameters
Data Buffer Overflow	0x1A	Link_Type

Description:

This event is used to indicate that the Host Controller's data buffers have been overflowed. This can occur if the Host has sent more packets than allowed. The Link_Type parameter is used to indicate that the overflow was caused by ACL or SCO data.

Event Parameters:

Link_Type: *Size: 1 Byte*

Value	Parameter Description
0x00	SCO Buffer Overflow (Voice Channels).
0x01	ACL Buffer Overflow (Data Channels).
0x02-0xFF	Reserved for Future Use.

5.2.27 Max Slots Change event

Event	Event Code	Event Parameters
Max Slots Change	0x1B	Connection_Handle, LMP_Max_Slots

Description:

This event is used to notify the Host about the LMP_Max_Slots parameter when the value of this parameter changes. It will be sent each time the maximum allowed length, in number of slots, for baseband packets transmitted by the local device, changes. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

LMP_Max_Slots: *Size: 1 byte*

Value	Parameter Description
0x01, 0x03, 0x05	Maximum number of slots allowed to use for baseband packets, see Section 3.22 on page 229 and Section 5.1 on page 239 in "Link Manager Protocol" on pag e183.

5.2.28 Read Clock Offset Complete event

Event	Event Code	Event Parameters
Read Clock Offset Complete	0x1C	Status, Connection_Handle, Clock_Offset

Description:

The Read Clock Offset Complete event is used to indicate the completion of the process of the Link Manager obtaining the Clock Offset information of the Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Read_Clock_Offset command succeeded.
0x01-0xFF	Read_Clock_Offset command failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Clock Offset parameter is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Clock_Offset: *Size: 2 Bytes*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Reserved.

5.2.29 Connection Packet Type Changed event

Event	Event Code	Event Parameters
Connection Packet Type Changed	0x1D	Status, Connection_Handle, Packet_Type

Description:

The Connection Packet Type Changed event is used to indicate that the process has completed of the Link Manager changing which packet types can be used for the connection. This allows current connections to be dynamically modified to support different types of user data. The Packet_Type event parameter specifies which packet types the Link Manager can use for the connection identified by the Connection_Handle event parameter for sending L2CAP data or voice. The Packet_Type event parameter does not decide which packet types the LM is allowed to use for sending LMP PDUs.

Event Parameters:

Status: *Size: 1 Byte*

Value	Parameter Description
0x00	Connection Packet Type changed successfully.
0x01-0xFF	Connection Packet Type Changed failed. See Table 6.1 on page 766 for list of Error Codes.

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Packet_Type: *Size: 2 Bytes*

For ACL_Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	DM1
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.

Value	Parameter Description
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

For SCO_Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

5.2.30 QoS Violation event

Event	Event Code	Event Parameters
QoS Violation	0x1E	Connection_Handle

Description:

The QoS Violation event is used to indicate the Link Manager is unable to provide the current QoS requirement for the Connection Handle. This event indicates that the Link Manager is unable to provide one or more of the agreed QoS parameters. The Host chooses what action should be done. The Host can reissue QoS_Setup command to renegotiate the QoS setting for Connection Handle. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Connection_Handle: *Size: 2 Bytes (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle that the LM is unable to provide the current QoS requested for. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

5.2.31 Page Scan Mode Change event

Event	Event Code	Event Parameters
Page Scan Mode Change	0x1F	BD_ADDR, Page_Scan_Mode

Description:

The Page Scan Mode Change event indicates that the remote Bluetooth device with the specified BD_ADDR has successfully changed the Page_Scan_Mode.

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the remote device.

Page_Scan_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	Mandatory Page Scan Mode.
0x01	Optional Page Scan Mode I.
0x02	Optional Page Scan Mode II.
0x03	Optional Page Scan Mode III.
0x04 – 0xFF	Reserved.

5.2.32 Page Scan Repetition Mode Change event

Event	Event Code	Event Parameters
Page Scan Repetition Mode Change	0x20	BD_ADDR, Page_Scan_Repetition_Mode

Description:

The Page Scan Repetition Mode Change event indicates that the remote Bluetooth device with the specified BD_ADDR has successfully changed the Page_Scan_Repetition_Mode (SR).

Event Parameters:

BD_ADDR: *Size: 6 Bytes*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the remote device.

Page_Scan_Repetition_Mode: *Size: 1 Byte*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

6 LIST OF ERROR CODES

6.1 LIST OF ERROR CODES

This section of the document lists the various possible error codes. When a command fails, Error codes are returned to indicate the reason for the error. Error codes have a size of one byte, and the possible range of failure codes is 0x01-0xFF. Section 6.2 provides an error code usage description for each error code.

Error Code	Description
0x01	Unknown HCI Command.
0x02	No Connection.
0x03	Hardware Failure.
0x04	Page Timeout.
0x05	Authentication Failure.
0x06	Key Missing.
0x07	Memory Full.
0x08	Connection Timeout.
0x09	Max Number Of Connections.
0x0A	Max Number Of SCO Connections To A Device.
0x0B	ACL connection already exists.
0x0C	Command Disallowed.
0x0D	Host Rejected due to limited resources.
0x0E	Host Rejected due to security reasons.
0x0F	Host Rejected due to remote device is only a personal device.
0x10	Host Timeout.
0x11	Unsupported Feature or Parameter Value.
0x12	Invalid HCI Command Parameters.
0x13	Other End Terminated Connection: User Ended Connection.
0x14	Other End Terminated Connection: Low Resources.
0x15	Other End Terminated Connection: About to Power Off.
0x16	Connection Terminated by Local Host.
0x17	Repeated Attempts.

Error Code	Description
0x18	Pairing Not Allowed.
0x19	Unknown LMP PDU.
0x1A	Unsupported Remote Feature.
0x1B	SCO Offset Rejected.
0x1C	SCO Interval Rejected.
0x1D	SCO Air Mode Rejected.
0x1E	Invalid LMP Parameters.
0x1F	Unspecified Error.
0x20	Unsupported LMP Parameter Value.
0x21	Role Change Not Allowed
0x22	LMP Response Timeout
0x23	LMP Error Transaction Collision
0x24	LMP PDU Not Allowed
0x25	Encryption Mode Not Acceptable
0x26	Unit Key Used
0x27	QoS is Not Supported
0x28	Instant Passed
0x29	Pairing with Unit Key Not Supported
0x2A-0xFF	Reserved for Future Use.

Table 6.1: List of Possible Error Codes

6.2 HCI ERROR CODE USAGE DESCRIPTIONS

The purpose of this section is to give descriptions of how the error codes specified in Table 6.1 on page 766 should be used. It is beyond the scope of this document to give detailed descriptions of all situations where error codes can be used – especially as this may also, in certain cases, be implementation-dependent. However, some error codes that are to be used only in very special cases are described in more detail than other, more general, error codes.

The following error codes are only used in LMP messages, and are therefore not described in this section:

- Unknown LMP PDU (0x19)
- SCO Offset Rejected (0x1B)
- SCO Interval Rejected (0x1C)
- SCO Air Mode Rejected (0x1D)
- Invalid LMP Parameters (0x1E)

Some of the following error code descriptions describe as implementation-dependent whether the error should be returned using a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00). In these cases, the command can not start executing because of the error, and it is therefore recommended to use the Command Status event. The reason for this suggested course of action is that it is not possible to use the Command Status event in all software architectures.

6.3 UNKNOWN HCI COMMAND (0X01)

The ‘Unknown HCI Command’ error code is returned by the Host Controller in the Status parameter in a Command Complete event or a Command Status event when the Host Controller receives an HCI Command Packet with an Opcode that it does not recognize. The OpCode given might not correspond to any of the OpCodes specified in this document, or any vendor-specific OpCodes, or the command may not have been implemented. If a Command Complete event is returned, the Status parameter is the only parameter contained in the Return_Parameters event parameter. Which of the two events is used is implementation-dependent.

6.4 NO CONNECTION (0X02)

The ‘No Connection’ error code is returned by the Host Controller in the Status parameter in an event when the Host has issued a command which requires an existing connection and there is currently no connection corresponding to the specified Connection Handle or BD Address. If the issued command is a command for which a Command Complete event should be returned, the event containing the error code is a Command Complete event. Otherwise, the event containing the error code is a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00), depending on the implementation.

6.5 HARDWARE FAILURE (0X03)

The ’Hardware Failure’ error code is returned by the Host Controller in the Status parameter in an event when the Host has issued a command and this command can not be executed because of a hardware failure. If the issued command is a command for which a Command Complete event should be returned, the event containing the error code is a Command Complete event. Otherwise, the event containing the error code is a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00) depending on the implementation.

6.6 PAGE TIMEOUT (0X04)

The ’Page Timeout’ error code is returned by the Host Controller in the Status parameter of the Connection Complete event when the Host has issued a Create_Connection command and the specified device to connect to does not respond to a page at baseband level before the page timer expires (a page timeout occurs). The error code can also be returned in the Status parameter of a Remote Name Request Complete event when the Host has issued a Remote_Name_Request command and a temporary connection needs to be established but a page timeout occurs. (The page timeout is set using the Write_Page_Timeout command.)

6.7 AUTHENTICATION FAILURE (0X05)

The ’Authentication Failure’ error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Authentication Complete event when pairing or authentication fails due to incorrect results in the pairing/ authentication calculations (because of incorrect PIN code or link key).

"The 'Authentication Failure' error code can also be used as a value for the Reason parameter in the Disconnect command (as a reason code). The error code will then be sent over the air so that it is returned in the Reason parameter of a Disconnection Complete event on the remote side. In the Disconnection Complete event following a Command Status event (where Status=0x00) on the local side on which the Disconnect command has been issued, the Reason parameter will however contain the reason code 'Connection Terminated By Local Host'.

6.8 KEY MISSING (0X06)

The 'Key Missing' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Authentication Complete event when pairing fails because of missing PIN code(s).

6.9 MEMORY FULL (0X07)

The 'Memory Full' error code is returned by the Host Controller in the Status parameter in a Command Complete event when the Host has issued a command that requires the Host Controller to store new parameters and the Host Controller does not have memory capacity for this. This may be the case after the Set_Event_Filter command has been issued. Note that for the Write_Stored_Link_Key command, no error is returned when the Host Controller can not store any more link keys. The Host Controller stores as many link keys as there is free memory to store in, and the Host is notified of how many link keys were successfully stored.

6.10 CONNECTION TIMEOUT (0X08)

Note: this error code is used to indicate a reason for disconnection. It is normally returned in the Reason parameter of a Disconnection Complete event. It is therefore called reason code in the following description.

The 'Connection Timeout' reason code is sent by the Host Controller in an event when the link supervision timer (see "Baseband Timers" on page1019) expires and the link therefore is considered to be lost. The link supervision timeout is set using Write_Link_Supervision_Timeout. The event that returns this reason code will most often be a Disconnection Complete event (in the Reason parameter). The event will be returned on both sides of the connection, where one Disconnection Complete event will be sent from the Host Controller to the Host for each Connection Handle that exists for the physical link to the other device.

(It is possible for a link loss to be detected during connection set up, in which case the reason code would be returned in a Connection Complete event.)

6.11 MAX NUMBER OF CONNECTIONS (0X09)

The 'Max Number Of Connections' error code is returned by the Host Controller in the Status parameter of a Command Status event, a Connection Complete event or a Remote Name Request Complete event when the Bluetooth module can not establish any more connections. It is implementation specific whether the error is returned in a Command Status event or the event following the Command Status event (where Status=0x00 in the Command Status event). The reason for this error may be hardware or firmware limitations. Before the error is returned, the Host has issued a Create_Connection, Add_SCO_Connection or Remote_Name_Request command. The error can be returned in a Remote Name Request Complete event when a temporary connection needs to be established to request the name.

6.12 MAX NUMBER OF SCO CONNECTIONS TO A DEVICE (0X0A)

The 'Max Number Of SCO Connections To A Device' error code is returned by the Host Controller in the Status parameter of a Command Status event or a Connection Complete event (following a Command Status event with Status=0x00) when the maximum number of SCO connections to a device has been reached. Which of the two events that is used depends on the implementation. The device is a device that has been specified in a previously issued Add_SCO_Connection command.

6.13 ACL CONNECTION ALREADY EXISTS (0X0B)

The 'ACL connection already exists' error code is returned by the Host Controller in the Status parameter of a Command Status event or a Connection Complete event (following a Command Status event with Status=0x00) when there already is one ACL connection to a device and the Host tries to establish another one using Create_Connection. Which of the two events that is used depends on the implementation.

6.14 COMMAND DISALLOWED (0X0C)

The 'Command Disallowed' error code is returned by the Host Controller in the Status parameter in a Command Complete event or a Command Status event when the Host Controller is in a state where it is only prepared to accept commands with certain OpCodes and the HCI Command Packet received does not contain any of these OpCodes. The Command Completeevent should be used if the issued command is a command for which a Command Complete event should be returned. Otherwise, the Command Status event should be used. The Host Controller is not required to use the 'Unknown HCI Command' error code, since this may require unnecessary processing of the received (and currently not allowed) OpCode. When to use the 'Command Disallowed' error code is mainly implementation-dependent. Certain implementations may, for example, only accept the appropriate HCI response commands after the Connection Request, Link Key Request or PIN Code Request events.
Note: the Reset command should always be allowed.

6.15 HOST REJECTED DUE TO … (0X0D-0X0F)

Note: these error codes are used to indicate a reason for rejecting an incoming connection. They are therefore called reason codes in the following description.

When a Connection Request event has been received by the Host and the Host rejects the incoming connection by issuing the Reject_Connection _ Request command, one of these reason codes is used as value for the Reason parameter. The issued reason code will be returned in the Status parameter of the Connection Complete event that will follow the Command Status event (with Status=0x00) returned by the Host Controller after the Reject_

Connection_Request command has been issued. The reason code issued in the Reason parameter of the Reject_Connection_Request command will also be sent over the air, so that it is returned in a Connection Complete event on the initiating side. Before this, the initiating side has issued a Create_ Connection command or Add_SCO_Connection command, and has received a Command Status event (with Status=0x00).

6.16 HOST TIMEOUT (0X10)

Note: this error code is used to indicate a reason for rejecting an incoming connection. It is therefore called reason code in the following description.

Assume that a Connection Request event has been received by the Host and that the Host does not issue the Accept_Connection_Request or Reject_ Connection_Request command before the connection accept timer expires (the connection accept timeout is set using Write_Connection_Accept_ Timeout). In this case, the 'Host Timeout' reason code will be sent by the Host Controller in the Status parameter of a Connection Complete event. The reason code will also be sent over the air, so that it is returned in a Connection Complete event on the initiating side. The initiating side has before this issued a Create_Connection or Add_SCO_Connection command and has received a Command Status event (with Status=0x00).

6.17 UNSUPPORTED FEATURE OR PARAMETER VALUE (0X11)

The 'Unsupported Feature or Parameter Value' error code is returned by the Host Controller in the Status parameter in an event when the Host Controller has received a command where one or more parameters have values that are not supported by the hardware (the parameters are, however, within the allowed parameter range specified in this document). If the issued command is a command for which a Command Complete event should be returned, the event containing the error code is a Command Complete event. Otherwise, the event containing the error code is a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00) depending on the implementation.

6.18 INVALID HCI COMMAND PARAMETERS (0X12)

The 'Invalid HCI Command Parameters' error code is returned by the Host Controller in the Status parameter of an event when the total parameter length (or the value of one or more parameters in a received command) does not conform to what is specified in this document.

The error code can also be returned if a parameter value is currently not allowed although it is inside the allowed range for the parameter. One case is when a command requires a Connection Handle for an ACL connection but the Host has given a Connection Handle for an SCO connection as a parameter

instead. Another case is when a link key, a PIN code or a reply to an incoming connection has been requested by the Host Controller by using an event but the Host replies using a response command with a BD_ADDR for which no request has been made.

If the issued command is a command for which a Command Complete event should be returned, the event containing the error code is a Command Complete event. Otherwise, the event containing the error code is a Command Status event or the event associated with the issued command (following a Command Status event with Status=0x00), depending on the implementation.

6.19 OTHER END TERMINATED CONNECTION: … (0X13-0X15)

Note: these error codes are used to indicate a reason for disconnecting a connection. They are therefore called reason codes in the following description.

When the Host issues the Disconnect command, one of these reason codes is used as value for the reason parameter. The 'Connection Terminated By Local Host' reason code will then be returned in the Reason parameter of the Disconnection Complete event that will follow the Command Status event (with Status=0x00) that is returned by the Host Controller after the Disconnect command has been issued. The reason code issued in the Reason parameter of the Disconnect command will also be sent over the air, so that it is returned in the Reason parameter of a Disconnection Complete event on the remote side.

6.20 CONNECTION TERMINATED BY LOCAL HOST (0X16)

See description in 6.19. This error code is called a reason code, since it is returned in the Reason parameter of a Disconnection Complete event.

6.21 REPEATED ATTEMPTS (0X17)

The 'Repeated Attempts' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Authentication Complete event when a device does not allow authentication or pairing because too little time has elapsed since an unsuccessful authentication or pairing attempt. See "Link Manager Protocol" on page 183 for a description of how repeated attempts work.

6.22 PAIRING NOT ALLOWED (0X18)

The 'Pairing Not Allowed' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Authentication Complete event when a device for some reason does not allow pairing. An example may be a PSTN adapter that only allows pairing during a certain time window after a button has been pressed on the adapter.

6.23 UNSUPPORTED REMOTE FEATURE (0X1A)

The 'Unsupported Remote Feature' error code is returned by the Host Controller in the Status parameter of the event associated with the issued command when a remote device that has been specified in the command parameters does not support the feature associated with the issued command. The 'Unsupported Remote Feature' error code can also be used as a value for the Reason parameter in the Disconnect command (as a reason code). The error code will then be sent over the air so that it is returned in the Reason parameter of a Disconnection Complete event on the remote side. In the Disconnection Complete event following a Command Status event (where Status=0x00) on the local side on which the Disconnect command has been issued, the Reason parameter will however contain the reason code 'Connection Terminated By Local Host'. (The 'Unsupported Remote Feature' error code is called 'Unsupported LMP Feature' in the LMP specification, see "Link Manager Protocol" on pa ge183.)

6.24 UNSPECIFIED ERROR (0X1F)

The 'Unspecified error' error code is used when no other error code specified in this document is appropriate to use.

6.25 UNSUPPORTED LMP PARAMETER VALUE (0X20)

The 'Unsupported LMP Parameter Value' error code is returned by the Host Controller in the Status parameter of the event associated with the issued command when a remote device that has been specified in the command parameters sent back an LMP message containing the LMP error code 0x20, 'Unsupported parameter values' (see "Link Manager Protocol" on pag e183).

6.26 ROLE CHANGE NOT ALLOWED (0X21)

The 'Role Change Not Allowed' error code is returned by the Host Controller in the Status parameter in a Connection Complete event or Role Change event when role change is not allowed. If the local Host issues the Switch_Role command and the remote device rejects the role change, the error code will be returned in a Role Change event. If a connection fails because a device accepts an incoming ACL connection with a request for role change and the role change is rejected by the initiating device, the error code will be returned in a Connection Complete event on both sides.

6.27 LMP RESPONSE TIMEOUT (0X22)

The 'LMP Response Timeout' error code is returned by the Host Controller in the Status parameter in a Command Complete event or an event associated with the issued command following a Command Status event with Status=0x00, when the remote device does not respond to the LMP PDUs from the local device as a result of the issued command within LMP response timeout. (See "Link Manager Protocol" on page183)

6.28 LMP ERROR TRANSACTION COLLISION (0X23)

The ’LMP Error Transaction Collision’ error code is returned by the Host Controller in the Status parameter of the event associated with the issued command when a remote device that has been specified in the command parameters sends back an LMP message containing the LMP error code 0x23, "LMP Error Transaction Collision" (see “Link Manager Protocol” on page183).

6.29 LMP PDU NOT ALLOWED (0X24)

The ’LMP PDU Not Allowed’ error code is returned by the Host Controller in the Status parameter of the event associated with the issued command when a remote device that has been specified in the command parameters sends back an LMP message containing the LMP error code 0x24, "PDU Not Allowed" (see “Link Manager Protocol” on page183).

6.30 ENCRYPTION MODE NOT ACCEPTABLE (0X25)

The 'Encryption Mode Not Acceptable' error code is returned by the Host Controller in the Status parameter of a Connection Complete event or an Encryption Change event when no agreement can be reached on which encryption mode to use.

6.31 UNIT KEY USED (0X26)

The 'Unit Key Used' error code is returned by the Host Controller in the Status parameter of a Command Status event or a Change Connection Link Key Complete event (following a Command Status event with Status=0x00) if the link key can not be changed because it is a unit key.

6.32 QOS IS NOT SUPPORTED (0X27)

The 'QoS is not supported' error code is returned by the Host Controller in the Status parameter of a Command Status event or a QoS Setup Complete event (following a Command Status event with Status=0x00) if the requested quality of service is not supported.

6.33 INSTANT PASSED (0X28)

The 'Instant passed' error code is returned by the Host Controller in the Status parameter of a Role Change event if a role switch can not be performed because the instant at which the role switch shall start is in the past.

6.34 PAIRING WITH UNIT KEY NOT SUPPORTED (0X29)

Note: this error code can be used to indicate a reason for disconnecting a connection. It is therefore called reason code in the following description.

When the Host issues the Disconnect command, this reason code can be used as a value for the Reason parameter (if the Key_Type parameter in a Link Key Notification event indicates that unit key is used and this is not supported). The reason code will then be returned in the Reason parameter of the Disconnection Complete event that will follow the Command Status event (with Status=0x00) that is returned by the Host Controller after the Disconnect command has been issued. The reason code issued in the Reason parameter of the Disconnect command will also be sent over the air, so that it is returned in the Reason parameter of a Disconnection Complete event on the remote side.

7 LIST OF ACRONYMS AND ABBREVIATIONS

Acronym or abbreviation	Complete name
ACL	Asynchronous Connection Less
BD_ADDR	Bluetooth Device Address
DH	Data High rate
DIAC	Dedicated Inquiry Access Code
DM	Data Medium rate
DUT	Device Under Test
DV	Data Voice
GIAC	General Inquiry Access Code
HCI	Host Controller Interface
L2CAP	Logical Link Control and Adaptation Protocol
L_CH	Logical Channel
LAP	Lower Address Part
LC	Link Controller
LM	Link Manager
LMP	Link Manager Protocol
OCF	Opcode Command Field
OGF	OpCode Group Field
RF	Radio Frequency
RSSI	Received Signal Strength Indication
SCO	Synchronous Connection Oriented
TBD	To Be Defined
UA	User Asynchronous
UI	User Isochronous
US	User Synchronous
USB	Universal Serial Bus

Table 7.1: List of Acronyms and Abbreviations

8 LIST OF FIGURES

Figure 1.1: Overview of the Lower Software Layers 546
Figure 1.2: End to End Overview of Lower Software Layers to Transfer Data 547
Figure 1.3: Bluetooth Hardware Architecture Overview. 548
Figure 1.4: Bluetooth Block Diagram with USB HCI 549
Figure 1.5: Bluetooth Block Diagram with PC-Card HCI 549
Figure 4.1: HCI Command Packet 556
Figure 4.2: HCI Event Packet 557
Figure 4.3: HCI ACL Data Packet 558
Figure 4.4: HCI SCO Data Packet 561
Figure 4.5: Local Loopback Mode 721
Figure 4.6: Remote Loopback Mode 721
Figure 4.7: Local Loopback Mode 724
Figure 4.8: Remote Loopback Mode 724

9 LIST OF TABLES

Table 5.1: List of Supported Events 727
Table 6.1: List of Possible Error Codes 770
Table 7.1: List of Acronyms and Abbreviations 780

Part H:2

HCI USB TRANSPORT LAYER

An addendum to the HCI document

This document describes the USB transport layer (between a host and the host controller). HCI commands flow through this layer, but the layer does not decode the commands.

CONTENTS

1 Overview 786

2 USB Endpoint Expectations 788

2.1 Descriptor Overview 788

2.2 Control Endpoint Expectations 793

2.3 Bulk Endpoints Expectations 793

2.4 Interrupt Endpoint Expectations 793

2.5 Isochronous Endpoints Expectations 794

3 Class Code 795

4 Device Firmware Upgrade 796

5 Limitations 797

5.1 Power Specific Limitations 797

5.2 Other Limitations 797

1 OVERVIEW

This document discusses the requirements of the Universal Serial Bus (USB) interface for Bluetooth hardware. Readers should be familiar with USB, USB design issues, Advanced Configuration Power Interface (ACPI), the overall Bluetooth architecture, and the basics of the radio interface.

The reader should also be familiar with the Bluetooth Host Controller Interface.

Referring to Figure 1.1 below, notice that this document discusses the implementation details of the two-way arrow labelled 'USB Function'.

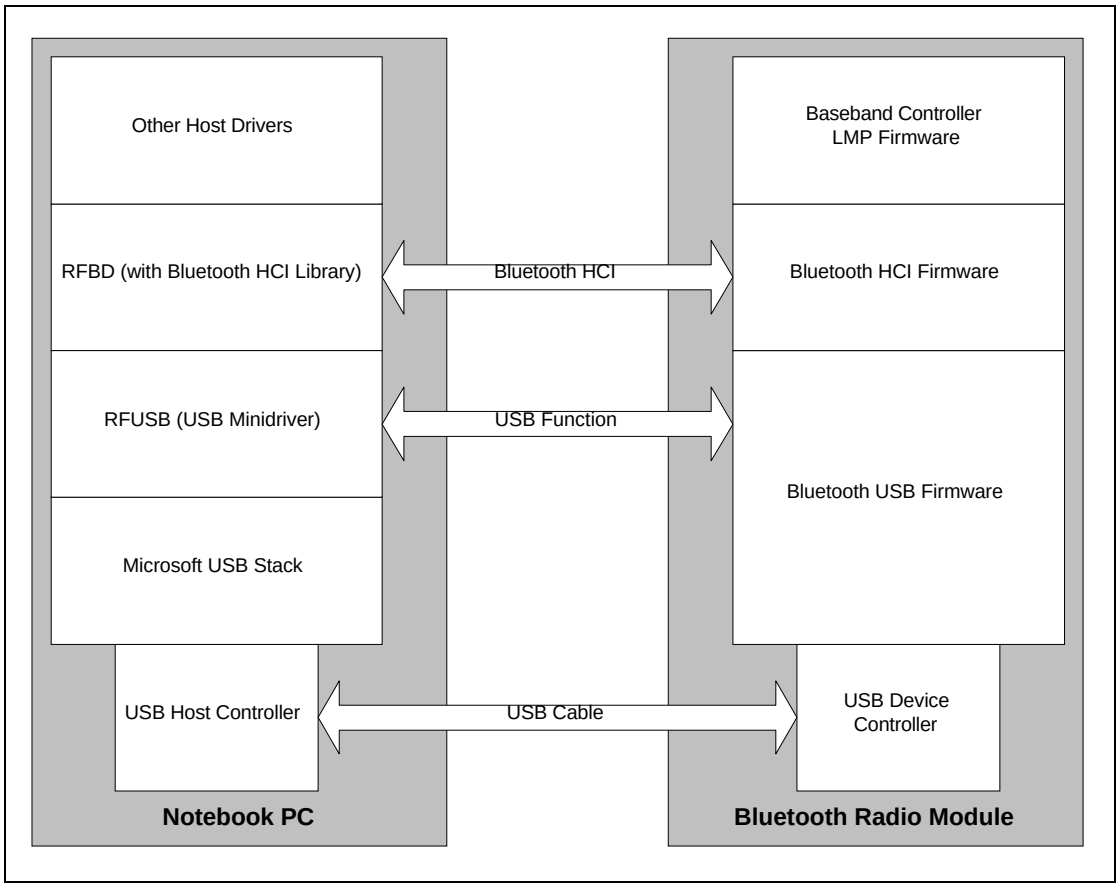


Figure 1.1: The Figure illustrates the relationship between the host and the Bluetooth Radio Module

The USB hardware can be embodied in one of two ways:

1. As a USB dongle, and
2. Integrated onto the motherboard of a notebook PC.

Finally, for an overview of the connection that is established between two Bluetooth devices, reference Figure 1.2, below.

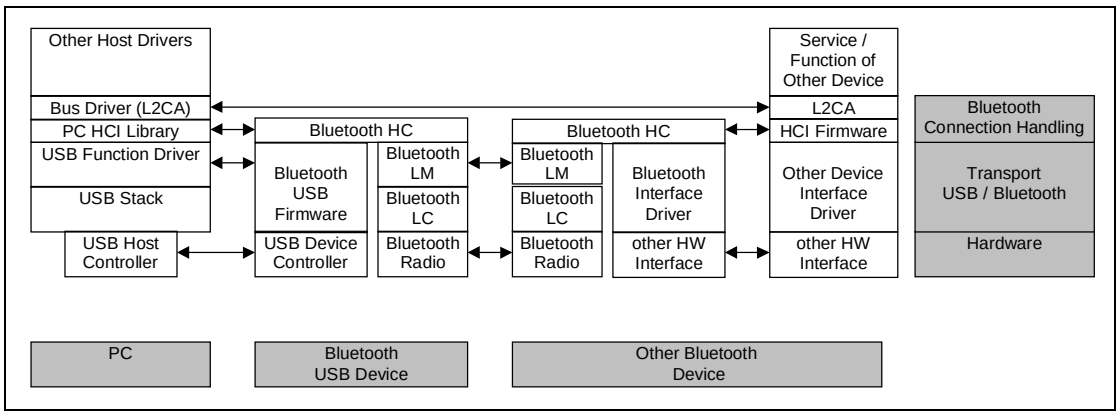


Figure 1.2: The figure illustrates the flow of data from one Bluetooth device to another

2 USB ENDPOINT EXPECTATIONS

This section outlines specific USB endpoints that are required in order to function properly with the host. This section assumes a basic familiarity with USB. The endpoint numbers (labelled 'Suggested Endpoint Address' below) may be dynamically recognized upon driver initialization – this depends on the implementation.

2.1 DESCRIPTOR OVERVIEW

The USB device is expected to be a high-speed device.

The firmware configuration consists of two interfaces. The first interface (interface zero) has no alternate settings and contains the bulk and interrupt endpoints. The second interface (interface one) provides scalable isochronous bandwidth consumption. The second interface has four alternate settings that provide different consumption based on the required isochronous bandwidth. The default interface is empty so that the device is capable of scaling down to no isochronous bandwidth.

An HCI frame - consisting of an HCI header and HCI data - should be contained in one USB transaction. A USB transaction is defined as one or more USB frames that contain the data from one IO request. For example, an ACL data packet containing 256 bytes (both HCI header and HCI data) would be sent over the bulk endpoint in one IO request. That IO request will require four 64-byte USB frames - and forms a transaction.

The endpoints are spread across two interfaces so, when adjusting isochronous bandwidth consumption (via select interface calls), any pending bulk and/or interrupt transactions do not have to be terminated and resubmitted.

The following table outlines the required configuration

Interface Number	Alternate Setting	Suggested Endpoint Address	Endpoint Type	Suggested Max Packet Size
HCI Commands				
0	0	0x00	Control	8/16/32/64
HCI Events				
0	0	0x81	Interrupt (IN)	16
ACL Data				
0	0	0x82	Bulk (IN)	32/64
0	0	0x02	Bulk (OUT)	32/64
No active voice channels (for USB compliance)				
1	0	0x83	Isoch (IN)	0
1	0	0x03	Isoch (OUT)	0
One voice channel with 8-bit encoding				
1	1	0x83	Isoch (IN)	9
1	1	0x03	Isoch (OUT)	9
Two voice channels with 8-bit encoding & One voice channel with 16-bit encoding				
1	2	0x83	Isoch (IN)	17
1	2	0x03	Isoch (OUT)	17
Three voice channels with 8-bit encoding				
1	3	0x83	Isoch (IN)	25
1	3	0x03	Isoch (OUT)	25
Two voice channels with 16-bit encoding				
1	4	0x83	Isoch (IN)	33
1	4	0x03	Isoch (OUT)	33
Three voice channels with 16-bit encoding				
1	5	0x83	Isoch (IN)	49
1	5	0x03	Isoch (OUT)	49

The following two examples are used to demonstrate the flow of data given the describe endpoints.

Number of voice channels	Duration of voice data	Encoding
One	3 ms per IO Request	8-bit

Time (ms	USB data (header refers to HCI header) (Receive & Send from the host)	Queued data (read / write)	Time (ms)	Air data	Amount Received / Sent (ms)
0	Receive 0 bytes Send 9 bytes (3 header, 6 data)	0 / 6	0	Send 0	0 / 0
		10 / 6	0.625	Receive 10	1.25 / 0
1	Receive 0 bytes Send 9 bytes (9 bytes HCI data)	10 / 15	1.25	Send 0	1.25 / 0
		20 / 15	1.875	Receive 10	2.50 / 0
2	Receive 0 bytes Send 9 bytes (9 bytes HCI data)	20 / 24	2.50	Send 0	2.50 / 0
		30 / 24	3.125	Receive 10	3.75 / 0
3	Receive 9 bytes (3 header, 6 data) Send 9 bytes (3 header, 6 data)	24 / 20	3.75	Send 10	3.75 / 1.25
4	Receive 9 bytes (9 bytes data) Send 9 bytes (9 bytes HCI data)	25 / 29	4.375	Receive 10	5.0 / 1.25
5	Receive 9 bytes (9 bytes data) Send 9 bytes (9 bytes HCI data)	16 / 28	5.0	Send 10	5.0 / 2.50
		26 / 28	5.625	Receive 10	6.25 / 2.50
6	Receive 9 bytes (3 header, 6 data) Send 9 bytes (3 header, 6 data)	20 / 24	6.25	Send 10	6.25 / 3.75
		30 / 24	6.875	Receive 10	7.5 / 3.75
7	Receive 9 bytes (9 bytes data) Send 9 bytes (9 bytes HCI data)	21 / 23	7.5	Send 10	7.5 / 5.0
8	Receive 9 bytes (9 bytes data) Send 9 bytes (9 bytes HCI data)	22 / 32	8.125	Receive 10	8.75 / 5.0
		22 / 22	8.75	Send 10	8.75 / 6.25
9	Receive 9 bytes (3 header, 6 data) Send 9 bytes (3 header, 6 data)	26 / 28	9.375	Receive 10	10.0 / 6.25

Table 2.1:

Time (ms	USB data (header refers to HCI header) (Receive & Send from the host)	Queued data (read / write)	Time (ms)	Air data	Amount Received / Sent (ms)
10	Receive 9 bytes (9 bytes data) Send 9 bytes (9 bytes HCI data)	17 / 27	10	Send 10	10.0 / 7.5
		27 / 27	10.625	Receive 10	11.25 / 7.5
11	Receive 9 bytes (9 bytes data) Send 9 bytes (9 bytes HCI data)	18 / 26	11.25	Send 10	11.25 / 8.75

Table 2.1:

Convergence is expected because the radio is sending out an average of 8 bytes of voice data every 1 ms and USB is sending 8 bytes of voice data every 1 ms.

Number of voice channels	Duration of voice data	Encoding
Two	3 ms per IO Request	8-bit

Time (ms	USB data (header refers to HCI header) (Receive & Send from the host)	Queued data (read / write)	Time (ms)	Air data	Amount Received / Sent (ms)
0	Receive 0 bytes for Channel #1 Send 17 bytes (3 header, 14 data) for Channel #1	C1- 0/14 C2- 0/0	0	Send 0 for C1	C1- 0/0 C2- 0/0
		C1- 20/14 C2- 0/0	0.625	Receive 20 for C1	C1- 2.5/0 C2- 0/0
1	Receive 0 bytes for Channel #1 Send 17 bytes (17 bytes HCI data) for Channel #1	C1- 20/31 C2- 0/0	1.25	Send 0 for C2	C1- 2.5/0 C2- 0/0
		C1- 20/31 C2- 20/0	1.875	Receive 20 for C2	C1- 2.5/0 C2- 2.5/0
2	Receive 0 bytes for Channel #1 Send 17 bytes (17 bytes HCI data) for Channel #1	C1- 20/28 C2- 20/0	2.50	Send 20 for C1	C1- 2.5/2.5 C2- 2.5/0
		C1- 40/28 C2- 0/0	3.125	Receive 20 for C1	C1- 5.0/2.5 C2- 2.5/0

Table 2.2:

Time (ms	USB data (header refers to HCI header) (Receive & Send from the host)	Queued data (read / write)	Time (ms)	Air data	Amount Received / Sent (ms)
3	Receive 0 bytes for Channel #2 Send 17 bytes (3 header, 14 data) for Channel #2	C1- 40/28 C2- 20/14	3.75	Send 0 for C2	C1- 5.0/2.5 C2- 2.5/0
4	Receive 0 bytes for Channel #2 Send 17 bytes (17 bytes HCI data) for Channel #2	C1- 40/28 C2- 40/31	4.375	Receive 20 for C2	C1- 5.0/2.5 C2- 5.0/0
5	Receive 0 bytes for Channel #2 Send 17 bytes (17 bytes HCI data) for Channel #2	C1- 40/8 C2- 40/48	5.0	Send 20 for C1	C1- 5.0/5.0 C2- 5.0/0
		C1- 60/8 C2- 40/48	5.625	Receive 20 for C1	C1- 7.5/5.0 C2- 5.0/0
6	Receive 17 bytes (3 header, 14 data) for Channel #1 Send 17 bytes (3 header, 14 data) for Channel #1	C1- 46/22 C2- 40/28	6.25	Send 20 for C2	C1- 7.5/5.0 C2- 5.0/2.5
		C1- 46/22 C2- 60/28	6.875	Receive 20 for C2	C1- 7.5/5.0 C2- 7.5/2.5
7	Receive 17 bytes (17 bytes data) for Channel #1 Send 17 bytes (17 bytes HCI data) for Channel #1	C1- 29/19 C2- 60/28	7.5	Send 20 for C1	C1- 7.5/7.5 C2- 7.5/2.5
8	Receive 17 bytes (17 bytes data) for Channel #1 Send 17 bytes (17 bytes HCI data) for Channel #1	C1- 32/36 C2- 60/28	8.125	Receive 20 for C1	C1- 10/7.5 C2- 7.5/2.5
		C1- 32/36 C2- 60/8	8.75	Send 20 for C2	C1- 10/7.5 C2- 7.5/5.0
9	Receive 17 bytes (3 header, 14 data) for Channel #2 Send 17 bytes (3 header, 14 data) for Channel #2	C1- 32/36 C2- 54/22	9.375	Receive 20 for C2	C1- 10/7.5 C2- 10/5.0
10	Receive 17 bytes (17 bytes data) for Channel #2 Send 17 bytes (17 bytes HCI data) for Channel #2	C1- 32/16 C2- 37/39	10	Send 20 for C1	C1- 10/10 C2- 10/5.0
		C1- 52/16 C2- 37/39	10.625	Receive 20 for C1	C1- 12.5/10 C2- 10/5.0

Table 2.2:

Time (ms	USB data (header refers to HCI header) (Receive & Send from the host)	Queued data (read / write)	Time (ms)	Air data	Amount Received / Sent (ms)
11	Receive 17 bytes (17 bytes data) for Channel #2 Send 17 bytes (17 bytes HCI data) for Channel #2	C1- 52/16 C2- 20/36	11.25	Send 20 for C2	C1- 12.5/10 C2- 10/7.5

Table 2.2:

2.2 CONTROL ENDPOINT EXPECTATIONS

Endpoint 0 is used to configure and control the USB device. Endpoint 0 will also be used to allow the host to send HCI-specific commands to the host controller. When the USB firmware receives a packet over this endpoint that has the Bluetooth class code, it should treat the packet as an HCI command packet.

2.3 BULK ENDPOINTS EXPECTATIONS

Data integrity is a critical aspect for ACL data. This, in combination with bandwidth requirements, is the reason for using a bulk endpoint. Multiple 64-byte packets can be shipped, per millisecond, across the bus.

Suggested bulk max packet size is 64 bytes. Bulk has the ability to transfer multiple 64-byte buffers per one millisecond frame, depending on available bus bandwidth.

Bulk has the ability to detect errors and correct them. Data flowing through this pipe might be destined for several different slaves. In order to avoid starvation, a flow control model similar to the shared endpoint model is recommended for the host controller.

2.4 INTERRUPT ENDPOINT EXPECTATIONS

An interrupt endpoint is necessary to ensure that events are delivered in a predictable and timely manner. Event packets can be sent across USB with a guaranteed latency.

The interrupt endpoint should have an interval of 1 ms.

The USB software and firmware requires no intimate knowledge of the events passed to the host controller.

2.5 ISOCHRONOUS ENDPOINTS EXPECTATIONS

These isochronous endpoints transfer SCO data to and from the host controller of the radio.

Time is the critical aspect for this type of data. The USB firmware should transfer the contents of the data to the host controllers' SCO FIFOs. If the FIFOs are full, the data should be overwritten with new data.

These endpoints have a one (1) ms interval, as required by Chapter 9 of the USB Specification, Versions 1.0 and 1.1.

The radio is capable of three (3) 64Kb/s voice channels (and can receive the data coded in different ways – 16-bit linear audio coding is the method that requires the most data). A suggested max packet size for this endpoint would be at least 64 bytes. (It is recommended that max packet sizes be on power of 2 boundaries for optimum throughput.) However, if it is not necessary to support three voice channels with 16-bit coding, 32 bytes could also be considered an acceptable max packet size.

3 CLASS CODE

A class code will be used that is specific to all USB Bluetooth devices. This will allow the proper driver stack to load, regardless of which vendor built the device. It also allows HCI commands to be differentiated from USB commands across the control endpoint.

The class code (bDeviceClass) is 0xE0 – Wireless Controller.

The SubClass code (bDeviceSubClass) is 0x01 – RF Controller.

The Protocol code (bDeviceProtocol) is 0x01 – Bluetooth programming.

4 DEVICE FIRMWARE UPGRADE

Firmware upgrade capability is not a required feature. But if implemented, the firmware upgrade shall be compliant with the "Universal Serial Bus Device Class Specification for Device Firmware Upgrade" (version 1.0 dated May 13, 1999) available on the USB Forum web site at http://www.usb.org.

5 LIMITATIONS

5.1 POWER SPECIFIC LIMITATIONS

Today, the host controller of USB-capable machines resides inside a chip known as PIIX4. Unfortunately, because of errata, the USB host controller will not receive power while the system is in S3 or S4. This means that a USB wake-up can only occur when the system is in S1 or S2.

Another issue with the USB host controller is that, while a device is attached, it continually snoops memory to see if there is any work that needs to be done. The frequency that it checks memory is 1ms. This prevents the processor from dropping into a low power state known as C3. Because the notebook processor is not able to enter the C3 state, significant power loss will occur. This is a real issue for business users – as a typical business user will spend almost 90% of their time in the C3 state.

5.2 OTHER LIMITATIONS

Data corruption may occur across isochronous endpoints. Endpoints one and two may suffer from data corruption.

USB provides 16-CRC on all data transfers. The USB has a bit error rate of 10^{-13}.

Note that when a dongle is removed from the system, the radio will lose power (assuming this is a bus-powered device). This means that devices will lose connection.

Part H:3

HCI RS232 TRANSPORT LAYER

An addendum to the HCI document

This document describes the RS232 transport layer (between the Host and the Host Controller). HCI command, event and data packets flow through this layer, but the layer does not decode them.

CONTENTS

1 General 802

2 Overview 803

3 Negotiation Protocol 804

4 Packet Transfer Protocol 808

5 Using delimiters with COBS for synchronization 809

5.1 Using delimiters with COBS and CRC, Protocol Mode 0x13 ... 809

5.2 Frame Format 810

5.3 Error Message Packet 810

5.4 Consistent Overhead Byte Stuffing 811

6 Using RTS/CTS for Synchronization 812

6.1 Using RTS/CTS for Sync without CRC, Protocol Mode 0x14 .. 812

6.2 Error Message Packet 813

6.3 Example of signalling 814

6.4 Control Flow Examples 815

6.4.1 Case 1, Normal Recovery Process 815

6.4.2 Case 2, Both sides detect an error simultaneously 815

6.4.3 Case 3, Error Message with an error 816

7 References 818

1 GENERAL

The objective of the HCI RS232 Transport Layer is to make it possible to use the Bluetooth HCI over one physical RS232 interface between the Bluetooth Host and the Bluetooth Host Controller.

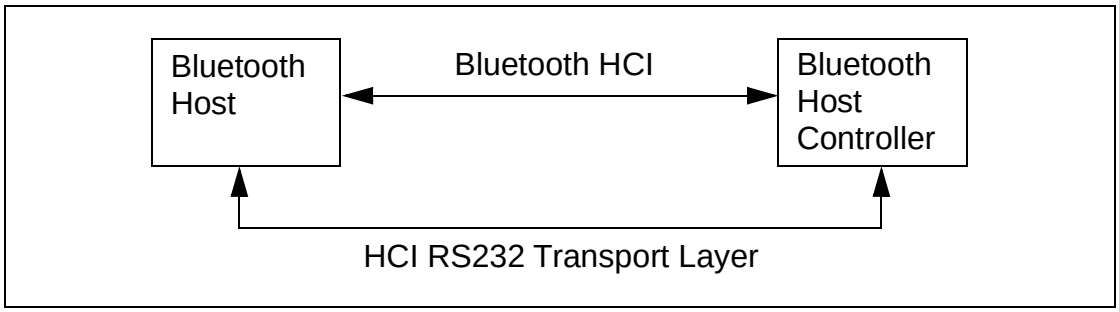


Figure 1.1:

2 OVERVIEW

There are four kinds of HCI packets that can be sent via the RS232 Transport Layer; i.e. HCI Command Packet, HCI Event Packet, HCI ACL Data Packet and HCI SCO Data Packet (see "Host Controller Interface Functional Specification" on page 535). HCI Command Packets can only be sent to the Bluetooth Host Controller, HCI Event Packets can only be sent from the Bluetooth Host Controller, and HCI ACL/SCO Data Packets can be sent both to and from the Bluetooth Host Controller.

However, HCI does not provide the ability to differentiate the four HCI packet types. Therefore, if the HCI packets are sent via a common physical interface, a HCI packet indicator has to be added according to the Table 2.1 below.

HCI packet type	HCI packet indicator
HCI Command Packet	0x01
HCI ACL Data Packet	0x02
HCI SCO Data Packet	0x03
HCI Event Packet	0x04
Error Message Packet*	0x05
Negotiation Packet*	0x06

Table 2.1: HCI RS232 Packet Header

In addition to those four HCI packet types, two additional packet types are introduced to support dynamic negotiation and error reporting. The Error Message Packet (0x05) is used by the receiver to report the nature of error to the transmitting side. The Negotiation Packet (0x06) is used to negotiate the communication settings and protocols.

The HCI packet indicator shall be followed by an 8-bit sequence number that is incremented by 1 every time any of the above packets are sent, except when the retransmission packets are sent as a part of the error recovery. The HCI packet shall immediately follow the sequence number field. All four kinds of HCI packets have a length field, which is used to determine how many bytes are expected for the HCI packet. The Error Message Packet and Negotiation Packet are fixed-length packets, although the negotiation packet can be extended up to 7 more bytes, based on the number in the extension field.

The frame of the basic RS232 Transport Packet is shown below.

LSB *MSB*

Packet Type (8-bit)	**SEQ No (8-bit)**	**HCI Packet or Error Message/Negotiation Packet payload**

The least significant byte is transmitted first.

3 NEGOTIATION PROTOCOL

Before sending any bytes over the RS232 link, the baud rate, parity type, number of stop bit and protocol mode should be negotiated between the Host Controller and the Host. Tdetect is the maximum time required for the transmitter to detect the CTS state change, plus the time it takes to flush the transmit buffer if RTS/CTS is used for error indication and re-synchronization. Otherwise, Tdetect represents the local-side interrupt latency. Host will first send a negotiation packet with the maximum suggested values, plus Host's Tdetect value with Ack code = 000b at the default UART settings specified below, using protocol mode = 0x13. At the same time, the Host Controller side also sets its UART settings to the same initiating parameters and waits for the negotiation packet from the Host.
If the Host Controller side can accept the suggested values from the Host, it sends back the negotiation packet with the same UART setting values, plus Host Controller's Tdetect value with Ack code = 001b. Then, the Host sends back the negotiation packet with the same UART setting values, plus Host's Tdetect with Ack code = 001b as the final acknowledgment, and then sets its Host's UART to the new value. After it has received the final acknowledgment packet from the Host, the Host Controller also changes its UART setting to the new values.

On the other hand, if the Host Controller side cannot accept the suggested value, it should send a set of new suggested values and its own Tdetect value with Ack code = 010b. Each side should continue these steps until both sides receive the accepted Ack code value. Error detection and error recovery during the initial negotiation are performed in the same manner as described in Section 5 on page 808 (Protocol Mode 0x13)

The negotiation phase can be initiated again by either side at any time in order to renegotiate the new values, or just to inform the new Tdetect time. When the negotiation is reinitiated during the data transfer, it should use the previously negotiated settings to exchange the new parameters rather than using the default values.

The initiating parameters:

baud rate: 9600 bps

parity type: no parity

number of data bit: 8 (Note: Only 8-bit data length is allowed.)

number of stop bit: 1

protocol mode: 0x13 (HDLC like framing with COBS/CCITT-CRC)

The negotiation packet format:

LSB *MSB*

Packet Type header 0x06 (8 bits)	SEQ No (8 bits)	UART Settings and ACK (8 bits)	Baud Rate (16 bits)	Tdetect Time (16 bits)	Protocol Mode (8 bit)

SEQ No:

This is an 8-bit number that is incremented by 1 each time a packet is transmitted, excluding the retransmission packet. The unsigned Little Endian format is used.

UART Settings and ACK Field

Bit 0-1	Bit 2	Bit 3	Bit 4	Bit 5-7
Reserved	Stop bit (1 bit)	Parity Enable (1 bit)	Parity Type (1 bit)	Ack Code (3 bits)

Stop Bit:

0: 1 stop bit
1: 2 stop bits

Parity Enable:

0: No parity
1: Parity

Parity Type:

0: Odd Parity
1: Even Parity

Ack Code:

000b: Request
001b: Accepted
010b: Not accepted with new suggested values
011b-111b: Reserved

Baud Rate:

N should be entered for baud rate where
Baud rate = 27,648,000 / N
N=0 is invalid

Maximum possible rate is therefore 27.648Mbps
Minimum possible rate is therefore 421.88bps

The unsigned Little Endian format is used, and the least significant byte should be transmitted first.

Tdetect Time:

This 16-bit field should be filled with the maximum required time for the transmitter to detect the CTS change, plus the time it takes to flush the transmit FIFO if RTS and CTS are used for resynchronization. Otherwise, it should be filled with the local interrupt latency.

The unit of time should be specified in 100 microseconds.

The unsigned Little Endian format is used, and the least significant byte should be transmitted first.

Protocol Mode

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
CRC Used	Delimiter Used	RTS /CTS used	RTS/CTS Mode	Error Recovery Used	Ext0	Ext1	Ext2

CRC Used:

0: CRC-CCITT is not attached at the end of the packet.

1: CRC-CCITT is attached at the end of the packet. (Default)

16-bit CRC can be used with either RTS/CTS or delimiters, although this specification only describes a case when it is used with delimiters.
Generator Polynomial = $x^{16}+x^{12}+x^5+1$

Delimiter Used:

0: Delimiter, 0x7E, is not used.

1: Delimiter, 0x7E, is used with COBS. (Default)
If this bit is set t o 1, RTS/CTS should not be used for the error indication and resynchronization. By the same token, if RTS/CTS is used for the error indication and resynchronization, delimiter should not be used.

RTS/CTS Used:

0: RTS/CTS is not used. (Default)

1: RTS/CTS is used.

RTS/CTS Mode:

0: RTS/CTS is used for Error indication and resynchronization. (Default)

1: RTS/CTS is used for hardware flow control. Please refer to “HCI UART Transport Layer” on page 819 for details.

Error Recovery Used:

0: Error Recovery is not supported.
Even if Error Recovery is not supported, Error Message has to be sent.

1: Error Recovery is supported. (Default)
Error Recovery retransmits the packet with error and all subsequent packets if RTS/CTS are used for synchronization. On the other hand, if 0x7E is used as a delimiter with COBS as a synchronization mechanism, then the error recovery retransmits only the packet with error. Please refer to following sections for details. Error Recovery may not always succeed due to repeated retransmissions, time-outs, retransmit buffer limitations, etc.

Ext2,Ext1,Ext0:

These three bits indicate the number of extra bytes attached to the negotiation packet for future expansion.

4 PACKET TRANSFER PROTOCOL

The packet can be transferred with parity enabled or disabled, and with or without CRC – depending on the environment – as a mechanism to detect the error.

As a synchronization mechanism, one can select either RTS/CTS, or delimiters. Usage of RTS/CTS reduces the computation time for COBS encoding, but it requires two extra copper wires which may not be suitable in some applications. If three-wire cable must be used, or programmable RTS and CTS are not available, delimiter, 0x7E, can be used with COBS.

However, error recovery for these two alternatives may differ slightly. If the RTS/CTS is used for resynchronization, it would be simpler to retransmit all the packets, starting with the packet that had an error. If delimiters are used, the transmitter should retransmit only the packet with an error. The error recovery can be disabled, but the error message packet should still be sent to the transmitter side when the receiver side detects an error.

The HCI RS232 transport layer always uses a data length of 8 bits, and this specification assumes the Little Endian format. Furthermore, the least significant byte should be transmitted first.

The Host Controller may choose to support only one protocol mode, but the Host should be able to support any combination.

Two common schemes (Protocol mode = 0x13 and 0x14) are defined in the following sections to illustrate the usage of each mode.

5 USING DELIMITERS WITH COBS FOR SYNCHRONIZATION

This section illustrates how delimiters with COBS are used for synchronization and how error recovery procedure is performed if delimiters are used as a mechanism to synchronize. This is described using protocol mode 0x13.

5.1 USING DELIMITERS WITH COBS AND CRC, PROTOCOL MODE 0X13

In case RTS/CTS are not available, or if they are hard-wired to be used as a hardware flow control, the HDLC-like framing with the 16-bit CRC (CRC-CCITT) and delimiter 0x7E with COBS (Consistent Overhead Byte Stuffing) [2] are used as a means to detect an error and to resynchronize.

The CRC-CCITT uses the following generator polynomial for 16-bit checksum: $x^{16}+x^{12}+x^5+1$. The 16-bit CRC should be attached to the end of the packet, but right before the ending delimiter, 0x7E. The beginning delimiter, 0x7E, should be followed by the packet type indicator field.

The Consistent Overhead Byte Stuffing is a recent proposal to PPP that yields less than 0.5% overhead, regardless of the data pattern. It uses two steps to escape the delimiter, 0x7E. The first step is eliminating zeros and then replacing all 0x7E with 0x00 between the beginning and ending delimiters.

A simple error recovery scheme is adapted here to minimize the overhead of supporting the error recovery. When the receiving end detects any error, it should send the error message packet with an error type back to the transmitting side. This error message packet will contain a Sequence Number with Error field (SEQ No with Error). The number in the SEQ No with Error field is the sequence number of the packet to be transmitted except for the error type 0x81.The Sequence Number field (SEQ No) that is on every packet is an 8-bit field that is incremented by 1 each time any type of packet is transmitted, except for the retransmission packets. The retransmitted packets should contain the original sequence number in the SEQ No field.

If the error recovery is enabled as in this case with the protocol mode 0x13, the transmitting side shall retransmit only the HCI packet with the sequence number that is indicated by the SEQ No with Error field. It is the responsibility of the receiving end to reorder the packets in the right order as well as to filter out the correctly received duplicate transmission or retransmission. If the transmitting side doesn't have the packet with the correct sequence number in the retransmission holding buffer, it should send the error message packet with the Error Type equal to 0x81 and SEQ No with Error field with the missing sequence number for the retransmission packet, so that the receiving end can detect missing packets. In this case the full error recovery cannot be performed. However, the receiving side can at least detect the loss of packets.

The receiving side should wait at least 4 times the sum of remote Tdetect, local Tdetect and the transmission time of the error message packet, plus the retransmission packet, before it times out when it is waiting for the retransmission packet. When it times out, the receiver has an option of re-requesting it by sending another error message packet with error type = 0x09, or simply dropping it and reporting it to the higher layer.

5.2 FRAME FORMAT

BOF(0x7E), CRC-CCITT, and EOF(0x7E) are added as shown below to those basic packets described in this document. When the CRC is transmitted, the least significant byte should be transmitted first.

LSB *MSB*

0x7E BOF (8 bits)	**Packet Type (8 bits)**	**SEQ No (8 bits)**	**….Payload….**	**CRC (16 bits)**	**0x7E EOF (8 bits)**

5.3 ERROR MESSAGE PACKET

The error-message packet format is the following:

LSB *MSB*

Packet Type, 0x05 (8-bit field)	**Sequence No (8-bit field)**	**Error Type (8-bit field)**	**SEQ No with Error (8-bit field)**

Error Type	Description
0x00	Reserved
0x01	Overrun Error
0x02	Parity Error
0x03	Reserved
0x04	Framing Error
0x05-0x07	Reserved
0x08	CRC Error
0x09	Missing SEQ No
0x0A-0x80	Reserved
0x81	Missing Retransmission Packet
0x82- 0xFF	Reserved

Table 5.1: Error Type available

5.4 CONSISTENT OVERHEAD BYTE STUFFING

Code(n)	Followed by	Description
0x00		Unused.
0x01-0xCF	n-1 data bytes	The n-1 data bytes plus implicit trailing zero.
0xD0	n-1 data bytes	The n-1 data bytes without trailing zero.
0XD1		Unused.
0xD2		Reserved for future.
0xD3-0xDF	none	A run of n-0xD0 zeros.
0xE0-0xFE	n-E0 data bytes	The data bytes with two trailing zeros.
0xFF		Unused.

Table 5.2:

The COBS requires two step encodes.

The first step is the zero-elimination step. This step takes place after attaching the 16-bit CRC if CRC is enabled, but before adding the beginning and ending delimiters, 0x7E. Each COBS code block consists of the COBS code followed by zero or more data bytes. Code bytes 0x00, 0xD1, 0xD2 and 0xFF are never used. The COBS zero-elimination procedure searches the packet for the first occurrence of value zero. To simplify the encoding, a zero is added temporarily at the end of the packet, after the CRC, as a temporary place holder. The number of octets up to and including the first zero determines the code to be used. If this number is 207 or less, then the number itself is used as a COBS code byte, followed by the actual non-zero data bytes themselves, excluding the last byte, which is zero. On the other hand, if the number is more than 207, then the code byte 0xD0 is used, followed by the first 207 non-zero bytes. This process is repeated until all of the bytes of the packet, including the temporary place-holding zero at the end, have been encoded. If a pair of 0x00 is detected after 0 to 30 non-zero octets, the count of octets plus 0xE0 is used as the COBS code, followed by the non-zero octets, excluding the pair of zeros. If a run of three to fifteen 0x00 octets are detected, then the count of these 0x00 octets, plus 0xD0, is used as the code, followed by no other bytes.

The second step is replacing 0x7E with 0x00. The two steps can be done together in a loop, to reduce the encoding time.

For more details and a reference code, please refer to “Consistent Overhead Byte Stuffing ” by J. Carlson et al [2].

6 USING RTS/CTS FOR SYNCHRONIZATION

This section illustrates how RTS and CTS are used to resynchronize, and how error-recovery procedure is performed if RTS and CTS are used as a mechanism to synchronize. This is described using protocol mode 0x14.

6.1 USING RTS/CTS FOR SYNC WITHOUT CRC, PROTOCOL MODE 0X14

The flow of HCI packet transfer is handled by two MODEM control/status signals, -RTS and -CTS. -RTS and -CTS are connected in a null MODEM fashion, meaning that the local-side -RTS should be connected to the remote-side -CTS, and the local-side -CTS should be connected to the remote-side -RTS. These MODEM control/status signals are used to notify the detection of an error to the other side, as well as to resynchronize the beginning of the packet after an error is detected. A very simple error-recovery scheme is adapted here to minimize the overhead of supporting this.

The HCI packet is transmitted only while CTS bit is 1. If the CTS bit changes to 0 during the HCI packet transfer or after the last byte is transmitted, this indicates that there was some error. The receiving end will deassert RTS as soon as it detects any error, and should send the error packet with an error type back to the transmission side. This error packet will contain a Sequence Number with Error field (SEQ No with Error). The number in the SEQ No With Error field is the beginning sequence number of the packets to be retransmitted except for the error type 0x81.The sequence number field that is on every packet is an 8-bit field that is incremented by 1 each time any type of packet is transmitted, except for the retransmission packets. The retransmitted packets should contain the original sequence number in the SEQ No field.

When the transmitting end detects CTS bit changing from 1 to 0 at any time, the transmitting end should hold the transmission and wait until the error packet is received before resuming the transmission. When the receiving end is ready to receive the new data, it should assert RTS after the minimum of Tdetect time. Tdetect time is the maximum time required for the transmit side to detect the state change on CTS bit, plus the time it takes to flush the transmit buffer. The Tdetect value of each side should be informed to the other side during the negotiation phase. The local Tdetect value and the remote side Tdetect value together, along with the baud rate, can also be used to estimate the queue length required for the retransmission holding buffer. Before the receiving side asserts RTS line again, it should flush the RX buffer.

If the error recovery is enabled as in this case with the protocol mode 0x14, the transmitting side shall retransmit all the HCI packets starting from the packet with the sequence number indicated by the SEQ No with Error field.

It is the responsibility of the receiving end to reorder the packets in the right order as well as to filter out the correctly received duplicate transmission or retransmission

. Before it retransmits, it should flush the transmit buffer that may hold the left-over data from the aborted previous packet. As it retransmits the packets from the retransmission holding buffer, it should start transmitting the packet with the Sequence Number that matches the SEQ No with Error. If the transmitting side doesn't have the packet with the correct sequence number in the retransmission holding buffer, the transmitter should send an error message packet with error type 0x81, and it should skip to the packet with the sequence number that is available in the buffer. In this case, the full error recovery cannot be performed. However, the receiving side can, at least, detect the loss of packets.

6.2 ERROR MESSAGE PACKET

The error-message packet format is the following:

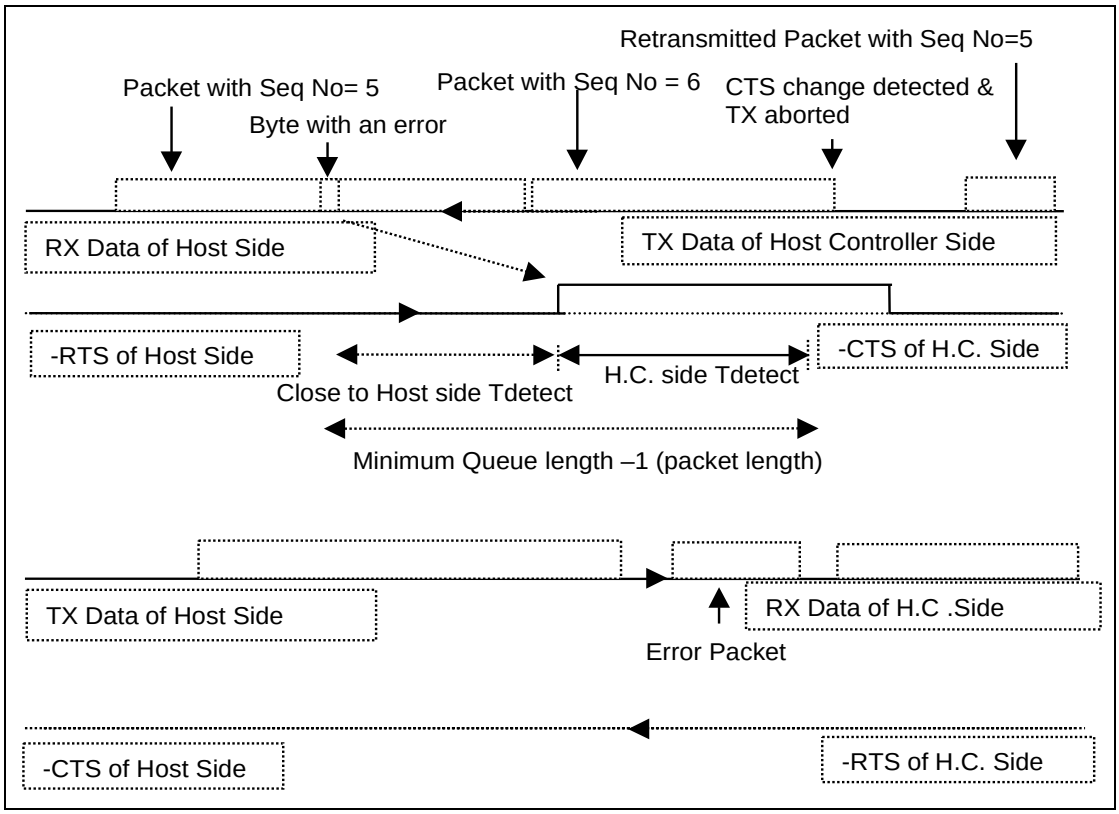
LSB *MSB*

Packet header 0x05 (8-bit field)	**Sequence No (8-bit field)**	**Error Type (8-bit field)**	**SEQ No with Error (8-bit field)**

Error Type	**Description**	**Comment**
0x00	Reserved	
0x01	Overrun Error	
0x02	Parity Error	
0x03	Reserved	
0x04	Framing Error	
0x05-0x07	Reserved	
0x08	CRC Error*	Not applicable In Mode 0x14
0x09	Missing SEQ No	
0x0A-0x80	Reserved	
0x81	Missing Retransmission Packet	
0x82- 0xFF	Reserved	

Table 6.1: Error Type available

6.3 EXAMPLE OF SIGNALLING



6.4 CONTROL FLOW EXAMPLES

6.4.1 Case 1, Normal Recovery Process

Controller Side	Host Side
0) RTS is asserted and the asserted CTS is detected.	0) RTS is asserted and the asserted CTS is detected.
	1) Ctrl/Data[n] is sent out and Ctrl/Data[n] is stored in the retransmission holding buffer.
2) Ctrl/Data[n] is received with an error.	
3) Deasserts RTS	
4a) Error message for [n] is sent and Error message for [n] is stored in the TX retransmit holding buffer. 4b) Empties the RX FIFO and waits for Tdetect (Host) amount of time.	4) Detects CTS deasserted.
	5a) Stops further transmission and waits until the TX FiFO is empty (or Flush the FIFO if it can.) 5b) Error message for [n] is received.
6) Asserts RTS	
.	7) The asserted CTS is detected.
	8) Retransmits Ctrl/Data[n].

6.4.2 Case 2, Both sides detect an error simultaneously

Controller Side	Host Side
0) RTS is asserted and the asserted CTS is detected.	0) RTS is asserted and the asserted CTS is detected.
1) Ctrl/Data[x] is sent and Ctrl/Data[x] is stored in the retransmission holding buffer.	1) Ctrl/Data[n] is sent and Ctrl/Data[n] is stored in the retransmission holding buffer.
2) Ctrl/Data[n] is received with an error.	2) Ctrl/Data[x] is received with an error.
3) Deasserts RTS.	3) Deasserts RTS.
4) Detects CTS deasserted.	4) Detects CTS deasserted
5a) Stops further transmission and waits until the TX FiFO is empty (or Flush the FIFO if it can).	5a) Stops further transmission and waits until the TX FiFO is empty (or Flush the FIFO if it can).
5b) Empties the RX FIFO and waits for Tdetect (Host) amount of time.	5b) Empties the RX FIFO and waits for Tdetect (Controller) amount of time.

Controller Side	Host Side
6) Asserts RTS.	6) Asserts RTS.
7) The asserted CTS is detected.	7) The asserted CTS is detected.
8) Error message for [n] is sent and Error message for [n] is stored in the TX retransmit holding buffer.	8) Error message for [x] is sent and Error message for [x] is stored in the TX retransmit holding buffer.
9) Error message for [x] is received.	9) Error message for [n] is received.
10) Retransmits Ctrl/Data[x].	10) Retransmits Ctrl/Data[n].

6.4.3 Case 3, Error Message with an error

Controller Side	Host Side
0) RTS is asserted and the asserted CTS is detected.	0) RTS is asserted and the asserted CTS is detected.
	1) Ctrl/Data[n] is sent and Ctrl/Data[n] is stored in the retransmission holding buffer.
2) Ctrl/Data[n] is received with an error.	
3) Deasserts RTS.	
4a) Error message for [n] (Err[n]) is sent and Err[n] is stored in the TX retransmit holding buffer. 4b) Empties the RX FIFO and waits for Tdetect (Host) amount of time.	4) Detects CTS deasserted.
	5a) Stops further transmission and waits until the TX FiFO is empty (or Flush the FIFO if it can.) 5b) Error message for [n] is received with an error.
6) Asserts RTS.	6a) Deasserts RTS. 6b) Empties the RX FIFO and waits for Tdetect (Controller) amount of time.
7) Detects CTS deasserted.	
8) Stops further transmission and waits until the TX FiFO is empty (or Flush the FIFO if it can.)	8) The asserted CTS detected.
	9a) Error message for Err[n] is sent and Error message for Err[n] is stored in the retransmission holding buffer. 9b) Asserts RTS.

Controller Side	Host Side
10a) Error message for Err[n] is received. 10b) The asserted CTS detected.	
11) Retransmits Error message for [n].	
	12) Error message for [n] is received.
	13) Retransmit Ctrl/Data[n].

7 REFERENCES

[1] Bluetooth Host Controller Interface Function Specification

[2] Stuart. Cheshire, Mary. Baker,“Consistent Overhead Byte Stuffing” IEEE/ ACM Transactions on Networking, Vol. 7, No. 2, April 1999

[3] Bluetooth HCI UART Transport Layer Specification

[4] CRC-CCITT computation example:
The following was copied from the Network Working Group Request for Comments: 1171.
Here is the code that will generate the look up table for CRC-CCITT.

```
/*
* Generate a FCS table for the HDLC FCS.
* Drew D. Perkins at Carnegie Mellon University.
* Code liberally borrowed from Mohsen Bannan and D. Hugh
Redelmeier.
*/
/* The HDLC polynomial: x**0 + x**5 + x**12 + x**16 (0x8408).    */

#define P       0x8408
main()
{
register unsigned int b, v;
register int i;


printf("typedef unsigned short u16; \n");
printf("static u16 fcstab[256] = {");
for (b=0; ; ) {
if (b % 8 == 0)
printf("\n");
v = b;
for (i=8; i--; )
v = v & 1 ? (v >> 1) ^ P : v >>1;

printf("0x%04x",  v & 0xFFFF);
if (++b == 256)
break;
printf(",");
}
printf("\n};\n");
}
```

Here is the code that computes CRC-CCITT

```
#define PPPINITFCS          0xffff     /* Initital FCS value */
#define PPPGOODFCS          0xf0b8     /* Good final FCS value */
```

```
/* Calculate a new FCS given the current FCS and the new data.  */

u16 pppfcs(fcs, cp, len)
register u16 fcs;
register unsigned char *cp;
register int len;

{
ASSERT(sizeof (u16) == 2);
ASSERT(((u16) -1 > 0);
while (len--)
fcs = (fcs >> 8) ^ fcstab[(fcs ^ *cp++) & 0xff];

return (fcs);
}
```

Part H:4

HCI UART TRANSPORT LAYER

An addendum to the HCI document

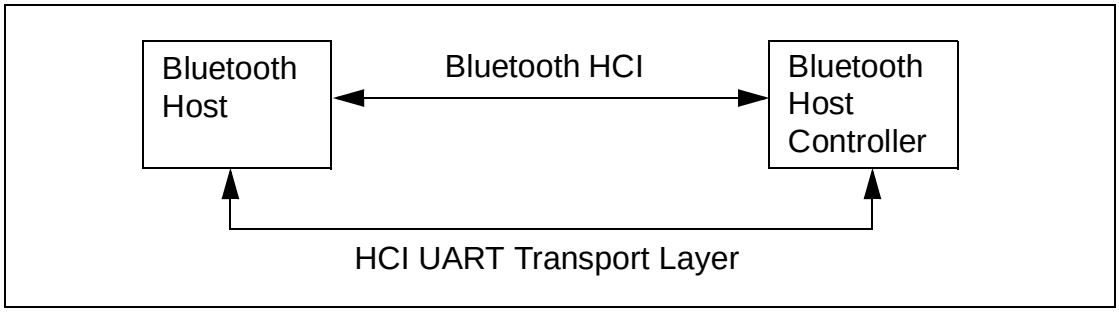
This document describes the UART transport layer (between the Host and the Host Controller). HCI command, event and data packets flow through this layer, but the layer does not decode them.

CONTENTS

1 General 824
2 Protocol 825
3 RS232 Settings 826
4 Error Recovery 827

1 GENERAL

The objective of this HCI UART Transport Layer is to make it possible to use the Bluetooth HCI over a serial interface between two UARTs on the same PCB. The HCI UART Transport Layer assumes that the UART communication is free from line errors. See also "HCI RS232 Transport Layer" on page 797.



2 PROTOCOL

There are four kinds of HCI packets that can be sent via the UART Transport Layer; i.e. HCI Command Packet, HCI Event Packet, HCI ACL Data Packet and HCI SCO Data Packet (see “Host Controller Interface Functional Specification” on page 535). HCI Command Packets can only be sent to the Bluetooth Host Controller, HCI Event Packets can only be sent from the Bluetooth Host Controller, and HCI ACL/SCO Data Packets can be sent both to and from the Bluetooth Host Controller.

HCI does not provide the ability to differentiate the four HCI packet types. Therefore, if the HCI packets are sent via a common physical interface, a HCI packet indicator has to be added according to Table 2.11 below.

HCI packet type	HCI packet indicator
HCI Command Packet	0x01
HCI ACL Data Packet	0x02
HCI SCO Data Packet	0x03
HCI Event Packet	0x04

Table 2.1: HCI packet indicators

The HCI packet indicator shall be sent immediately before the HCI packet. All four kinds of HCI packets have a length field, which is used to determine how many bytes are expected for the HCI packet. When an entire HCI packet has been received, the next HCI packet indicator is expected for the next HCI packet. Over the UART Transport Layer, only HCI packet indicators followed by HCI packets are allowed.

3 RS232 SETTINGS

The HCI UART Transport Layer uses the following settings for RS232:

Baud rate:	**manufacturer-specific**
Number of data bits:	**8**
Parity bit:	**no parity**
Stop bit:	**1 stop bit**
Flow control:	**RTS/CTS**
Flow-off response time:	**manufacturer specific**

Flow control with RTS/CTS is used to prevent temporary UART buffer overrun. It should not be used for flow control of HCI, since HCI has its own flow control mechanisms for HCI commands, HCI events and HCI data.

If CTS is 1, then the Host/Host Controller is allowed to send.
If CTS is 0, then the Host/Host Controller is not allowed to send.

The flow-off response time defines the maximum time from setting RTS to 0 until the byte flow actually stops.

The RS232 signals should be connected in a null-modem fashion; i.e. the local TXD should be connected to the remote RXD and the local RTS should be connected to the remote CTS and vice versa.

4 ERROR RECOVERY

If the Host or the Host Controller lose synchronization in the communication over RS232, then a reset is needed. A loss of synchronization means that an incorrect HCI packet indicator has been detected, or that the length field in an HCI packet is out of range.

If the UART synchronization is lost in the communication from Host to Host Controller, then the Host Controller shall send a Hardware Error Event to tell the Host about the synchronization error. The Host Controller will then expect to receive an HCI_Reset command from the Host in order to perform a reset. The Host Controller will also use the HCI_Reset command in the byte stream from Host to Host Controller to re-synchronize.

If the UART synchronization is lost in the communication from Host Controller to Host, then the Host shall send the HCI_Reset command in order to reset the Host Controller. The Host shall then re-synchronize by looking for the HCI Command Complete event for the HCI_Reset command in the byte stream from Host Controller to Host.

See “Host Controller Interface Functional Specification” on page535 for HCI commands and HCI events.

Part I:1

BLUETOOTH TEST MODE

This document describes the test mode for hardware and low-level functionality tests of Bluetooth devices. The test mode includes transmitter tests (packets with constant bit patterns) and loop back tests.

CONTENTS

1 General Description 832
1.1 Test Setup 832
1.2 Activation 833
1.3 Control 833
2 Test Scenarios 834
2.1 Transmitter Test 834
2.1.1 Packet Format 834
2.1.2 Pseudorandom Sequence 836
2.1.3 Reduced Hopping Sequence 837
2.1.4 Control of Transmit Parameters 837
2.1.5 Power Control 838
2.1.6 Switch Between Different Frequency Settings 838
2.2 LoopBack Test 838
2.3 Pause Test 842
3 Outline of Proposed LMP Messages 843
4 References 846

1 GENERAL DESCRIPTION

The test mode supports testing of the Bluetooth transmitter and receiver. It is intended mainly for certification/compliance testing of the radio and baseband layer, and may also be used for regulatory approval or in-production and after-sales testing.

A device in test mode must not support normal operation. For security reasons the test mode is designed such that it offers no benefit to the user. Therefore, no data output or acceptance on a HW or SW interface is allowed.

1.1 TEST SETUP

The setup consists of a device under test (DUT) and a tester. Optionally, additional measurement equipment may be used.

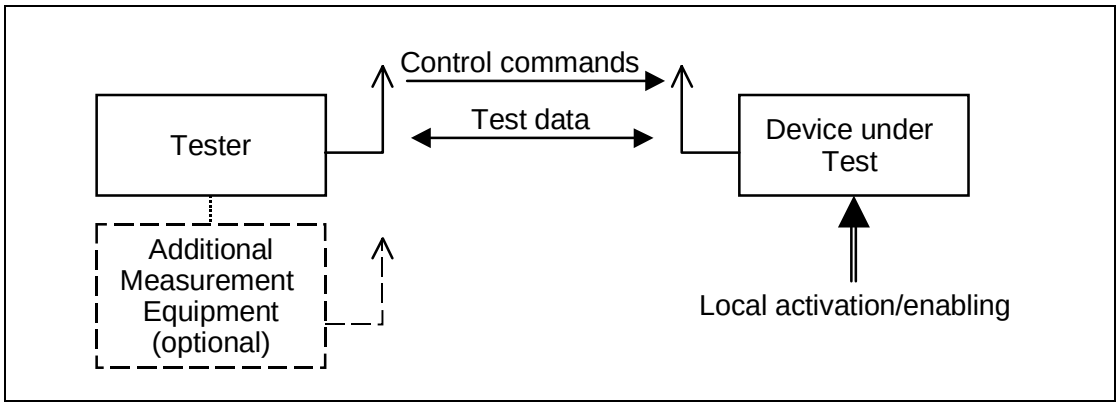


Figure 1.1: Setup for Test Mode

Tester and DUT form a piconet where the tester acts as master and has full control over the test procedure. The DUT acts as slave.

The control is done via the air interface using LMP commands (see Section 3 on page 842 and “Link Manager Protocol” on pag e183). Hardware interfaces to the DUT may exist, but are not subject to standardization.

The test mode is a special state of the Bluetooth model. For security and type approval reasons, a device in test mode may not support normal operation. When the DUT leaves the test mode it enters the standby state. After power-off the Bluetooth device must return to standby state.

1.2 ACTIVATION

The activation may be carried out locally (via a HW or SW interface), or using the air interface.

- For activation over the air interface, entering the test mode must be locally enabled for security and type approval reasons. The implementation of this local enabling is not subject to standardization.

  The tester sends an LMP command that forces the DUT to enter test mode. The DUT terminates all normal operation before entering the test mode.

  The DUT shall return an LMP_Accepted on reception of an activation command. LMP_Not_Accepted shall be returned if the DUT is not locally enabled.

- If the activation is performed locally using a HW or SW interface, the DUT terminates all normal operation before entering the test mode.

  Until a connection to the tester exists, the device shall perform page scan and inquiry scan. Extended scan activity is recommended.

1.3 CONTROL

Control and configuration is performed using special LMP commands (see Section 3 on page 842). These commands must be rejected if the Bluetooth device is not in test mode. In this case, an LMP_not_accepted is returned. The DUT shall return an LMP_accepted on reception of a control command when in test mode.

A Bluetooth device in test mode must ignore all LMP commands not related to control of the test mode. LMP commands dealing with power control and the request for LMP features (LMP_features_req) are allowed in test mode; the normal procedures are also used to test the adaptive power control.

The DUT can be commanded to leave the test mode by an LMP_Detach command or by sending an LMP_test_control command with test scenario set to ’exit test mode’.

2 TEST SCENARIOS

2.1 TRANSMITTER TEST

The Bluetooth device transmits a constant bit pattern. This pattern is transmitted periodically with packets aligned to the slave TX timing of the piconet formed by tester and DUT. The same test packet is repeated for each transmission.

The transmitter test is started when the master sends the first POLL packet. In non-hopping mode agreed frequency is used for this POLL packet.

The tester transmits at his TX slots (control commands or POLL packets). The slave starts burst transmission in the following slave TX slot. The master's polling interval is fixed and defined beforehand. The device under test may transmit its burst according to the normal timing even if no packet from the tester was received. In this case, it is highly recommended that the ARQN bit is set to NAK.

The burst length may exceed the length of a one slot packet. In this case the tester may take the next free master TX slot for polling. The timing is illustrated in Figure 2.1.

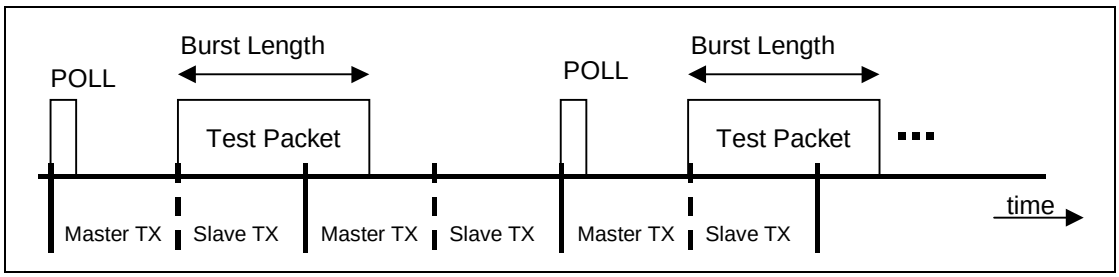


Figure 2.1: Timing for Transmitter Test

2.1.1 Packet Format

The test packet is a normal Bluetooth packet, see Figure 2.2. For the payload itself see below.

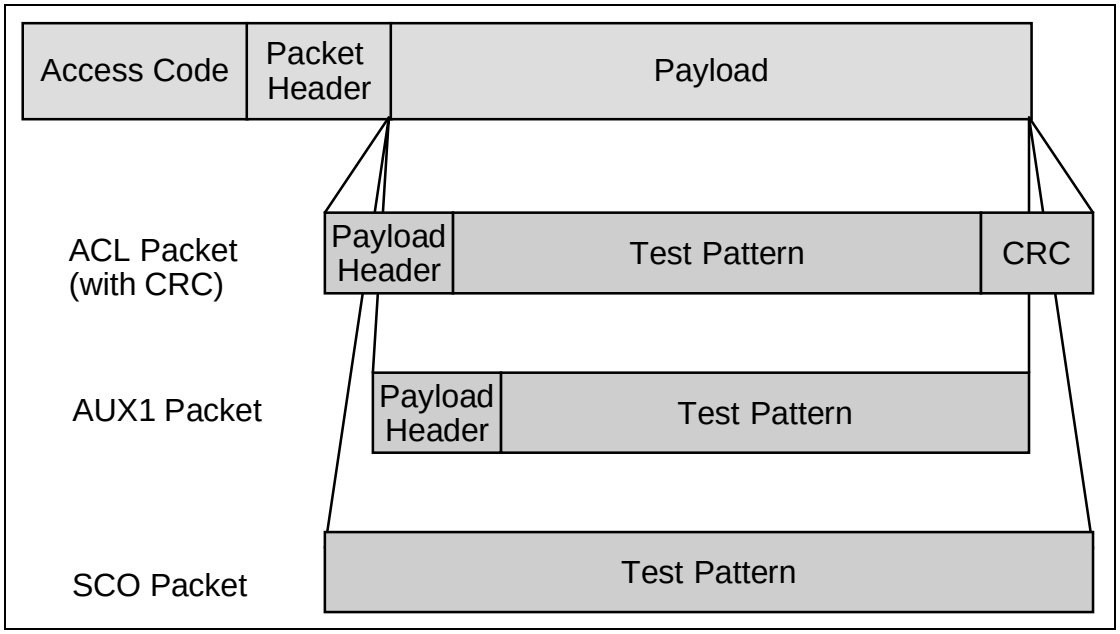


Figure 2.2: General Format of TX Packet

During configuration the tester defines:

- the packet type to be used
- payload length

For the payload length, the restrictions from the baseband specification apply (see “Baseband Specification” on page 33.). In case of ACL packets the payload structure defined in the baseband specification is preserved as well, see Figure 2.2.

For the transmitter test mode, only packets without FEC should be used; i.e. HV3, DH1, DH3, DH5 and AUX1packets. Support of packet type is only mandatory up to the longest implemented packet type.

In transmitter test mode, the packets exchanged between tester and DUT are not scrambled with the whitening sequence. Whitening is turned off when the DUT has accepted to enter the transmitter test mode, and is turned on when the DUT has accepted to exit the transmitter test mode, see Figure 2.3.[1]

1. Note: Implementations must ensure that retransmissions of the LMP_Accepted messages use the same whitening status.

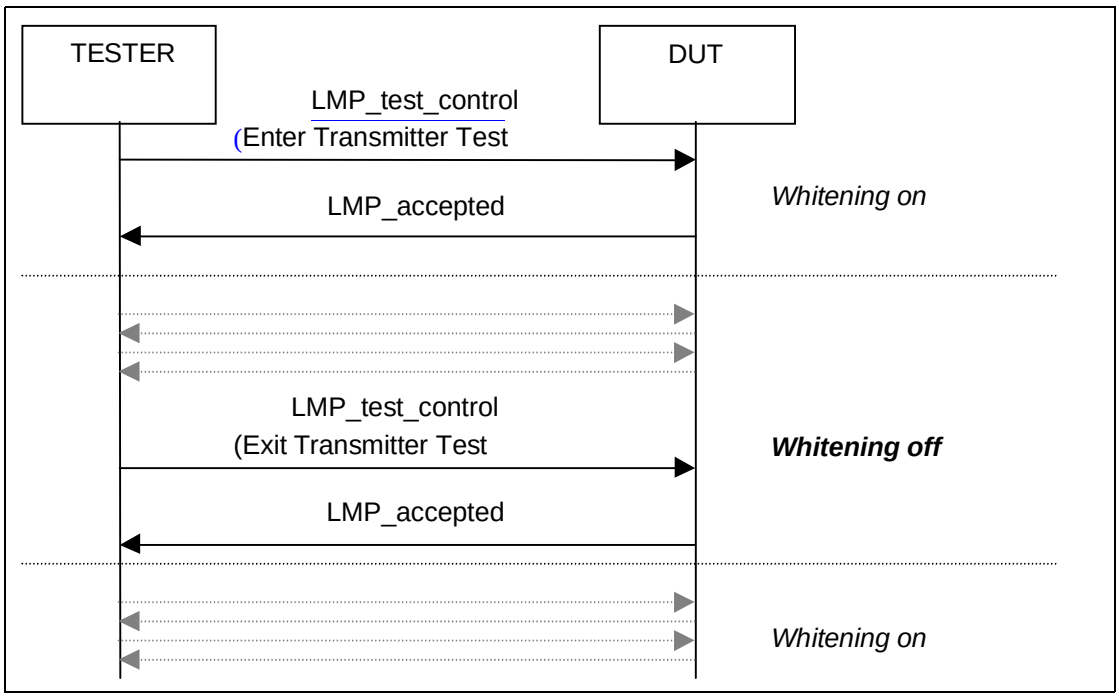


Figure 2.3: Use of whitening in Transmitter mode

2.1.2 Pseudorandom Sequence

In case of pseudorandom bit sequence, the same sequence of bits is used for each transmission (i.e. the packet is repeated, see above). A PRBS-9 Sequence[2] is used, see [2] and [3].

The properties of this sequence are as follows (see [3]). The sequence may be generated in a nine-stage shift register whose 5th and 9th stage outputs are added in a modulo-two addition stage (see Figure 2.4), and the result is fed back to the input of the first stage. The sequence begins with the first ONE of 9 consecutive ONEs; i.e. the shift register is initialized with nine ones.

- Number of shift register stages: 9
- Length of pseudo-random sequence: $2^9-1 = 511$ bits
- Longest sequence of zeros: 8 (non-inverted signal)

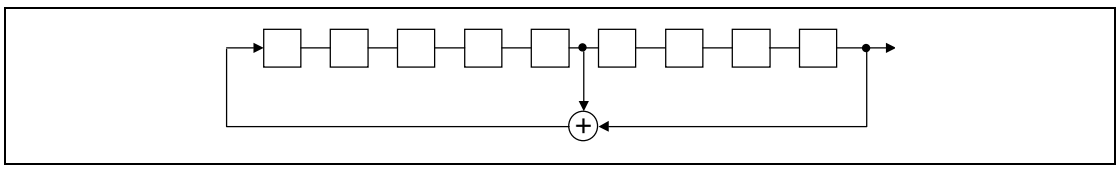


Figure 2.4: Linear Feedback Shift Register for Generation of the PRBS sequence

2. Some uncertainties about Japanese regulatory requirements have been reported. If necessary for regulatory type approval in Japan, some features might be added; e.g. a longer PN sequence.

2.1.3 Reduced Hopping Sequence

To support quick testing of the radio over the complete frequency range, a reduced hopping mode is defined. Implementation of this mode is optional for Bluetooth devices and modules.

Reduced hopping uses only five frequencies, on which a sequential hopping is done (channels 0, 23, 46, 69 and 93 are used[3]), see Figure 2.5.

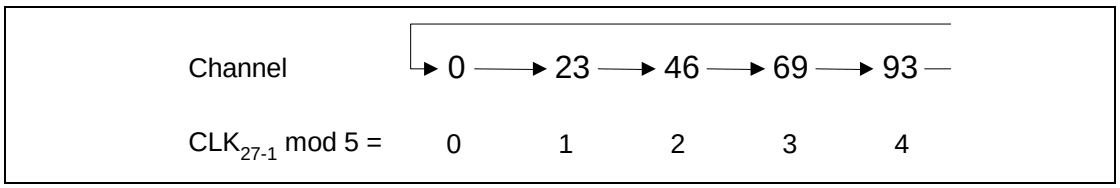


Figure 2.5: Reduced hopping scheme

The timing is based on the Bluetooth clock of the tester. The value of CLK_{27-1} (i.e. not using CLK_0, representing half slots) modulo 5 is used to determine the transmit frequency.

2.1.4 Control of Transmit Parameters

The following parameters can be set to configure the transmitter test:

1. Bit pattern:
   - Constant zero
   - Constant one
   - Alternating 1010...[4]
   - Alternating 1111 0000 1111 0000...[4]
   - Pseudorandom bit pattern
   - Transmission off
2. Frequency selection:
   - Single frequency
   - Hopping Europe/USA
   - Hopping France
   - Reduced Hopping (implementation in Bluetooth devices and modules is optional)
3. TX frequency
   - k ⇒ f := (2402 + k) MHz
4. Default poll period in TDD frames (n * 1.25 ms)

3. The range is chosen to test the whole frequency range, which covers the normal 79 channels, as well as the French hopping scheme. The frequency assignment rule is the same as for the fixed TX frequency: f = (2402 + k) MHz.
4. It is recommended that the sequence starts with a one; but, as this is irrelevant for measurements, it is also allowed to start with a zero.

5. Packet Type
6. Length ofTest Sequence (user data of packet definition in Baseband Specification on pag e33.)

2.1.5 Power Control

If adaptive power control is tested, the normal LMP commands will be used. The DUT starts to transmit at the maximum power and reduces/increases its power by one step on every command received.

2.1.6 Switch Between Different Frequency Settings

A change in the frequency selection becomes effective when the LMP procedure is completed:

When the tester receives the LMP_accepted it then transmits Poll packets containing the Ack for at least 8 slots (4 transmissions). When these transmissions have been completed the tester moves to the new frequency hop and whitening settings.

After sending LMP_accepted the DUT waits for the LC level Ack for the LMP_accepted. When this is received it moves to the new frequency hop and whitening settings.

There will be an implementation defined delay after sending the LMP_accepted before the TX or loopback test starts. Testers must be able to cope with this.

Note: Loss of the LMP_Accepted packet will eventually lead to a loss of frequency synchronization that cannot be recovered. Similar problems occur in normal operation, when the hopping pattern changes.

2.2 LOOPBACK TEST

The device under test receives normal baseband packets. The received packets are decoded in the DUT, and the payload is sent back using the same packet type. The return packet is sent back in either the TX slot directly following the transmission of the tester, or it is delayed and sent back in the slot after the next transmission of the tester (see Figure 2.7 to Figure 2.9 on page 840).

Alternatively, it is possible to implement a delayed loopback instead. Then the return packet is delayed to the following TX slot. There is no signalling to determine or control the mode. The device behavior must be fixed or adjusted by other means, but must not change randomly.

The tester can select, whether whitening is on or off. This setting holds for both up- and downlink. For switching the whitening status, the same rules as in Section 2.1 on page 833 (Figure 2.3) apply.

The following rules apply (for illustration see Figure 2.6 on page 839):

- Clearly, if the synch word was not detected, there will be no reply.
- If the header error check (HEC) fails, the DUT replies with a NULL packet with the ARQN bit set to NAK. It is not mandatory to return a NULL packet in this case; the DUT may send nothing.
- If the packet contains an LMP message relating to the control of the test mode this command is executed and the packet is not returned, though ACK or NAK is still returned as usual procedure. Other LMP commands are ignored and no packet is returned.
- The payload FEC is decoded and the payload is coded again for transmission. This allows testing of the FEC handling. If the pure bit error rate shall be determined the tester chooses a packet type without FEC.
- The CRC is evaluated. In case of a failure, the payload is returned with ARQN = NAK. The CRC for the return packet is calculated for the returned payload.
- If the CRC fails the number of bytes as indicated in the (possibly erroneous) payload header shall be looped back.

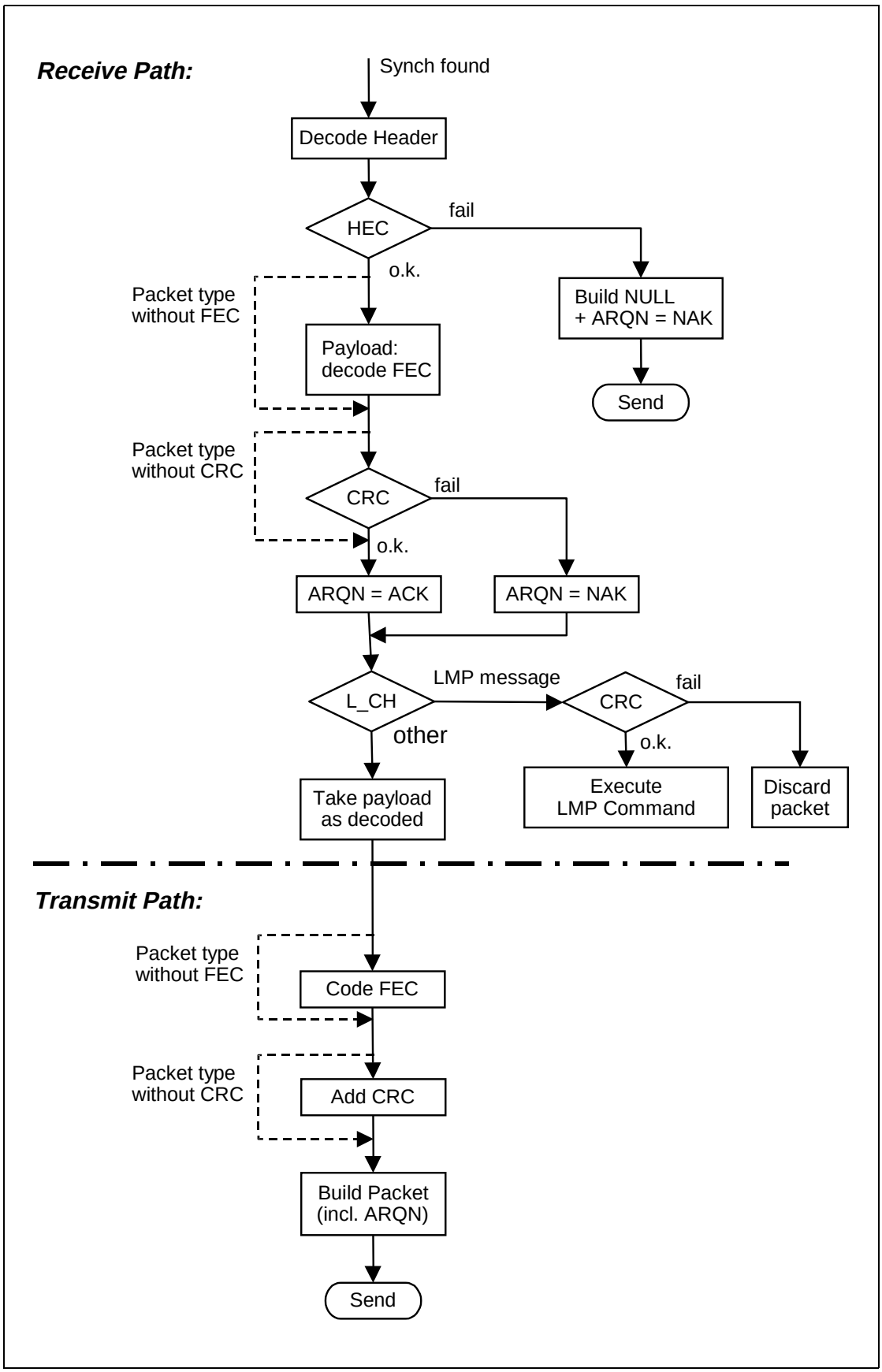


Figure 2.6: DUT Packet Handling in Loop Back Test

The timing for normal and delayed loopback is illustrated in Figure 2.7 to Figure 2.9:

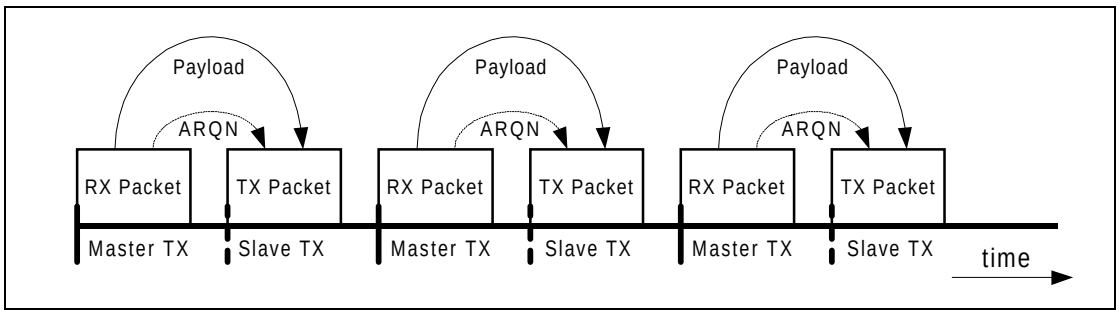


Figure 2.7: Payload & ARQN handling in normal loopback.

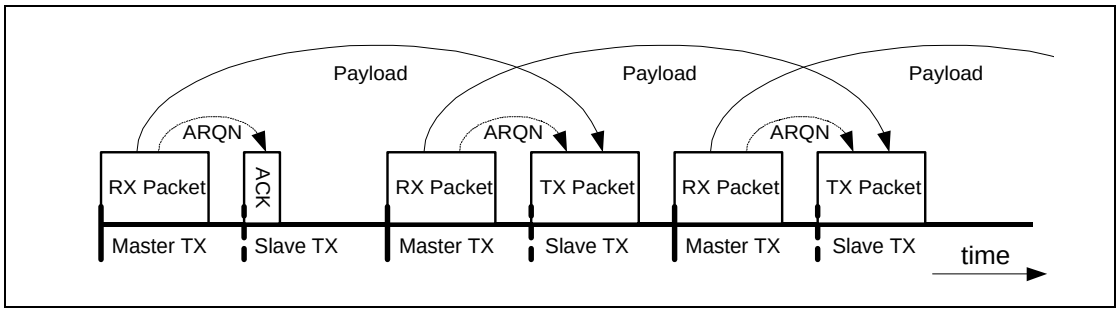


Figure 2.8: Payload & ARQN handling in delayed loopback - start.

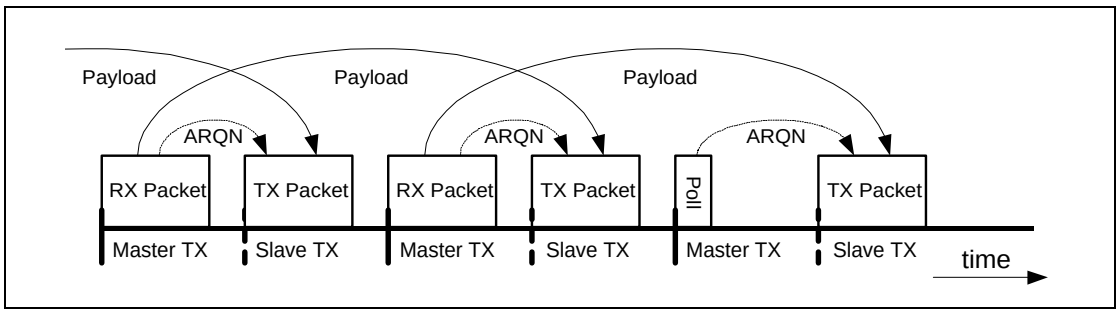


Figure 2.9: Payload & ARQN handling in delayed loopback - end.

The whitening is performed in the same way as it is used in normal active mode.

The following parameters can be set to configure the loop back test:

1. Packet Class[5]
   - ACL Packets
   - SCO Packets
   - ACL Packets without whitening
   - SCO Packets without whitening

5. This is included because, in the future, the packet type numbering may not remain unambiguous.

2. Frequency Selection
   - Single frequency (independent for RX and TX)
   - Hopping Europe/USA
   - Hopping France
   - Hopping reduced (optional)

Hopping reduced uses only five frequencies on which a sequential hopping is done on (channel: 0, 23, 46, 69 and 93 is used).

3. Power level: (To be used according radio specification requirements)
   - power control or fixed TX power

The switch of the frequency setting is done exactly as for the transmitter test (see Section 2.1.6 on page 837).

2.3 PAUSE TEST

Pause test is used by testers when they want the device under test to stop the current loopback or transmitter test and enter a standby mode.

When an LMP_test_control PDU that specifies pause mode is received the device under test stops its current test. In the case of a transmitter test this means that no more packets will be transmitted. While in pause test mode the DUT responds normally to POLL packets i.e. responds with a NULL packet. It also responds normally to all the LMP packets that are allowed in test mode.

When the test scenario is set to Pause Test all the other fields in the LMP_test_control are ignored. There is no change in hopping scheme or whitening as a result of a request to pause test.

3 OUTLINE OF PROPOSED LMP MESSAGES

Table 3.1 lists all LMP messages used for test mode (see Link Manager Protocol, Section 6 on page 246)
To ensure that the contents of LMP_test_control PDU are suitably whitened (important when sent in transmitter mode), all parameters listed in Table 3.2 are XORed with 0x55 before being sent.

LMP PDU	PDU number	Possible Direction	Contents	Position in Payload
LMP_test_activate	56	m → s		
LMP_test_control	57	m → s	test scenario hopping mode TX frequency RX frequency power control mode poll period packet type length of test data	2 3 4 5 6 7 8 9-10
LMP_detach	7	m → s		
LMP_accepted	3	m ← s		
LMP_not_accepted	4	m ← s		

Table 3.1: LMP messages used for Test Mode

Name	Length (bytes)	Type	Unit	Detailed
Test scenario	1	u_int8		0 Pause Test Mode 1 Transmitter test – 0 pattern 2 Transmitter test – 1 pattern 3 Transmitter test – 1010 pattern 4 Pseudorandom bit sequence 5 Closed Loop Back – ACL packets 6 Closed Loop Back – SCO packets 7 ACL Packets without whitening 8 SCO Packets without whitening 9 Transmitter test – 1111 0000 pattern 10–254 reserved 255 Exit Test Mode The value is XORed with 0x55.

Table 3.2: Parameters used in LMP_Test_Control PDU

Name	Length (bytes)	Type	Unit	Detailed
Hopping mode	1	u_int8		0 RX/TX on single frequency 1 Hopping Europe/USA 2 Reserved 3 Hopping France 4 Reserved 5 Reduced Hopping (optional) 6–255 reserved The value is XORed with 0x55.
TX frequency (for DUT)	1	u_int8		f = [2402 + k] MHz The value is XORed with 0x55.
RX frequency (for DUT)	1	u_int8		f = [2402 + k] MHz The value is XORed with 0x55.
Power control mode	1	u_int8		0 fixed TX output power 1 adaptive power control The value is XORed with 0x55.
Poll period	1	u_int8	1.25 ms	The value is XORed with 0x55.
Packet type	1	u_int8		numbering as in packet header, see Baseband Specification) The value is XORed with 0x55.
length of test sequence (=length of user data in Baseband Specification)	2	u_int16	1 byte	unsigned binary number The value is XORed with 0x5555.

Table 3.2: Parameters used in LMP_Test_Control PDU

The control PDU is used for both transmitter and loop back tests. The following restrictions apply for the parameter settings:

Parameter	Restrictions Transmitter Test	Restrictions Loopback Test
TX frequency	0 ≤ k ≤ 93	0 ≤ k ≤ 93
RX frequency	same as TX frequency	0 ≤ k ≤ 93
Poll period		not applicable (set to 0)
Length of test sequence	depends on packet type: DH1: <= 27 bytes DH3: <= 183 bytes DH5: <= 339 bytes AUX1: <= 29 bytes HV3: = 30 bytes	not applicable (set to 0)

Table 3.3: Restrictions for Parameters used in LMP_Test_Control PDU

4 REFERENCES

[1] Bluetooth Link Manager Protocol.

[2] CCITT Recommendation O.153 (1992), Basic parameters for the measurement of error performance at bit rates below the primary rate.

[3] ITU-T Recommendation O.150 (1996), General requirements for instrumentation for performance measurements on digital transmission equipment.

[4] Bluetooth Baseband Specification.

Part I:2

BLUETOOTH COMPLIANCE REQUIREMENTS

This document specifies the requirements for Bluetooth compliance.

CONTENTS

1 Scope 850
2 Definitions 851
3 Legal Aspects 853
4 Introduction to the Bluetooth Qualification Program 854

1 SCOPE

The Bluetooth Promoters and the Bluetooth Adopters have signed the applicable Bluetooth Agreements. These agreements grant Bluetooth Promoters and Bluetooth Adopters a Bluetooth license for "products which comply with the Specification".

This document specifies the requirements which must be met by a Bluetooth Promoter or Bluetooth Adopter to demonstrate that a particular product does "comply with the Specification", thereby qualifying that particular product to be subject to the rights extended by the applicable Bluetooth Agreement.

The Bluetooth Qualification Program is the process by which a Bluetooth Promoter or a Bluetooth Adopter demonstrates that a particular product meets the requirements specified herein. This document provides an introduction to the Bluetooth Qualification Program. The Bluetooth Qualification Program is defined in the Bluetooth Qualification Program Reference Document (PRD) available on the Bluetooth Web site.

Regulatory requirements and governmental type approval requirements are outside the scope of this document.

Requirements on Trademark usage are outside the scope of this document and the Qualification Program. However, successful completion of the Qualification program is a precondition to usage of the Bluetooth Trademark. Details on trademark usage can be found in the applicable Bluetooth agreements and Bluetooth Trademark License Agreement.

2 DEFINITIONS

Bluetooth Agreements - Promoter, Early Adopters or Adopters Agreement, whichever is applicable.

Bluetooth License - All the rights, granted in the Promoter, Early Adopters or Adopters Agreements and the Bluetooth Trademark License

Bluetooth Qualification Administrator (BQA) - Responsible for administering the Bluetooth Qualification Program on behalf of the BQRB.

Bluetooth Qualification Body (BQB) - An individual person recognized by the BQRB to be responsible for checking declarations and documents against requirements, verifying the authenticity of product test reports, and listing products on the official Bluetooth Qualified Products List.

Bluetooth Qualification process - The process defined in the Bluetooth Qualification Program Reference Document (PRD) for Bluetooth Qualification.

Bluetooth Qualification program - The implementation of the Bluetooth Qualification process.

Bluetooth Qualification Review Board (BQRB) - Responsible for managing, reviewing and improving the Bluetooth Qualification program. The Bluetooth SIG promoter companies appoint delegates to the BQRB.

Bluetooth Qualification Test Facility (BQTF) - A test facility officially accredited by the BQRB to test Bluetooth products.

Bluetooth System Specification - The adopted version of this document.

Bluetooth Trademark - As defined in the Bluetooth Promoter, Early Adopters Agreement or Adopters Agreement for 1.0 Bluetooth Core Specification, or the Bluetooth Trademark License Agreement, whichever is applicable.

Compliant Portion - Only those specific portions of products (hardware, software or combinations thereof) that: (i) implement and are compliant with the Bluetooth System Specifications, (ii) are qualified pursuant to the Qualification Process, (iii) are within the bounds of the Scope and (iv) meet the requirements set forth in the Compliance Requirements.

Scope - As used in the definition of Compliant Portion, the protocols and data formats needed for Bluetooth interoperability, and the electrical signaling characteristics solely to the extent disclosed with particularity in the Bluetooth System Specifications where the sole purpose of such disclosure is to enable

products to interoperate, interconnect or communicate as defined within the Bluetooth System Specifications. For clarification, the Scope shall not include (i) any enabling technologies that may be necessary to make or use any prod-

uct or portion thereof that complies with the Bluetooth System Specifications, but are not themselves expressly set forth in the Bluetooth System Specifications (e.g., semiconductor manufacturing technology, compiler technology, object oriented technology, basic operating system technology, etc.); or (ii) the implementation of other published specifications developed elsewhere but referred to in the body of the Bluetooth System Specifications; or (iii) any portions of any product and any combinations thereof the purpose or function of which is not required for compliance with the Bluetooth System Specifications; or (iv) Application Programming Interfaces, applications, or user interfaces; including the technology used to generate, display or interact with a user.

3 LEGAL ASPECTS

The copyright in these specifications is owned by the Promoter Members of Bluetooth SIG, Inc. (“Bluetooth SIG”). Use of these specifications and any related intellectual property (collectively, the “Specification”), is governed by the Promoters Membership Agreement among the Promoter Members and Bluetooth SIG (the “Promoters Agreement”), certain membership agreements between Bluetooth SIG and its Adopter and Associate Members (the “Membership Agreements”) and the Bluetooth Specification Early Adopters Agreements (“1.2 Early Adopters Agreements”) among Early Adopter members of the unincorporated Bluetooth special interest group and the Promoter Members (the “Early Adopters Agreement”). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to Bluetooth SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of Bluetooth SIG or a party to an Early Adopters Agreement (each such person or party, a “Member”), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by aplicable law to Bluetooth SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the Bluetooth™ technology (“Bluetooth™ Products”) may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of Bluetooth™ Products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Bluetooth™ Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Bluetooth™ Products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in

the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate and to adopt a process for adding new Bluetooth™ profiles after the release of the Specification.

4 INTRODUCTION TO THE BLUETOOTH QUALIFICATION PROGRAM

The Bluetooth Qualification program is the framework that establishes the qualification rules and procedures. This section introduces the framework. The Bluetooth Qualification Program as well as Qualification requirements are defined in the Bluetooth Qualification Program Reference Document (PRD) available on the Bluetooth web site. Additionally, Guidelines for Bluetooth assigned names and numbers are defined in the PRD.

The Program defines the following entities:

- Bluetooth Qualification Review Board (BQRB)
- Bluetooth Qualification Administrator (BQA)
- Bluetooth Qualification Test Facility (BQTF)
- Bluetooth Qualification Body (BQB)

Figure 4.1introduces the process by which a Member qualifies a product.

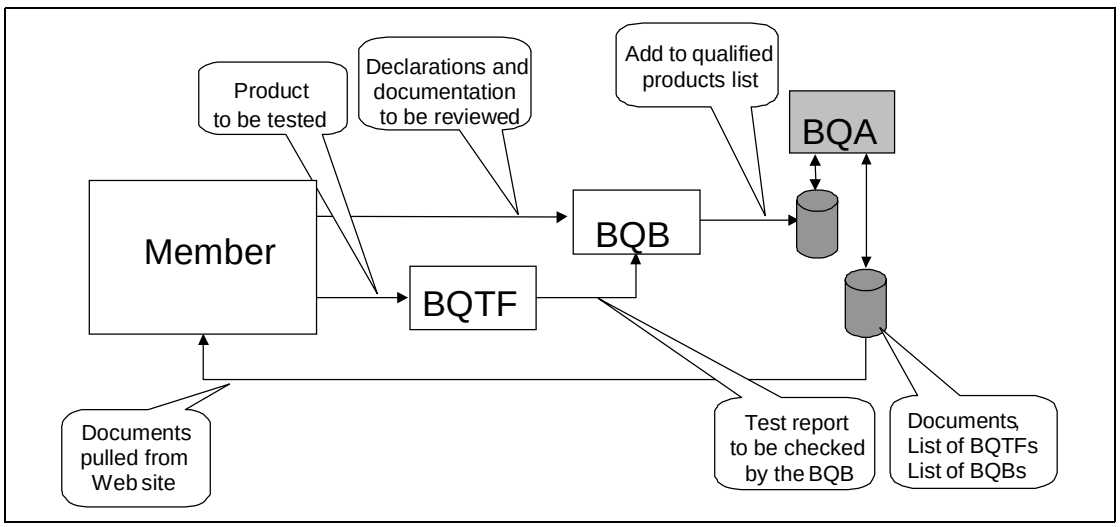


Figure 4.1:

An entity intending to produce and/or market a product, which includes Bluetooth functionality, must first become a Bluetooth Member by executing the applicable Bluetooth Agreements available via the Bluetooth Web site. Next, the Member must demonstrate that each particular product design complies with the Bluetooth System Specifications (including Compliance Requirements). This is accomplished when the particular product is listed as a.qualified product after successfully completing the Bluetooth Qualification process. For purposes of the rights granted under the applicable Bluetooth Agreement, the condition of Qualification is met from the date of listing. Under all applicable Bluetooth Agreements, every product is required to successfully complete the Bluetooth Qualification process. The rights granted under the applicable Blue-

tooth agreements do not apply to products which have not completed the Qualification process.

The Member selects a BQB from the list of recognized BQBs found on the Bluetooth web site. The BQB can assist the Member throughout the entire product qualification process. The Member prepares a compliance folder and works with the BQB to develop a product test plan. If the product test plan requires testing services of a BQTF, the Member can select a BQTF that has been accredited to perform the type of testing required. All necessary documentation to demonstrate compliance to the specification is provided to the BQB by the Member.

The BQB reviews the submittals and prepares a qualified product notice. In the event the BQB has a question regarding the conformance or interoperability of a product the BQB may, with the Member's permission, submit information via the BQA to the BQRB for a ruling. In the event the BQRB must be consulted, the Member will be requested to prepare a submission according to BQRB guidelines.

When all current requirements are satisfied, the BQB informs the Member that the product is ready for listing as a Bluetooth qualified product. With the consent of the Member, the BQB posts the product information to the official Qualified Product List.

A fee for product listing shall be paid by the BQB prior to product listing to finance the administration associated with the Qualification program.

Any Member can view the Bluetooth Qualified Product List. Confidential product information will not be published.

Part I:3

TEST CONTROL INTERFACE

This document describes the Bluetooth Test Control Interface (TCI). The TCI provides a uniform method of accessing the upper interface of the System Under Test (SUT) and/or the Implementation Under Test (IUT).

CONTENTS

1 Introduction 861
1.1 Terms Used 861
1.2 The needs for a unified test interface 861
1.3 Usage of the interface 862
2 General Description 863
2.1 Baseband and Link Management verification 863
2.2 HCI verification 865
2.3 L2CAP verification 866
3 Test Configurations 868
3.1 Bluetooth RF Link requirements 868
3.1.1 Required Interface(s) 868
3.2 Bluetooth protocol requirements 868
3.2.1 Required Interface(s) 869
3.3 Bluetooth profile requirements 869
3.3.1 Required Interface(s) 869
4 TCI-L2CAP Specification 870
4.1 Events 870
4.1.1 Connect Indication 870
4.1.2 Configuration Indication 871
4.1.3 Disconnect Indication 871
4.1.4 Violation Indication 871
4.1.5 Time Out Indication 871
4.2 Commands 872
4.2.1 Connection Establishment 873
4.2.2 Connect Response 873
4.2.3 Connection Release (Disconnect) 874
4.2.4 Configuration 874
4.2.5 Configure Response 874
4.2.6 Disable Connectionless Traffic 875
4.2.7 Enable Connectionless Traffic 875
4.2.8 Group Create 875
4.2.9 Group Close 876
4.2.10 Group Add Member 876
4.2.11 Group Remove Member 876
4.2.12 Group Membership 877
4.2.13 Ping 877
4.2.14 Get Info 877

4.2.15 Connect Release Response 878
4.3 Data Transfer 878
4.3.1 Write 878
4.3.2 Read 879
5 Abbreviations 881

1 INTRODUCTION

1.1 TERMS USED

IUT = Implementation Under Test: An implementation of one or more OSI protocols in an adjacent user/provider relationship, being that part of a real open system which is to be studied by testing.
This term will be used when describing the test concept for Bluetooth accessory products and Bluetooth components. The definition of Bluetooth accessory products and Bluetooth components can be found in Part I:2.

SUT = System Under Test: The real open system in which the IUT resides.
This term will be used when describing the test concept for Bluetooth products.

TCI = Test Control Interface: The interface and protocol used by the tester to send and receive commands and messages to and from the upper interface of the SUT/IUT.

1.2 THE NEEDS FOR A UNIFIED TEST INTERFACE

For all Bluetooth accessory products, Bluetooth components and Bluetooth products, protocol testing will be used to verify the implemented functionality in the lowest layers; i.e. conformance testing.

For this type of testing, an upper tester (UT) will be required to completely test the implementation.

In order to shield the tester from having to adopt to each and every implementation of IUTs or SUTs, the use of a standardized control interface is mandated. This concept puts some extra burden upon the manufacturer of the IUT/SUT. The manufacturer must:

- adopt the implementation-dependent interface to the TCI
- supply, with the IUT, the adapter needed (can be HW, SW or FW)

1.3 USAGE OF THE INTERFACE

The Bluetooth Test Control Interface, TCI, will be used when verifying the Bluetooth protocol requirements for a Bluetooth accessory product, Bluetooth component or a Bluetooth product. More specifically, the TCI will be used when verifying implemented functionality of the:

- Baseband layer, BB (the protocol-related part)
- Link Manager Protocol, LMP
- Logical Link Control and Adaptation Protocol, L2CAP

and, if support of the HCI is claimed by the manufacturer:

- Host Control Interface, HCI

2 GENERAL DESCRIPTION

The interface used between the tester and the SUT/IUT will be either of two types:

1. TCI-HCI
   This interface is semantically and syntactically identical to the HCI interface described in “Part H:1” on page 535.
2. TCI-L2CAP
   This interface is based on the HCI interface, and will be used during verification of the L2CAP layer of the SUT/IUT.

The proposed physical bearer is one of the transport layers specified for the HCI: USB, RS232 or UART, see “Part H:2” on page 781, “Part H:3” on page 797 or “Part H:4” on pag e819. However, alternatives do exist. More details will be given in the following sections.

2.1 BASEBAND AND LINK MANAGEMENT VERIFICATION

For the verification of the link control part of the Baseband layer and for the Link Manager layer, the TCI-HCI interface will be used as the interface between the test system and the upper interface of the SUT/IUT. The test system accesses the upper interface of the SUT/IUT by sending HCI commands and receiving HCI events from the SUT/IUT as described in the “Host Controller Interface Functional Specification” on page 535. The supported functionality on the TCI-HCI interface depends on the implemented functionality of the BB and LM layers.

The transport bearer used between the tester and the SUT/IUT can be of either of two types:

1. A physical bearer of one of the types USB, RS232 or UART, as defined in Part H:2, Part H:3 or Part H:4. It is recommended to use one of these three physical bearers as transport bearer between the SUT/IUT and the test system.
2. A ‘software’ transfer bearer; i.e. there is no physical connection between the tester and the SUT/IUT. In this case, the manufacturer of the SUT/IUT must supply, when sending in the device for testing, a test software that can be operated by a test operator. The operator will receive instructions from the tester and will execute them on the SUT/IUT. The software must support the same functionality as if using the TCI-HCI with a physical bearer. Use of the ‘software’ interface must be agreed upon between the manufacturer of the SUT/IUT and the test facility that will perform the verification. The test facilities can themselves specify requirements placed on such an interface.

A schematic example is shown in Figure 2.1 of a possible test configuration for BB and LM verification of Bluetooth products which do not support HCI, and which use a physical transport bearer for the TCI-HCI interface. In this figure,

the TC (Test Control) Software represents what the manufacturer has to supply with the SUT/IUT when sending it in for verification. The functionality of the TC software is to adapt the implementation-dependent interface to the TCI-HCI interface.

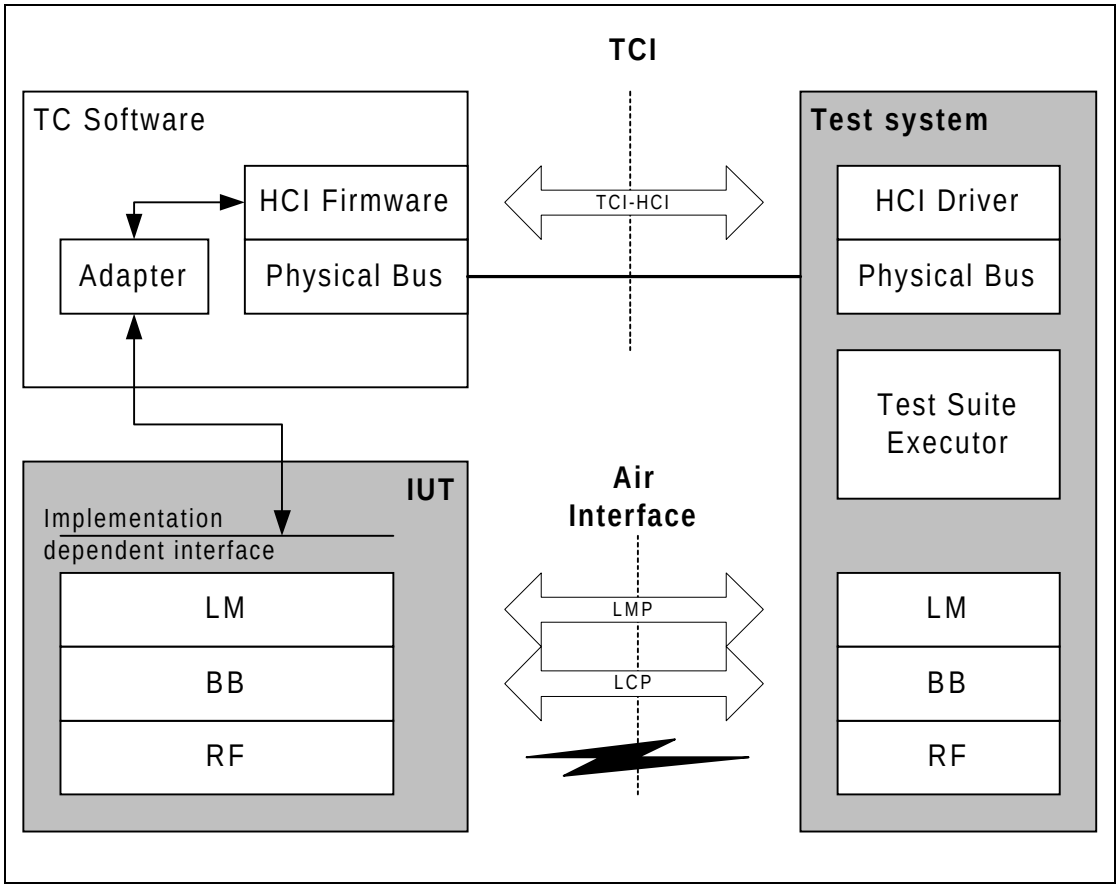


Figure 2.1: Baseband and LM verification without HCI – physical transport bearer

Figure 2.2 shows a schematic example of the test configuration for the same Bluetooth product using a ‘software’ transfer bearer for the TCI-HCI interface. Here, the role of the TC Software is to represent the application that can be controlled by the test operator.

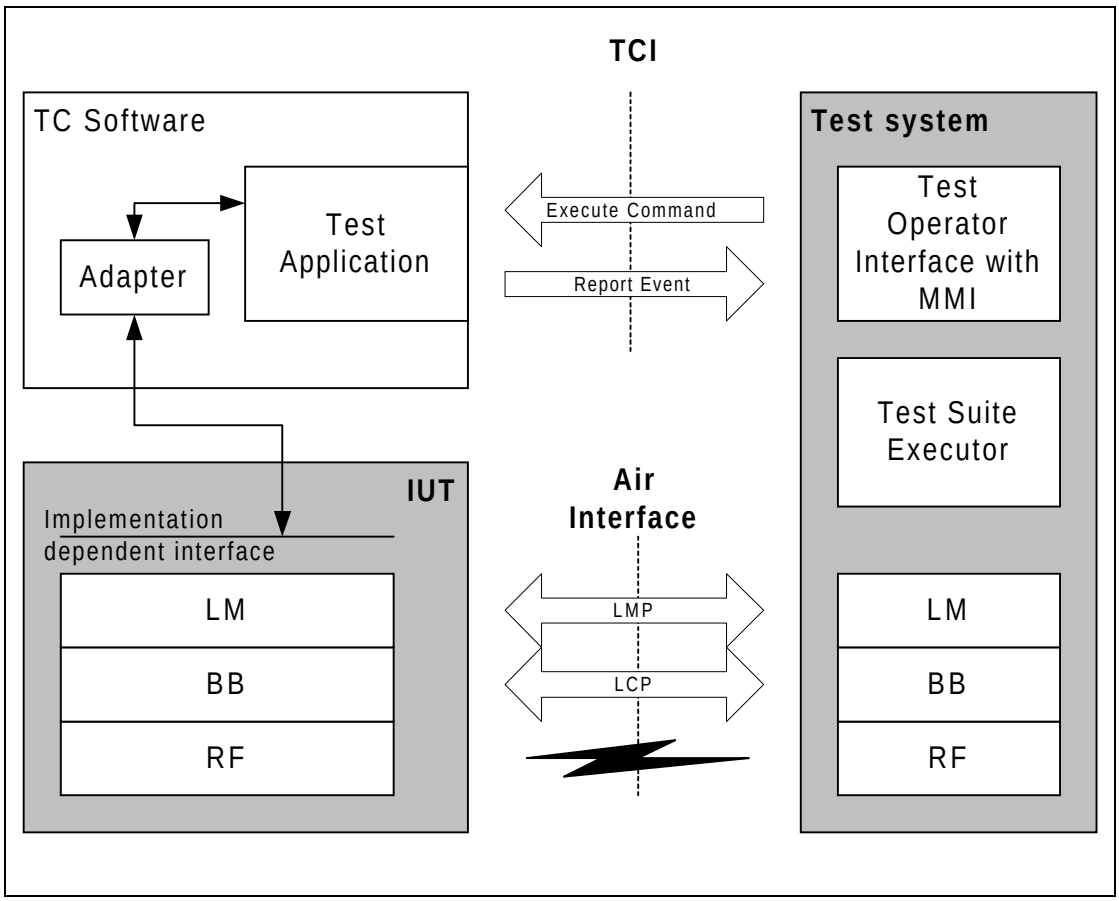


Figure 2.2: Baseband and LM verification without HCI – software transport bearer

2.2 HCI VERIFICATION

The TCI-HCI interface may also be used for HCI signalling verification. The HCI signalling will only be verified if support of the HCI functionality is claimed by the manufacturer.

The transport bearer between the tester and the SUT/IUT shall be one of the types USB, RS232, or UART, as defined in Part H:2, Part H:3 or Part H:4.

A schematic example is shown in Figure 2.3 of one possible test configuration for HCI verification of Bluetooth products, using a physical transport bearer for the TCI-HCI interface. As can be seen from the figure, no extra test control software is needed. Instead, the implemented HCI will be used as the interface to the tester.

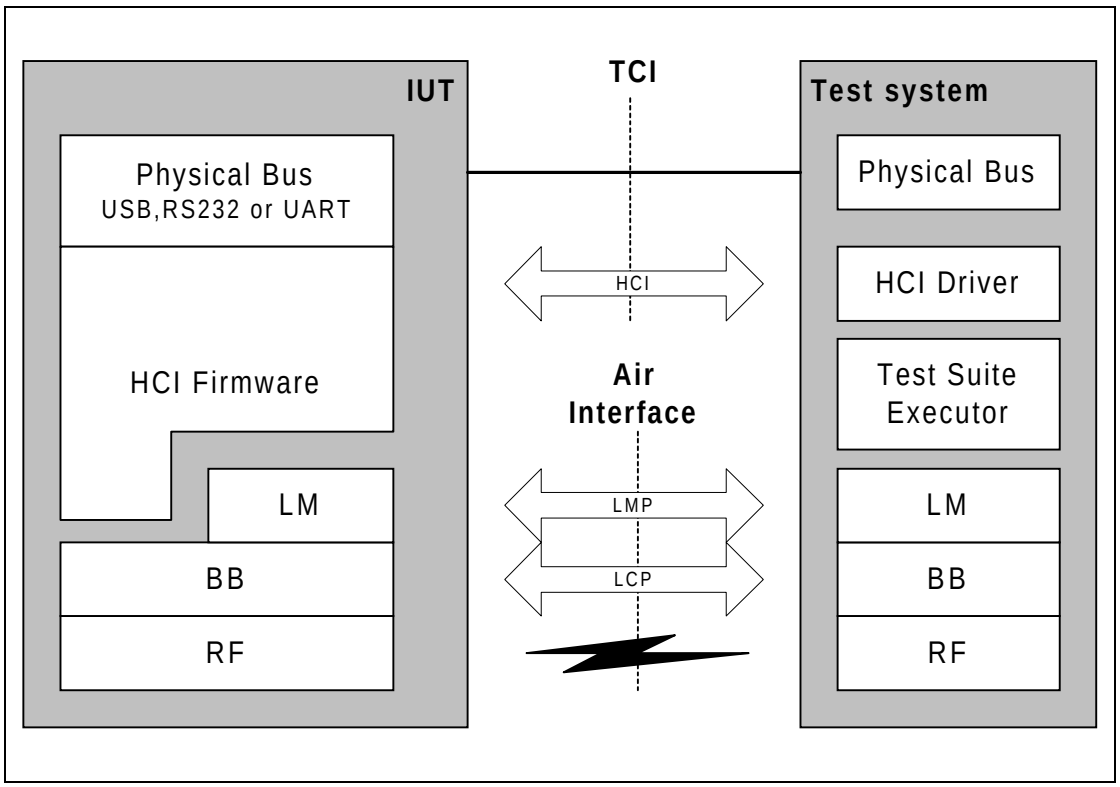


Figure 2.3: HCI verification

2.3 L2CAP VERIFICATION

The TCI-L2CAP interface is based on the HCI and will be used during verification of the L2CAP layer of the SUT/IUT. It uses the general event and command syntax as specified in Part H:1, and the mapping to transport layers is also identical to the ones defined in Part H:2, Part H:3 or Part H:4. Commands and events are defined according to the specified L2CAP service interface. See Part D / Section 7 on page 303.

The defined service primitives in the Logical Link and Control Layer specification, Part D / Section 7 on page 303, will be used as reference. However, the primitives for L2CAP events and commands must be converted into messages of the same format as used for the HCI events and commands. The mapping of the L2CAP events and commands to HCI format is described in Section 4 of this document.

A schematic example is shown in Figure 2.4. of how the test configuration can look for L2CAP verification of Bluetooth products, using a physical transport bearer for the TCI-L2CAP interface. In this figure, the TC (Test Control) Software represents what the manufacturer has to supply with the SUT/IUT when sending it in for verification. The functionality of the TC software is to adapt the implementation-dependent interface to the TCI-L2CAP interface.

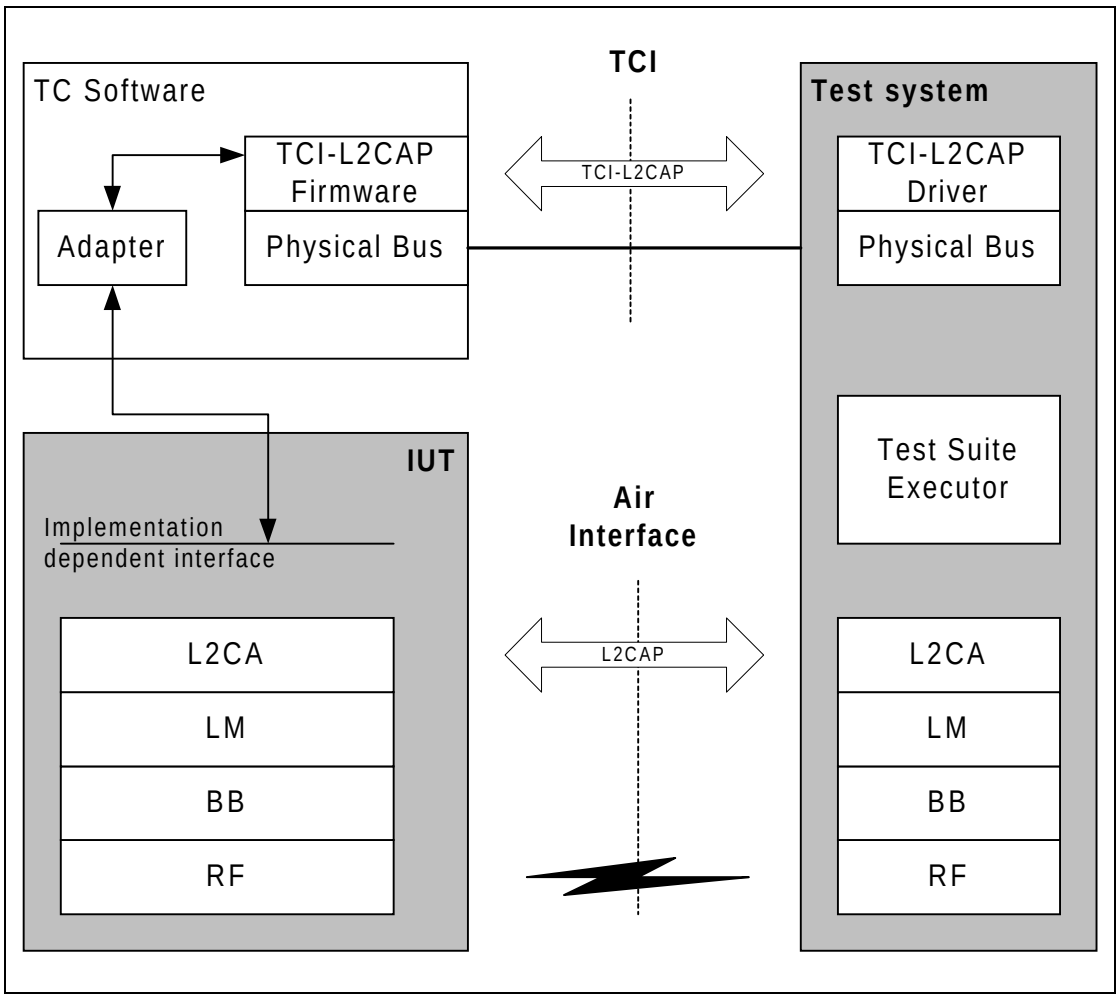


Figure 2.4: L2CAP verification

3 TEST CONFIGURATIONS

This section describes the test configurations that will be used when verifying the different Bluetooth requirements.

3.1 BLUETOOTH RF LINK REQUIREMENTS

For the verification of the Bluetooth RF Link requirements, the defined test mode will be used, see “Part I:1” on page827 .

The Test Specification for the Bluetooth radio link requirements will be based on the Bluetooth specification Parts A and B, and will contain the relevant test instructions that should be carried out on the SUT/IUT.

3.1.1 Required Interface(s)

For this type of verification, only the air interface is required. See Figure 3.1. As stated in Part I:1 / Section 1.2 on page 832, for security reasons, the test mode must be locally enabled. The implementation of this local enabling is not subject to standardization.

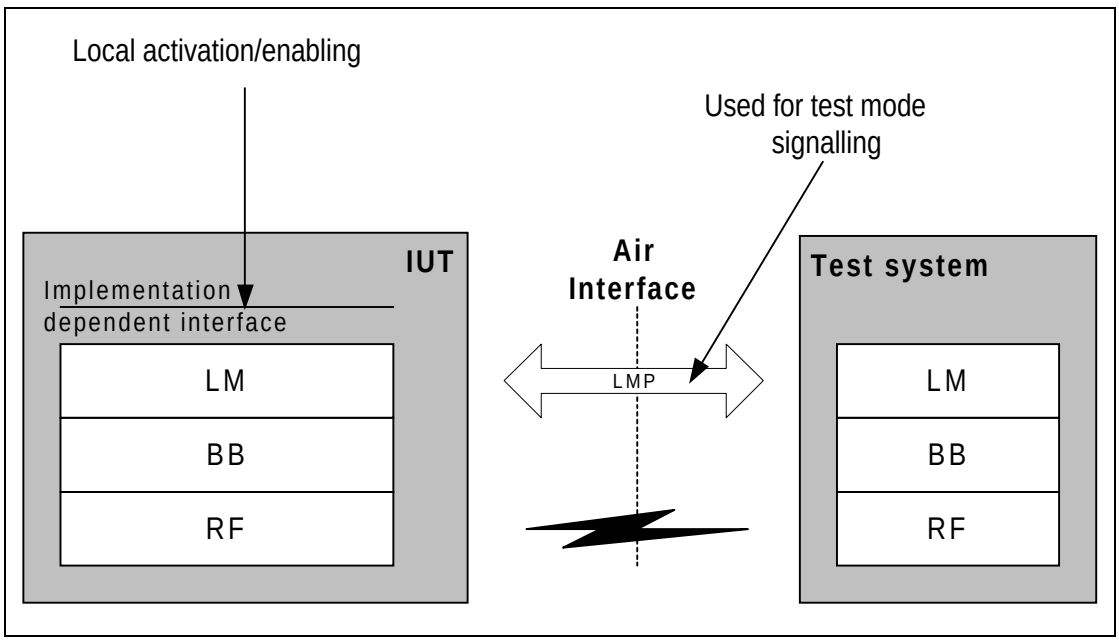


Figure 3.1: Test Configuration for RF link requirement verification

3.2 BLUETOOTH PROTOCOL REQUIREMENTS

Dependent on which Bluetooth layers BB, LM, HCI or L2CAP are implemented in the product sent in for verification, the amount of testing needed to verify the Bluetooth protocol requirements will differ. Also, the TCI used during the verification may be different.

The Test Specification for the Bluetooth protocol requirements will be based on the Bluetooth specification Part A to Part D and Part H, if applicable, and will contain the relevant test instructions that should be carried out on the SUT/IUT.

3.2.1 Required Interface(s)

For this type of verification, both the air interface of the SUT/IUT and the test control interface are required. The latter will be one of the types described in section 2.

3.3 BLUETOOTH PROFILE REQUIREMENTS

For each profile the Bluetooth product claims to conform to, profile testing will be performed to verify the Bluetooth profile requirements in order to ensure interoperability between products; i.e. interoperability testing.

The Test Specification for the Bluetooth profile requirements will be based on the Bluetooth specification Part K Volume 2, and will contain the relevant test instructions that should be carried out on the SUT.

3.3.1 Required Interface(s)

For this type of verification, both the air interface of the SUT and the supported MMI, as described in the profile, will be used during verification, see Figure 3.2.

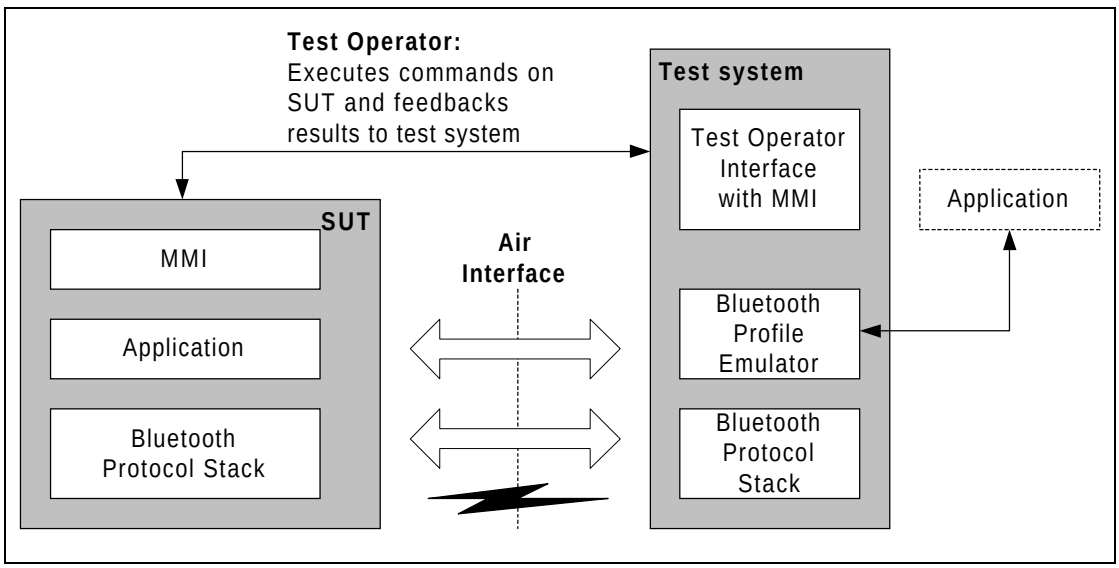


Figure 3.2: Test Configuration for Profile requirement verification

4 TCI-L2CAP SPECIFICATION

Note: This specification maps the L2CAP service interface to an appropriate TCI.

4.1 EVENTS

In the L2CAP service interface, indications are mapped to callback functions. The corresponding response parameters are submitted in the return parameter of these functions. For the TCI, the indications are mapped to events and the responses to commands.

A single event code is reserved for testing purposes: 0xFE. To distinguish the L2CAP events, a parameter 'Event_ID' is submitted as first parameter. This parameter is a single octet, resulting in 256 possible events. The assignment is given in Table 4.1.

Event_ID	L2CAP event
0x00	Reserved
0x01	L2CA_ConnectInd
0x02	L2CA_ConfigInd
0x03	L2CA_DisconnectInd
0x04	L2CA_QoSViolationInd
0x05	LC2A_TimeOutInd
0x06 – 0xFF	Reserved

Table 4.1: Assignment of event IDs

The events in this test interface follow the HCI syntax as defined in Part H:1 / Section 4.4.2 on page 554.

4.1.1 Connect Indication

Event	Event Code	Event Parameters
L2CA_ConnectInd	0xFE	Event_ID, BD_ADDR, CID, PSM, Identifier

For more details and the event parameter, see Part D / Section 7.1 on page 303. The parameter "Identifier" is ignored, i.e. the bytes must be present but the value can be arbitrary.

4.1.2 Configuration Indication

Event	Event Code	Event Parameters
L2CA_ConfigInd	0xFE	Event_ID, CID, OutMTU, InFlow, FlushTO

For more details and the event parameter, see Part D / Section 7.1 on page 303.

4.1.3 Disconnect Indication

Event	Event Code	Event Parameters
L2CA_DisconnectInd	0xFE	Event_ID, CID

For more details and the event parameter, see Part D / Section 7.1 on page 303.

4.1.4 Violation Indication

Event	Event Code	Event Parameters
L2CA_QoSViolationInd	0xFE	Event_ID, BD_ADDR

For more details and the event parameter, see Part D / Section 7.1 on page 303.

4.1.5 Time Out Indication

Event	Event Code	Event Parameters
L2CA_TimeOutInd	0xFE	Event_ID, CID, Timer

The local CID for which the timer has expired and the type of timer is indicated. The time value (size 2 octets) is 0x0001 if RTX expired and 0x0002 if ERTX expired.

For more details and the event parameter, see Part D / Section 7.1 on page 303.

4.2 COMMANDS

The commands in this test interface follow the HCI syntax as defined in Part H:1 / Section 4.4.1 on page 551. The return parameters are sent back using a Command Complete event, see Part H:1 / Section 5.2.14 on page 744 and Part H:1 / Section 4.4.1 on page 551.

To distinguish the commands used for L2CAP testing from HCI commands, a single subgroup is reserved for the L2CAP test interface. Figure 4.1 shows how to code and decode the OpCode field in the HCI command packet used for testing.

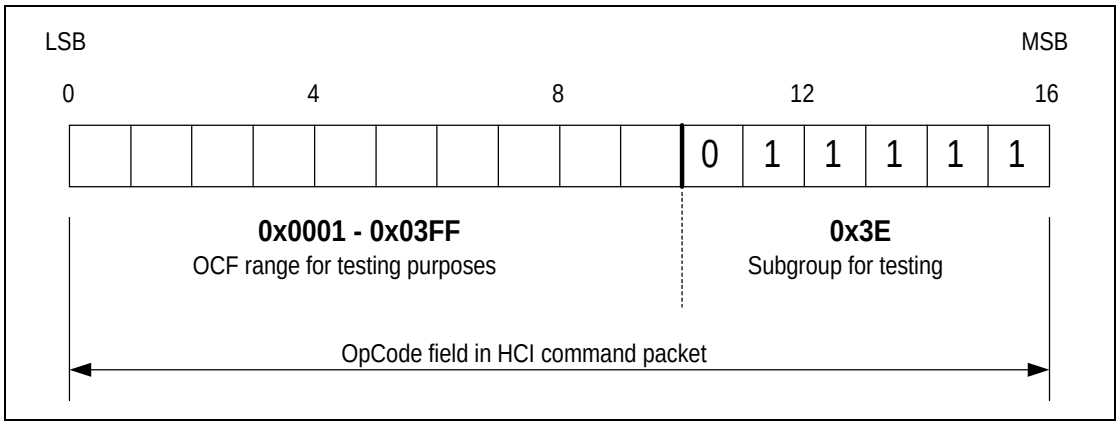


Figure 4.1: HCI Opcode field values used for testing

The assignment of the OpCode Command Field, OCF, for the L2CAP commands is summarized in Table 4.2. It is also detailed following the table, in a format similar to the HCI specification, see “Part H:1” on pag e535.

OCF	L2CAP command
0x0000	Reserved
0x0001	L2CA_ConnectReq
0x0002	L2CA_DisconnectReq
0x0003	L2CA_ConfigReq
0x0004	L2CA_DisableCLT
0x0005	L2CA_EnableCLT
0x0006	L2CA_GroupCreate
0x0007	L2CA_GroupClose
0x0008	L2CA_GroupAddMember
0x0009	L2CA_GroupRemoveMember

Table 4.2: Assignment of Opcode Command Field values

OCF	L2CAP command
0x000A	L2CA_GroupMemebership
0x000B	L2CA_WriteData
0x000C	L2CA_ReadData
0x000D	L2CA_Ping
0x000E	L2CA_GetInfo
0x000F	Reserved
0x0010	Reserved
0x0011	L2CA_ConnectRsp
0x0012	L2CA_DisconnectRsp
0x0013	L2CA_ConfigRsp
0x0014 – 0x03FF	Reserved

Table 4.2: Assignment of Opcode Command Field values

4.2.1 Connection Establishment

Command	OCF	Command Parameters	Return Parameters
L2CA_ConnectReq	0x0001	PSM, BD_ADDR	LCID, Result, Status

Description:

Requests the creation of a channel representing a logical connection to a physical address (for more details and input/output parameter definition see Part D / Section 7.2 on page 304).

4.2.2 Connect Response

Command	OCF	Command Parameters	Return Parameters
L2CA_ConnectRsp	0x0011	BD_ADDR, Identifier, LCID, Response, Status	Result

Description:

Issues a response to a connection request event indication. The parameter “Identifier” is ignored, i.e. the bytes must be present but the value can be arbitrary.For more details and input/output parameters definition see Part D / Section 7.3 on page 306

4.2.3 Connection Release (Disconnect)

Command	OCF	Command Parameters	Return Parameters
L2CA_DisconnectReq	0x0002	CID	Result

Description:

Requests the disconnection of the channel. Input parameter is the *CID* representing the local channel endpoint (for more details and input/output parameter definition see Part D / Section 7.6 on page 310).

4.2.4 Configuration

Command	OCF	Command Parameters	Return Parameters
L2CA_ConfigReq	0x0003	CID, InMTU, OutFlow, FlushTO, LinkTO	Result, InMTU, OutFlow, FlushTO

Description:

Requests the initial or new configuration of a channel to a new set of channel parameters. The parameter “LinkTO” is ignored, i.e. the bytes must be present but the value can be arbitrary.For more details and input/output parameter definition see Part D / Section 7.4 on page 307.

4.2.5 Configure Response

Command	OCF	Command Parameters	Return Parameters
L2CA_ConfigRsp	0x0013	CID, OutMTU, InFlow	Result

Description:

Issues a response to a configuration request event indication (for more details and input/output parameter definition see Part D / Section 7.5 on page 309).

4.2.6 Disable Connectionless Traffic

Command	OCF	Command Parameters	Return Parameters
L2CA_DisableCLT	0x0004	N, List of PSMs	Result

Description:

For details and input/output parameter definition see Part D / Section 7.16 on page 319.

4.2.7 Enable Connectionless Traffic

Command	OCF	Command Parameters	Return Parameters
L2CA_EnableCLT	0x0005	N, List of PSMs	Result

Description:

For details and input/output parameter definition see Part D / Section 7.17 on page 320.

4.2.8 Group Create

Command	OCF	Command Parameters	Return Parameters
L2CA_GroupCreate	0x0006	PSM	CID

Description:

Request the creation of a channel identifier to represent a logical connection to multiple devices. On creation, the group is empty (for more details and input/output parameter definition see Part D / Section 7.9 on page 313).

4.2.9 Group Close

Command	OCF	Command Parameters	Return Parameters
L2CA_GroupClose	0x0007	CID	Result

Description:

This command closes down a Group (for more details and input/output parameter definition see Part D / Section 7.10 on page 313).

4.2.10 Group Add Member

Command	OCF	Command Parameters	Return Parameters
L2CA_GroupAddMember	0x0008	CID, BD_ADDR	Result

Description:

This command adds a member to the group (for more details and input/output parameter definition see Part D / Section 7.11 on page 314).

4.2.11 Group Remove Member

Command	OCF	Command Parameters	Return Parameters
L2CA_GroupRemoveMember	0x0009	CID, BD_ADDR	Result

Description:

Remove a member from the group (for more details and input/output parameter definition see Part D / Section 7.12 on page 315).

4.2.12 Group Membership

Command	OCF	Command Parameters	Return Parameters
L2CA_GroupMembership	0x000A	CID	Result, N, BD_ADDR_Lst

Description:

Get report of the members of the group (for more details and input/output parameter definition see Part D / Section 7.13 on page 316).

4.2.13 Ping

Command	OCF	Command Parameters	Return Parameters
L2CA_Ping	0x000D	BD_ADDR, Length, ECHO_DATA	Result, Size, ECHO_DATA

Description:

For more details and input/output parameter definition see Part D / Section 7.14 on page 317.

NOTE: Before the ECHO_DATA is sent over the TCI, the field should be truncated to a maximum size of 44bytes, i.e. only the first 44 bytes of the ECHO_DATA should be sent.

The reason for this has to do with the restriction on the minimum MTU size for the commands on the signalling channel (44 bytes of ECHO_DATA + 4 bytes of command header)

4.2.14 Get Info

Command	OCF	Command Parameters	Return Parameters
L2CA_GetInfo	0x000E	BD_ADDR, InfoType	Result, InfoData

Description:

For more details and input/output parameter definition see Part D / Section 7.15 on page 318.

4.2.15 Connect Release Response

Command	OCF	Command Parameters	Return Parameters
L2CA_Disconnect_Rsp	0x0012		

Description:

No parameters are sent because the disconnection always has to be accepted.

Note: No parameters are specified. It only indicates to the local device that the disconnect request had been executed.

4.3 DATA TRANSFER

Data transfer is modelled with read and write functions. Handling is like an L2CAP command.

To be able to send the amount of data that is needed to verify how the L2CAP implementation handles large chunks of data (i.e. segmentation and reassembly), and since it is not possible to use HCI Command packets as well as HCI Event packets to send the data, the use of HCI ACL Data packets will be used. The procedure/signalling used on the TCI-L2CAP interface to transfer data packets will be described with MSCs.

4.3.1 Write

Command	OCF	Command Parameters	Return Parameters
L2CA_WriteData	0x000B	CID, Length, OutBuffer	Size, Result

The parameter “Size” is ignored, i.e. the bytes must be present but the value can be arbitrary.

Description:

Parameters are the CID, the length of the data and the data itself. The data will be sent in a HCI ACL data packets as described in Figure 4.2. For more details and input/output parameter definition see Part D / Section 7.7 on page 311.

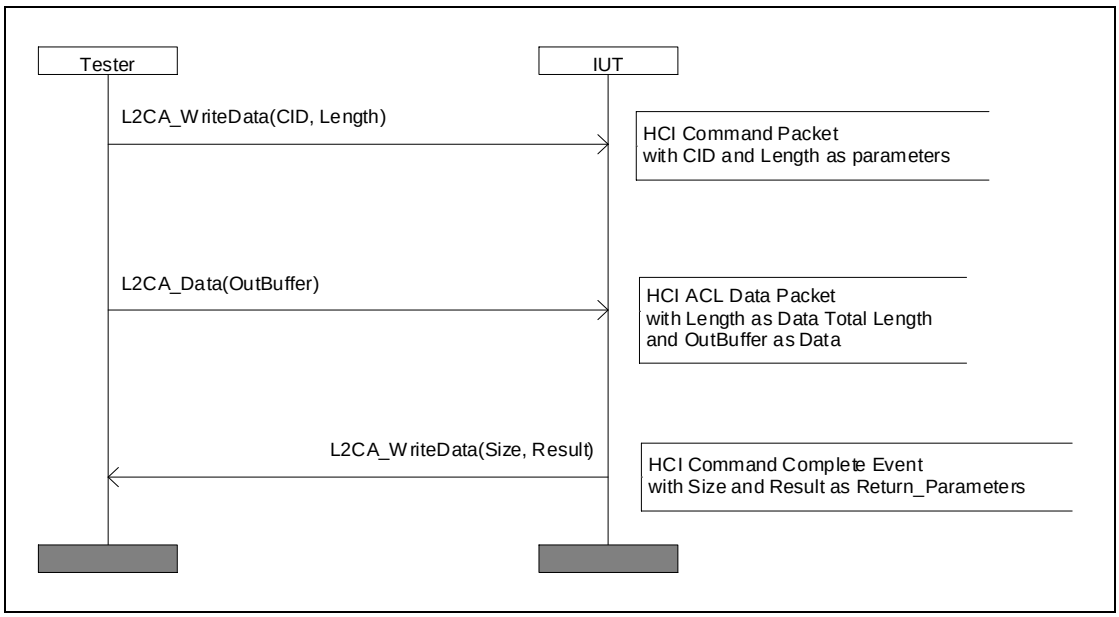


Figure 4.2: MSC showing how to write data to L2CAP

The L2CA_Data primitive is used as an abstract name for the data transmitted between the Tester and the IUT. The Tester will use Connection Handle 0x0001 for the data and will set the Flags-field to 0x02. The DataTotal Length field will contain the length of the OutBuffer.

After the IUT has received the data, it shall send back an HCI Command Complete Event (named L2CAP_WriteData in the figure) with the N parameter set to 0x01, the OpCode parameter set to the corresponding OCF and subgroup (that is OCF = 0x000B and the subgroup = 0x3E). The Size and Result are sent in the Return_Parameters field of the HCI ACL Data packet. The parameter size is ignored, i.e. the bytes must be present but the value can be arbitrary.

4.3.2 Read

Command	OCF	Command Parameters	Return Parameters
L2CA_ReadData	0x000C	CID, Length, InBuffer	Result

Description:

Input parameter is the CID, length and the InBuffer. Output parameters are the result. The data will be sent in HCI ACL data packets as described in Figure 4.3. For more details and input/output parameter definition see Part D / Section 7.8 on page 312).

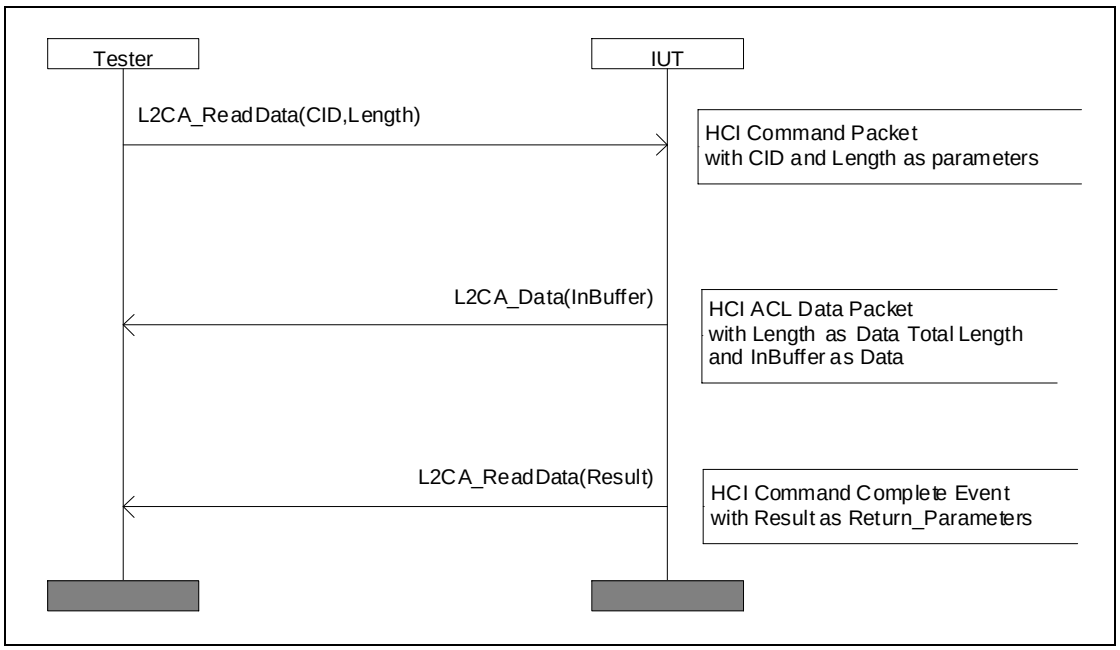


Figure 4.3: MSC showing how to read data from L2CAP

The L2CA_Data primitive is used as an abstract name for the data transmitted between the Tester and the IUT. The IUT shall use Connection Handle 0x0001 for the data and shall set the Flags-field to 0x02. The DataTotal Length field shall contain the length of the InBuffer.

After the IUT has sent the data, it shall send back an HCI Command Complete Event (named L2CA_ReadData in the figure) with the N parameter set to 0x01, the OpCode parameter set to the corresponding OCF and subgroup (that is OCF = 0x000C and the subgroup = 0x3E). The Size and Result are sent in the Return_Parameters field of the HCI ACL Data packet.

5 ABBREVIATIONS

BB	BaseBand (see LC)
FW	Firmware
HCI	Host Controller Interface
HW	Hardware
IUT	Implementation Under Test
L2CA	Logical Link Control And Adaption part of the Bluetooth protocol stack
L2CAP	Logical Link Control And Adaption Protocol
LC	Link Controller (or baseband) part of the Bluetooth protocol stack
LCP	Link Control Protocol
LM	Link Manager part of the Bluetooth Protocol Stack
LMP	Link Manager Protocol
MMI	Man-Machine Interface
OCF	Opcode Command Field
RF	Radio part of the Bluetooth protocol stack
SUT	System Under Test
SW	Software
TC	Test Control layer for the test interface
TCI	Test Control Interface
UART	Universal Asynchronous receiver Transmitter
USB	Universal Serial Bus
UT	Upper Tester

Appendix I

REVISION HISTORY

Revision History

Part A / Radio Specification

Rev	Date	Comments
0.8	Jan 21st 1999	• System ambient temperature range • Power control step size • Transmit Spectrum mask • Frequency drift in a packet • New paragraph "Receiver susceptibility to frequency drift" • Adjacent interference levels • Measurement frequency for intermodulation test • Maximum useable level defined • New paragraph "Reference Interference-Signal Definition"
0.9	April 30th 1999	• Eye-diagram added • Out-of band blocking included • RSSI included
1.0 draft	July 5th 1999	• Appendix A and B added, (extreme conditions definition and test conditions) • Tolerances of the Eye-diagram added
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part B / Baseband Specification

Rev	Date	Comments
0.7	Oct 19th 1998	• Minor changes in chapter 1-9. • ”Link Monitoring” chapter removed (chapter 12 in v0.6). • Most parts of chapter 10 - 14 re-written.
0.8	Jan 21st 1999	• Some editorial changes in chapter 1-13. • Chapter 14 revised and partly re-written.
1.0 draft	July 5th 1999	• Please see revision bars in document.
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part C / Link Manager Protocol

Rev	Date	Comments
0.7	Oct 19th 1998	• EEE address ⇒ BD_ADDR (or BD address) • M_ADDR ⇒ AM_ADDR • response nbr ⇒ PM_ADDR • Added section 3.1.4 • Added section 3.4 “Change the current link key” • Changed section 3.5 “Encryption”. Added negotiation for encryption mode and encryption key size. • 3.14 “Sniff mode”. Removed author’s note in italic. It is now decided how to determine the first sniff slot. • 3.15 “Park mode”. This section is totally changed. • 3.18 “Quality of Service”. Added a parameter N_{BC} in one of the LMP messages. • 3.19 “SCO links”. Removed author’s note in italic. It is now decided how to determine the first SCO slot. • Added section 4 “Connection establishment” • 5.1 “Description of parameters” The length, type and unit of many parameters changed. Especially, all time-parameters are now measured in slots. • Added section 6 “Test modes”
0.8	Jan 21st 1999	• general: Changed "PDU nbr" to OpCode throughout the document. Two errors found in 0.80 were corrected. Table5.1: It says that the length of LMP_SCO_link_req is 7. This should be 8. Table 5.2: It says that the length of hold_time is 1. This should be 2. • 2. Added transactionID in bit0 of the byte in the payload where we have the OpCode. • 3.1 Removed LMP_accepted/not_accepted after receiving LMP_sres (3.1). • 3.1.2 Removed the last sentence "It the claimant is Ö initiate pairing, see 3.2.1" • 3.2 Major change since the initiator of the pairing procedure does not have to be the master. • 3.2.3 Minor clarification. • 3.3 Clearified that if the unit key is changed, the units must go through the initialisation procedure in order to change the link key. • 3.5.3 Modified the calculation of Kc according to changes in chapter 14 of the Baseband Specification. • 3.10 Switch can now be done anytime during the connection and it can be initiated by both master and slave. • 3.15 Clearified that the broadcast scan window is only valid for the current beacon. • 3.19.5 Added description that the PDU includes a reason parameter with information about why the SCO link is removed. • 3.21 This section was removed. Instead error handling is described in a separate chapter (CH. 7) • 4. Added LMP_host_connection_req and removed the LMP_accepted or LMP_not_accepted that the slave sends after the paging procedure. • 5.1 Added reason "invalid parameters" • 6. Added sequences. The slave can return LMP_not_accepted if not allowed to enter test or if not in test mode. Changed OpCodes for the two test mode PDU

0.9	April 30th 1999	• Name request procedure: Coding of the characters was changed from ASCII to UTF-8. The length of the name parameter was increased from 16 bytes to 248 bytes. • The detach reason and the reason parameter were merged into one parameter (reason). The coding of this parameter was changed and is now the same as in HCI. • Sniff procedure: Sniff interval and sniff offset parameters were changed from one byte to two bytes. Two parameters, sniff offset and sniff timeout, were also added. • The PDU LMP_slot_offset was added. • The PDU LMP_page_mode_req was added. • The PDU LMP_page_scan_mode_req was added. • Caption text was added to all figures, tables and sequences where such text was missing. • A maximum reply time was defined for response messages in LMP procedures. • The pairing procedure was modified to allow a claimant with fixed PIN to request to become verifier. • Changed some parameters in LMP_test_control to match the latest revision of "Bluetooth test modes". • The PDU LMP_supervision_timeout was added. • The parameter user data rate was removed from the PDU LMP_SCO_link request. • The length of the parameters CompId and SubVersNr included in the PDU LMP_version_req/res was increased to two bytes. • Some editorial changes and clarifications were made here and there.
1.0 draft	July 5th 1999	• Added figure 3 explaining sequence diagram conventions (Section 3). • Modified procedure to negotiate encryption key size to match changes in the Baseband specification (Section 3.6.2). • Added further description about the parameters in the version request (Section 3.10). • Clearified name request procedure and changed parameter name from "name" to "name fragment" (Section 3.13). • Added description that test mode can also be ended by sending LMP_test_control (Section 6.1). • Some editorial comments. • Decoupled RSSI and power control in the features parameter. Changed parameter "TBD" to "for future use". • Rewrote Section 3.20.
1.0a	July 26th 1999	• Correction ofTable 5.3: Byte 1/Bit 1 “and power control” deleted Byte 2/Bit 2 “power control” added
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part D
Logical Link Control and Adaptation Protocol Specification

Rev	Date	Comments
0.8	Jan 21st 1999	• Changes include the addition of an operational overview section, increasing CID lengths to 16 bits and removing the source CID from normal data flow. Moved all connection, termination, and configuration commands to a separate "CID". Changed the name of the "Connection ID" to the "Channel ID". • Added the state machine • Re-defined the timers to more clearly indicate their responsibilities. New Flags field in Configuration Request packet defined. • Several minor editorial corrections.
0.9	April 30th 1999	•
1.0 draft	July 5th 1999	• Replaced all references to "termination" with "disconnection". KTX timer removed - link loss results in channel loss. OSI naming terms applied to state machine. State machine table revised to remove superfluous states. Added some message sequence charts for clarification. Service interface defined in more detail and no longer specified as a guideline - this interface still needs work to complete. • Service interface revised corrections to state machine added, and security key management removed from service interface. • Editorial cleanup of various sections with the majority of the edits being in the state machine and service interface. • Added L2CA_Response service primitives. Clean up of state machined, closed outstanding issues dealing with MTU, • Implements major editorial changes to the configuration process (section 6.3), replacing SCID and DCID with RCID and LCID, removal of flush signalling PDU, and editorial comments from reviewers.
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part E / Service Discovery Protocol (SDP)

Rev	Date	Comments
1.0 draft	July 5th 1999	• Removed the Icon10 attribute and the Icon data element type. Re-assigned the IconURL attribute ID to be 0x000C. Updated Example 3. Removed "notes to reviewers". Removed the list of TBD items, since it is empty. • A few corrections to example 3. Modified the BluetoothProfileList attribute to become the BluetoothProfileDescriptorList attribute, which contains a version number for each profile as well as the profile's UUID • Added a description of the protocol version number included in the BluetoothProfileDescriptorList attribute. No change to the 'phonebook' and 'calendar' data store indicators as the values in example 3 already match those of the synchronization profile. Replaced the browse hierarchy diagram in 2.8.1 because the previous version triggered a bug in MS-Word.
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part F:1 / RFCOMM with TS 0710

Rev	Date	Comments
0.8	Jan 21st 1999	• Table 2.1 corrected. • Revised section 2.3. • Revised chapter 3. • Revised section 5.2, 5.3. • Clarifications added to Figure 5.1. • Changed title and contents of section 5.4 (old section 5.4 not needed any more). • Text removed from section 5.5 and 5.6. • Added text on Service Discovery in section 7.3.
0.9	April 30th 1999	• Lots of editorial changes and clarifications. • Added statement on baud rate settings vs. RFCOMM throughput in chapter 2 • Removed section 7.2 on flow control (information duplicated in TS 07.10). • Added DLC parameter negotiation command support (section 4.3, 5.7) • Added clarifications on session closure handling in section 5.2. • Major update regarding SDP; section 7.2. • Added section 7.3 on lower layer dependencies.
1.0 draft	July 5th 1999	• Editorial changes and clarifications in chapters 5 and 7. • Removed sections implying possibility to have more than one RFCOMM entity in a device.
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part F:2 / IrDA Interoperability

Rev	Date	Comments
0.9	April 30th 1999	• Many linguistical changes made, Bluetooth OBEX related specifications chapter added, Separate Application profile chapters gathered into one chapter, OBEX operation requirements moved into the profile specifications, Refers to COMM-ports removed. • Service Records moved into the profile specifications, and added clarification for the use of connection-oriented OBEX
1.0 draft	July 5th 1999	• Changed reference to new version of IrOBEX specification, corrected wrong TCP port number, reference list updated • Updated Chapter 3.2
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.

Part F:3 / Telephony Control Specification (TCS)

Rev	Date	Comments
1.0 draft	July 5th 1999	• Version for 1.0 Release, only editorial changes since 0.9
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars. Figure A and Figure B in Appendix are replaced.

Part F:4
Interoperability Requirements for Bluetooth as a WAP Bearer

Rev	Date	Comments
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part H:1
Bluetooth Host Controller Interface Functional Specification

Rev	Date	Comments
0.8	Jan 21st 1999	• Many editorial corrections
0.9	April 30th 1999	•
1.0 draft	July 5th 1999	• Flow control for data changed. • Format of HCI Data Packet header changed. • Command Pending Event replaced by Command Status Event and functionality regarding which event should be returned when a command involving LMP actions can not start to execute due to an error changed. • Some commands and events have been renamed. • Changed parameters, descriptions and functionality for many commands and events. • Many new commands have been added. • HCI_Store_Clock_Offset command has been removed. This information is now provided at connection set up or name request as a parameter together with other new baseband related parameters. • Some new events have been added. • Descriptions of error codes added.
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part H:2 / HCI USB Transport Layer

Rev	Date	Comments
0.8	Jan 21st 1999	• Added info about 64 byte isochronous endpoints. • Added a section detailing one of the mail messages - that discussed how SCO traffic would travel across the interface
0.9	April 30th 1999	• Updated revision # & HCI Header sizes •
1.0 draft	July 5th 1999	• Updated endpoint information, interface information relating to isoch, and Device Firmware Upgrade Requirements • Tidied up table describing interface/endpoint/alternate setting information
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part H:3 / HCI RS232 Transport Layer

Rev	Date	Comments
0.9	April 30th 1999	• Assumption about error free link has been removed. • A simple error recovery, negotiation scheme and a resynchronisation/error indication scheme using RTS/CTS were added as proposed by IBM. • New document outline. • Added synchronisation using delimiters with COBS. • Added a support for CCITT-CRC. • Available Baud Rate Changed. • Only 8 bit data length is valid. • HCI Event packet type is added. • Available error type has been modified. • Editorial changes.
1.0 draft	July 5th 1999	• Assumption about error free link has been removed. • A simple error recovery, negotiation scheme and a resynchronisation/error indication scheme using RTS/CTS were added . • New document outline. • Added synchronisation using delimiters with COBS. • Added a support for CCITT-CRC • Available Baud Rate Changed. • Only 8 bit data length is valid. • HCI Event packet type is added. • Available error type has been modified. • Editorial changes.
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part H:4 HCI UART Transport Layer

Rev	Date	Comments
0.9	April 30th 1999	• was not a part of 0.8 • First revision. Based on HCI RS232 Transport Layer 0.80. • Added Default Settings, HW flow control and Error Recovery. • Improved description of RTS/CTS • Changed HCI packet indicator for HCI event packet
1.0 draft	July 5th 1999	• No changes since 0.9
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part I:1 / Test Mode

Rev	Date	Comments
0.9	April 30th 1999	• Transmitter Test: added pseudorandom bit sequence • Corresponding changes for LMP messages • Editorial Changes: Explanatory paragraph and figure added to the test packet format of the transmitter test • Proposal of a loopback alternative that has less demanding time constraints. • Reduced Hopping Sequence added • Modification/Clearification of TX packet format • Editorial changes • Statement in introduction that test mode may be used also for regulatory approval. • Description of delayed loopback added • proposed timing for reduced hopping scheme from Ericsson • timing for reduced hopping scheme refined • Loopback: transmission of NULL packet on failed HEC is not mandatory. • Added AUX1 packet explicitely to Figure 3 • Included AUX1 to Table 3 • Mentioned delayed loopback in first paragraph of Section 2.2 • Features request command is allowed, while in test mode. • Added codes for LMP packets according to LMP V0.9 review document Changed maximum packet length in Table 3 • FH timing: clearify that clock of tester is used • editorial changes • Clearifications about Whitening for both TX and Loopback mode • Clearification that RX and TX in the control command refer to the DUT. • Clearification in Table 3: For ACL packets a maximum length is given, for HV3 the exact length is given. • Editorial Changes
1.0 draft	July 5th 1999	• Description over use of whitening in transmitter test mode. • Added an exit command in the test mode control. • Editorial changes to whitening in transmitter test.
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Part I:2 Bluetooth Compliance Requirements

Rev	Date	Comments
1.0A	July 26th 1999	• The text in Section 6.6.1 “Definition of Bluetooth components” first paragraph has been revised.
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.
1.1	Dec 1st 2000	• The v1.0B compliance requirements is out-of-date with respect to the current qualification program and has some conflicts with the current approved PRD. Part I:2 is re-written to reflect the situation of today.

Part I:3 Test Control Interface

Rev	Date	Comments
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Appendix IV / Sample Data

Rev	Date	Comments
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Appendix V / Bluetooth Audio

Rev	Date	Comments
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.

Appendix VI / Baseband Timers

Rev	Date	Comments
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Appendix VII /Optional Paging Scheme

Rev	Date	Comments
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.

Appendix IX / Message Sequence Charts

Rev	Date	Comments
1.0 draft	July 5th 1999	• Command and Parameter updated; Chapter 7 added; Local- and Remote Hold-Mode removed; Chapter 8 clarified; No HCI-Num-Completed-Packets-Event in ACL- and SCO-Connection creation; LMP-feature-req/-res added in ACL-Connection-Setup; LMP-feature-req/-res removed in SCO-Connection-Setup • MSC "Onetime-Inquiry" and "Periodic-Inquiry" clarified with additional ID-Packet; • MSC "Local Loopback-Mode" exited with HCI_Write_Loopback_Mode instead with HCI_Reset; • MSC "Remote Loopback-Mode" exited with HCI_Write_Loopback_Mode instead with HCI_Reset; • MSC "Switch Role" updated with LMP_clkoffset_req/-res and LMP_slot_offset; • Editorial changes; • MSC "Switch Role": subscenario 2 added; Modified asToru's proposal; • MSC "ACL-Connection Request": slot-offset and clock-offset exchange subscenario3; • Figure 4: no loopback to Page/Page-Res in case of Role-Switch; • LMP_features_req/-res: added in MSC "Pairing" and MSC "Authentication", removed in MSC "Encryption and Setup Complete"; • MSC "Local Loopback Mode": two additional SCO-Connections added;
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars. The following figures are modified: 3.2, 3.3, 4.3 and 4.9
1.1	Dec 1st 2000	• Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

Appendix II

CONTRIBUTORS

Contributors

Part A:
Radio Specification

Steve Williams	3Com Corporation
Todor V. Cooklov	Aware
Poul Hove Kristensen	Digianswer A/S
Kurt B. Fischer	Hyper Corporation
Kevin D. Marquess	Hyper Corporation
Troy Beukema	IBM Corporation
Brian Gaucher	IBM Corporation
Jeff Schiffer	Intel Corporation
James P. Gilb	Mobilian
Rich L. Ditch	Motorola, Inc.
Paul Burgess	Nokia Mobile Phones
Olaf Joeressen	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Arto T. Palin	Nokia Mobile Phones
Steven J. Shellhammer	Symbol
Sven Mattisson	Telefonaktiebolaget LM Ericsson
Lars Nord (Section Owner)	Telefonaktiebolaget LM Ericsson
Anders Svensson	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Allen Hotari	Toshiba Corporation

Part B:
Baseband Specification

Kevin D. Marquess	Hyper Corporation
Chatschik Bisdikian	IBM Corporation
Kris Fleming	Intel Corporation
James P. Gilb	Mobilian
David E. Cypher	NIST
Nada Golmie	NIST

Olaf Joeressen	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Charlie Mellone	Motorola, Inc.
Harmke de Groot	Philips
Terry Bourk	Silicon Wave
Steven J. Shellhammer	Symbol
Jaap Haartsen	Telefonaktiebolaget LM Ericsson
Henrik Hedlund (Section Owner)	Telefonaktiebolaget LM Ericsson
Tobias Melin	Telefonaktiebolaget LM Ericsson
Joakim Persson	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Onn Haran	Texas Instruments
Thomas M. Siep	Texas Instruments
Ayse Findikli	Zeevo, Inc.

Part C:
Link Manager Protocol

Kim Schneider	Digianswer A/S
Toru Aihara	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Kris Fleming	Intel Corporation
David E. Cypher	NIST
Thomas Busse	Nokia Mobile Phones
Julien Corthial	Nokia Mobile Phones
Olaf Joeressen	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Dong Nguyen	Nokia Mobile Phones
Harmke de Groot	Philips
Terry Bourk	Silicon Wave
Johannes Elg	Telefonaktiebolaget LM Ericsson
Jaap Haartsen	Telefonaktiebolaget LM Ericsson
Tobias Melin (Section Owner)	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Onn Haran	Texas Instruments
John Mersh	TTPCom

Part D:
Logical Link Control and Adaptation Protocol Specification

Jon Burgess	3Com Corporation
Paul Moran	3Com Corporation
Doug Kogan	Extended Systems
Kevin D. Marquess	Hyper Corporation
Toru Aihara	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Kris Fleming	Intel Corporation
Uma Gadamsetty	Intel Corporation
Robert Hunter	Intel Corporation
Jon Inouye	Intel Corporation
Steve C. Lo	Intel Corporation
Chunrong Zhu	Intel Corporation
Sergey Solyanik	Microsoft Corporation
David E. Cypher	NIST
Nada Golmie	NIST
Thomas Busse	Nokia Mobile Phones
Rauno Makinen	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Petri Nykänen	Nokia Mobile Phones
Peter Ollikainen	Nokia Mobile Phones
Petri O. Nurminen	PumaTech
Johannes Elg	Telefonaktiebolaget LM Ericsson
Jaap Haartsen	Telefonaktiebolaget LM Ericsson
Elco Nijboer	Telefonaktiebolaget LM Ericsson
Ingemar Nilsson	Telefonaktiebolaget LM Ericsson
Stefan Runesson	Telefonaktiebolaget LM Ericsson
Gerrit Slot	Telefonaktiebolaget LM Ericsson
Johan Sörensen	Telefonaktiebolaget LM Ericsson
Goran Svennarp	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Thomas M. Siep	Texas Instruments
Kinoshita Katsuhiro	Toshiba Corporation

Part E:
Service Discovery Protocol (SDP)

Ned Plasson	3Com Corporation
John Avery	Convergence
Jason Kronz	Convergence
Chatschik Bisdikian	IBM Corporation
Parviz Kermani	IBM Corporation
Brent Miller	IBM Corporation
Dick Osterman	IBM Corporation
Bob Pascoe	IBM Corporation
Jon Inouye	Intel Corporation
Srikanth Kambhatla	Intel Corporation
Jay Eaglstun	Motorola, Inc.
Dale Farnsworth (Section Owner)	Motorola, Inc.
Jean-Michel Rosso	Motorola, Inc.
Jan Grönholm	Nokia Mobile Phones
Kati Rantala	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Johannes Elg	Telefonaktiebolaget LM Ericsson
Kazuaki Iwamura	Toshiba Corporation

Part F-1:
RFCOMM with TS 07.10

Christian Andersson	Telefonaktiebolaget LM Ericsson
Ingemar Nilsson	Telefonaktiebolaget LM Ericsson
Patrik Olsson	Telefonaktiebolaget LM Ericsson
Gerrit Slot	Telefonaktiebolaget LM Ericsson
Johan Sörensen (Section Owner)	Telefonaktiebolaget LM Ericsson
Donald Liechty	Extended Systems
Robert Hunter	Intel Corporation
Srikanth Kambhatla	Intel Corporation
Stan Adermann	Microsoft Corporation
Michael Camp	Nokia Mobile Phones
Pere Godia-Canero	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phones
Malcolm Box	Symbian

Part F-2:
IrDA Interoperability

Dave Suvak	Extended Systems
Chatschik Bisdikian	IBM Corporation
Brent Miller	IBM Corporation
Apratim Purakayastha	IBM Corporation
Aron Walker	IBM Corporation
Jon Inouye	Intel Corporation
Stephane Bouet (Section Owner)	Nokia Mobile Phones
Michael Camp	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phones
Peter Ollikainen	Nokia Mobile Phones
James Scales	Nokia Mobile Phones
David Kammer	Palm
Steve Rybicki	PumaTech
John Stossel	PumaTech
Christian Andersson	Telefonaktiebolaget LM Ericsson
Johannes Elg	Telefonaktiebolaget LM Ericsson

Patrik Olsson	Telefonaktiebolaget LM Ericsson
Johan Sörensen	Telefonaktiebolaget LM Ericsson

Part F-3:
Telephony Control Protocol Specification

Ken Morley	3Com Corporation
Richard Shaw	3Com Corporation
Sridhar Rajagopal	Intel Corporation
Ramu Ramakesavan	Intel Corporation
Brian Redding	Motorola, Inc.
Thomas Müller	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Jun'ichi Yoshizawa	Toshiba Corporation
Olof Dellien	Telefonaktiebolaget LM Ericsson
Erik Slotboom (Section Owner)	Telefonaktiebolaget LM Ericsson
Gert-jan van Lieshout	Telefonaktiebolaget LM Ericsson

Part F-4:
Interoperability Requirements for Bluetooth as a WAP Bearer

Jon Inouye	Intel Corporation
Michael T. Camp	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phones
Thomas Müller (Section Owner)	Nokia Mobile Phones
Kevin Wagner	Nokia Mobile Phones
Johannes Elg	Telefonaktiebolaget LM Ericsson
Robert Hed	Telefonaktiebolaget LM Ericsson
Patrick Olsson	Telefonaktiebolaget LM Ericsson

Part H-1:
Bluetooth Host Controller Interface Functional Specification

Todor Cooklev	3Com Corporation
Toru Aihara	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Nathan Lee	IBM Corporation
Akihiko Mizutani	IBM Corporation
Les Cline	Intel Corporation
Bailey Cross	Intel Corporation
Kris Fleming	Intel Corporation
Robert Hunter	Intel Corporation
Jon Inouye	Intel Corporation
Srikanth Kambhatla	Intel Corporation
Steve Lo	Intel Corporation
Vijay Suthar	Intel Corporation
Bruce P. Kraemer	Intersil
Greg Muchnik	Motorola, Inc.
David E. Cypher	NIST
Thomas Busse	Nokia Mobile Phones
Julien Courthial	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Dong Nguyen	Nokia Mobile Phones
Jürgen Schnitzler	Nokia Mobile Phones
Fujio Watanabe	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Johannes Elg	Telefonaktiebolaget LM Ericsson
Christian Johansson (Section Owner)	Telefonaktiebolaget LM Ericsson
Patrik Lundin	Telefonaktiebolaget LM Ericsson
Tobias Melin	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Thomas M. Siep	Texas Instruments
Masahiro Tada	Toshiba Corporation
John Mersh	TTPCom

Part H-2:
HCI USB Transport Layer

Nathan Lee	IBM Corporation
Les Cline	Intel Corporation
Brad Hosler	Intel Corporation
John Howard	Intel Corporation
Robert Hunter (Section Owner)	Intel Corporation
Srikanth Kambhatla	Intel Corporation
Kosta Koeman	Intel Corporation
John McGrath	Intel Corporation
Uwe Gondrum	Nokia Mobile Phones
Patrik Lundin	Telefonaktiebolaget LM Ericsson

Part H-3:
HCI RS232 Transport Layer

Steve Ross	Digianswer A/S
Chatschik Bisdikian	IBM Corporation
Edgar Kerstan	IBM Corporation
Nathan Lee (Section Owner)	IBM Corporation
Robert Hunter	Intel Corporation
Patrick Kane	Motorola, Inc.
Uwe Gondrum	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Johannes Elg	Telefonaktiebolaget LM Ericsson
Sven Jerlhagen	Telefonaktiebolaget LM Ericsson
Patrik Lundin	Telefonaktiebolaget LM Ericsson
Masahiro Tada	Toshiba Corporation

Part H-4:
HCI UART Transport Layer

Toru Aihara	IBM Corporation
Edgar Kerstan	IBM Corporation
Nathan Lee	IBM Corporation
Kris Fleming	Intel Corporation
Robert Hunter	Intel Corporation
Patrick Kane	Motorola, Inc.
Uwe Gondrum	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Johannes Elg	Telefonaktiebolaget LM Ericsson
Sven Jerlhagen	Telefonaktiebolaget LM Ericsson
Christian Johansson (Section Owner)	Telefonaktiebolaget LM Ericsson
Patrik Lundin	Telefonaktiebolaget LM Ericsson
Lars Novak	Telefonaktiebolaget LM Ericsson
Masahiro Tada	Toshiba Corporation

Part I-1:
Bluetooth Test Mode

Jeffrey Schiffer	Intel Corporation
David E. Cypher	NIST
Daniel Bencak	Nokia Mobile Phones
Arno Kefenbaum	Nokia Mobile Phones
Thomas Müller (Section Owner)	Nokia Mobile Phones
Roland Schmale	Nokia Mobile Phones
Fujio Watanabe	Nokia Mobile Phones
Stefan Agnani	Telefonaktiebolaget LM Ericsson
Mårten Mattsson	Telefonaktiebolaget LM Ericsson
Tobias Melin	Telefonaktiebolaget LM Ericsson
Lars Nord	Telefonaktiebolaget LM Ericsson
Fredrik Töörn	Telefonaktiebolaget LM Ericsson
John Mersh	TTPCom
Ayse Findikli	Zeevo, Inc.

Part I-2:
Bluetooth Compliance Requirements

BQRB (Editor)	
Wayne Park	3Com Corporation
Lawrence Jones	ComBit, Inc.
Gary Robinson	IBM Corporation
Georges Seuron	IBM Corporation
Rick Jessop	Intel Corporation
John Webb	Intel Corporation
Bruce Littlefield	Lucent Technologies, Inc.
Brian A. Redding	Motorola, Inc.
Waldemar Hontscha	Nokia Mobile Phones
Petri Morko	Nokia Mobile Phones
Magnus Hansson	Telefonaktiebolaget LM Ericsson
Magnus Sommansson	Telefonaktiebolaget LM Ericsson
Göran Svennarp	Telefonaktiebolaget LM Ericsson
Warren Allen	Toshiba Corporation
John Shi	Toshiba Corporation

Part I-3:
Bluetooth Test Control Interface

Mike Feldman	Motorola, Inc.
Thomas Müller	Nokia Mobile Phones
Stefan Agnani (Section Owner)	Telefonaktiebolaget LM Ericsson
Mårten Mattsson	Telefonaktiebolaget LM Ericsson
Dan Sönnerstam	Telefonaktiebolaget LM Ericsson

Appendix IV:
Encryption Sample Data

Thomas Müller	Nokia Mobile Phones
Thomas Sander	Nokia Mobile Phones
Joakim Persson (Section Owner)	Telefonaktiebolaget LM Ericsson

Appendix V: Bluetooth Audio

Magnus Hansson	Telefonaktiebolaget LM Ericsson
Fisseha Mekuria	Telefonaktiebolaget LM Ericsson
Mats Omrin	Telefonaktiebolaget LM Ericsson
Joakim Persson (Section Owner)	Telefonaktiebolaget LM Ericsson

Appendix VI: Baseband Timers

David E. Cyper	NIST
Jaap Haartsen (Section Owner)	Telefonaktiebolaget LM Ericsson
Joakim Persson	Telefonaktiebolaget LM Ericsson
Ayse Findikli	Zeevo, Inc.

Appendix VII: Optional Paging Scheme

Olaf Joeressen	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Markus Schetelig	Nokia Mobile Phones
Fujio Watanabe	Nokia Mobile Phones
Jaap Haartsen (Section Owner)	Telefonaktiebolaget LM Ericsson
Joakim Persson	Telefonaktiebolaget LM Ericsson
Ayse Findikli	Zeevo, Inc.

Appendix IX: Message Sequence Charts

Todor Cooklev	3Com Corporation
Toru Aihara	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Nathan Lee	IBM Corporation
Kris Fleming	Intel Corporation.
Greg Muchnik	Motorola, Inc.
David E. Cypher	NIST
Thomas Busse	Nokia Mobile Phones

Dong Nguyen (Section Owner)	Nokia Mobile Phones
Fujio Watanabe	Nokia Mobile Phones
Christian Johansson	Telefonaktiebolaget LM Ericsson
Tobias Melin	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments

Appendix III

ACRONYMS AND ABBREVIATIONS

List of Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
ACK	Acknowledge	
ACL link	Asynchronous Connection-Less link	Provides a packet-switched connection.(Master to any slave)
ACO	Authenticated Ciphering Offset	
AM_ADDR	Active Member Address	
AR_ADDR	Access Request Address	
ARQ	Automatic Repeat reQuest	
B		
BB	BaseBand	
BCH	Bose, Chaudhuri & Hocquenghem	Type of code The persons who discovered these codes in 1959 (H) and 1960 (B&C)
BD_ADDR	Bluetooth Device Address	
BER	Bit Error Rate	
BT	Bandwidth Time	
BT	Bluetooth	
C		
CAC	Channel Access Code	
CC	Call Control	
CL	Connectionless	
CODEC	COder DECoder	
COF	Ciphering Offset	
CRC	Cyclic Redundancy Check	
CVSD	Continuous Variable Slope Delta Modulation	
D		
DAC	Device Access Code	
DCE	Data Communication Equipment	

Acronym or abbreviation	Writing out in full	Which means
DCE	Data Circuit-Terminating Equipment	In serial communications, DCE refers to a device between the communication endpoints whose sole task is to facilitate the communications process; typically a modem
DCI	Default Check Initialization	
DH	Data-High Rate	Data packet type for high rate data
DIAC	Dedicated Inquiry Access Code	
DM	Data - Medium Rate	Data packet type for medium rate data
DTE	Data Terminal Equipment	In serial communications, DTE refers to a device at the endpoint of the communications path; typically a computer or terminal.
DTMF	Dual Tone Multiple Frequency	
DUT	Device Under Test	
DV	Data Voice	Data packet type for data and voice
E		
ETSI	European Telecommunications Standards Institute	
F		
FCC	Federal Communications Commission	
FEC	Forward Error Correction code	
FH	Frequency Hopping	
FHS	Frequency Hop Synchronization	
FIFO	First In First Out	
FSK	Frequency Shift Keying	type of modulation
FW	Firmware	
G		
GEOP	Generic Object Exchange Profile	
GFSK	Gaussian Frequency Shift Keying	
GIAC	General Inquiry Access Code	
GM	Group Management	

Acronym or abbreviation	Writing out in full	Which means
H		
HA	Host Application	SW using Bluetooth
HCI	Host Controller Interface	
HEC	Header-Error-Check	
HID	Human Interface Device	
HV	High quality Voice	e.g. HV1 packet
HW	Hardware	
I		
IAC	Inquiry Access Code	
IEEE	Institute of Electronic and Electrical Engineering	
IETF	Internet Engineering Task Force	
IP	Internet Protocol	
IrDA	Infra-red Data Association	
IrMC	Ir Mobile Communications	
ISDN	Integrated Services Digital Networks	
ISM	Industrial, Scientific, Medical	
IUT	Implementation Under Tes	
L		
L_CH	Logical Channel	
L2CA	Logical Link Control and Adaption	Logical Link Control And Management part of the Bluetooth protocol stack
L2CAP	Logical Link Control and Adaption Protocol	
LAP	Lower Address Part	
LC	Link Controller	Link Controller (or baseband) part of the Bluetooth protocol stack Low level Baseband protocol handler
LCP	Link Control Protocol	
LCSS	Link Controller Service Signalling	
LFSR	Linear Feedback Shift Register	
LM	Link Manager	

Acronym or abbreviation	Writing out in full	Which means
LMP	Link Manager Protocol	For LM peer to peer communication
LSB	Least Significant Bit	
M		
M	Master or Mandatory	
M_ADDR	Medium Access Control Address	
MAC	Medium Access Control	
MAPI	Messaging Application Procedure Interface	
MMI	Man Machine Interface	
MS	Mobile Station	
MS	Multiplexing sublayer	
MSB	Most Significant Bit	
MSC	Message Sequence Chart	
MTU	Maximum Transmission Unit	
MUX	Multiplexing Sublayer	a sublayer of the L2CAP layer
N		
NAK	Negative Acknowledge	
NAP	Non-significant Address Part	
O		
O	Optional	
OBEX	OBject EXchange protocol	
OCF	Opcode Command Field	
P		
PCM	Pulse Coded Modulation	
PCMCIA	Personal Computer Memory Card International Association	
PDU	Protocol Data Unit	a message
PIN	Personal Identification Number	
PM_ADDR	Parked Member Address	
PN	Pseudo-random Noise	
PnP	Plug and Play	
POTS	Plain Old Telephone system	

Acronym or abbreviation	Writing out in full	Which means
PPM	Part Per Million	
PPP	Point-to-Point Protocol	
PRBS	Pseudo Random Bit Sequence	
PRNG	Pseudo Random Noise Generation	
PSTN	Public Switched Telephone Network	
Q		
QoS	Quality of Service	
R		
RAND	Random number	
RF	Radio Frequency	
RFC	Request For Comments	
RFCOMM		Serial cable emulation protocol based on ETSI TS 07.10
RSSI	Received Signal Strength Indication	
RX	Receiver	
S		
S	Slave	
SAP	Service Access Points	
SAR	Segmentation and Reassembly	
SCO link	Synchronous Connection-Oriented link	Supports time-bounded information like voice. (Master to single slave)
SD	Service Discovery	
SDDB	Service Discovery Database	
SDP	Service Discovery Protocol	
SEQN	Sequential Numbering scheme	
SRES	Signed Response	
SS	Supplementary Services	
SSI	Signal Strength Indication	
SUT	System Under Test	
SW	Software	

Acronym or abbreviation	Writing out in full	Which means
T		
TAE	Terminal Adapter Equipment	
TBD	To Be Defined	
TC	Test Control	Test Control layer for the test interface
TCI	Test Control Interface	
TCP/IP	Transport Control Protocol/Internet Protocol	
TCS	Telephony Control protocol Specification	
TDD	Time-Division Duplex	
TTP	Tiny Transport Protocol between OBEX and UDP [TBD]	
TX	Transmit	
U		
UA	User Asynchronous	Asynchronous user data
UAP	Upper Address Part	
UART	Universal Asynchronous receiver Transmitter	
UC	User Control	
UDP/IP	User Datagram Protocol/Internet Protocol	
UI	User Isochronous	Isochronous user data
UIAC	Unlimited Inquiry Access Code	
US	User Synchronous	Synchronous user data
USB	Universal Serial Bus	
UT	Upper Tester	
W		
WAP	Wireless Application Protocol	
WUG	Wireless User Group	

Definitions

Baseband. The Bluetooth baseband specifies the medium access and physical layers procedures to support the exchange of real-time voice and data information streams and ad hoc networking between Bluetooth units.

Coverage area . The area where two Bluetooth units can exchange messages with acceptable quality and performance.

Host Terminal interface. Host terminal interface is the Interface between Bluetooth Host and Bluetooth Unit.

Inquiry. A Bluetooth unit transmits inquiry messages in order to discover the other Bluetooth units that are active within the coverage area. The Bluetooth units that capture inquiry messages may send a response to the inquiring Bluetooth unit. The response contains information about the Bluetooth unit itself and its Bluetooth Host.

Isochronous user channel . Channel used for time bounded information like i.e. compressed audio (ACL link).

Logical Channel. The different types of channels on a Physical Link.

Bluetooth Host. Bluetooth Host is a computing device, peripheral, cellular telephone, access point to PSTN network, etc. A Bluetooth Host attached to a Bluetooth unit may communicate with other Bluetooth Hosts attached to their Bluetooth units as well. The communication channel through the Bluetooth units provides almost wire-like transparency.

Bluetooth Unit. Bluetooth Unit is a voice/data circuit equipment for a short-range wireless communication link. It allows voice and data communications between Bluetooth Hosts.

Bluetooth. Bluetooth is a wireless communication link, operating in the unlicensed ISM band at 2,4 GHz using a frequency hopping transceiver. It allows real-time voice and data communications between Bluetooth Hosts. The link protocol is based on time slots.

Packet. Format of aggregated bits that can be transmitted in 1, 3, or 5 time slots.

Paging. An Bluetooth unit transmits paging messages in order to set up a communication link to another Bluetooth unit who is active within the coverage area.

Physical Channel. Synchronized RF hopping sequence in a piconet

Physical Link. Connection between devices.

Piconet. In the Bluetooth system, the channel is shared among several Bluetooth units. The units sharing a common channel constitute a piconet.

RFCOMM Client. An RFCOMM client is an application that requests a connection to another application (RFCOMM server).

RFCOMM initiator. The device initiating the RFCOMM session, i.e.setting up RFCOMM channel on L2CAP and starting RFCOMM multiplexing with the SABM command on DLCI 0 (zero).

RFCOMM Server. An RFCOMM server is an application that awaits a connection from an RFCOMM client on another device. What happens after such a connection is established is out of scope of this definition.

RFCOMM Server Channel. This is a subfield of the TS 07.10 DLCI number. This abstraction is used to allow both server and client applications to reside on both sides of an RFCOMM session.

Service Discovery. The ability to discover the capability of connecting devices or hosts

Scatternet. Two or more piconets co-located in the same area (with or without inter-piconet communication).

Time Slot. The Physical Channel is divided into 625 µs long time slots.

Appendix IV

SAMPLE DATA

This appendix contains sample data for various parts of the Bluetooth baseband specification. All sample data are provided for reference purpose only; they are intended as a complement to the definitions provided elsewhere in the specification. They can be used to check the behavior of an implementation and avoid misunderstandings. Fulfilling these sample data is a necessary but not sufficient condition for an implementation to be fully Bluetooth compliant.

CONTENTS

1 Encryption Sample Data 920
1.1 Generating Kc' from Kc, ... 920
1.2 First Set of Sample Data 923
1.3 Second Set of Sample Data 931
1.4 Third Set of Samples 939
1.5 Fourth Set of Samples 947

2 Frequency Hopping Sample Data—Mandatory Scheme 955
2.1 The 79-hop System Sample Data 955
2.1.1 First set 956
2.1.2 Second set 958
2.1.3 Third set 960
2.2 The 23-hop System Sample Data 961
2.2.1 First set 961
2.2.2 Second set 964
2.2.3 Third set 966

3 Access Code Sample Data 968

4 HEC and Packet Header Sample Data 971

5 CRC Sample Data 972

6 Complete Sample Packets 973
6.1 Example of DH1 Packet 973
6.2 Example of DM1 Packet 974

7 Whitening Sequence Sample Data 975

8 FEC Sample Data 978

9 Encryption Key Sample Data 979
9.1 Four Tests of E1 979
9.2 Four Tests of E21 983
9.3 Three Tests of E22 986
9.4 Tests of E22 With Pin Augmenting 989
9.5 Four Tests of E3 999

1 ENCRYPTION SAMPLE DATA

This part consist of four sets of sample data for the encryption process.

With respect to the functional description of the encryption engine in the Bluetooth baseband specification, the contents of registers and resulting concurrent values are listed as well. This by no means excludes different implementations (as far as they produce the same encryption stream) but is intended to describe the functional behavior.

In case of misunderstandings or inconsistencies, these sample data form the normative reference.

1.1 GENERATING KC' FROM KC,

where Kc'(x) = g2(x)(Kc(x) mod g1(x)).

```
Note: All polynomials are in hexadecimal notation.
      'L' is the effective key length in bytes.
      The notation 'p: [m]' implies that deg(p(x)) = m.

                              MSB                                     LSB
L = 1
g1:                [8]        00000000 00000000 00000000 0000011d
g2:                [119]      00e275a0 abd218d4 cf928b9b bf6cb08f
Kc:                           a2b230a4 93f281bb 61a85b82 a9d4a30e
Kc mod g1:         [7]        00000000 00000000 00000000 0000009f
g2(Kc mod g1):     [126]      7aa16f39 59836ba3 22049a7b 87f1d8a5
-------------------------------------------------------------
L = 2
g1:                [16]       00000000 00000000 00000000 0001003f
g2:                [112]      0001e3f6 3d7659b3 7f18c258 cff6efef
Kc:                           64e7df78 bb7ccaa4 61433123 5b3222ad
Kc mod g1:         [12]       00000000 00000000 00000000 00001ff0
g2(Kc mod g1):     [124]      142057bb 0bceac4c 58bd142e 1e710a50
-------------------------------------------------------------
L = 3
g1:                [24]       00000000 00000000 00000000 010000db
g2:                [104]      000001be f66c6c3a b1030a5a 1919808b
Kc:                           575e5156 ba685dc6 112124ac edb2c179
Kc mod g1:         [23]       00000000 00000000 00000000 008ddbc8
g2(Kc mod g1):     [127]      d56d0adb 8216cb39 7fe3c591 1ff95618
-------------------------------------------------------------
L = 4
g1:                [32]       00000000 00000000 00000001 000000af
g2:                [96]       00000001 6ab89969 de17467f d3736ad9
Kc:                           8917b4fc 403b6db2 1596b86d 1cb8adab
Kc mod g1:         [31]       00000000 00000000 00000000 aa1e78aa
g2(Kc mod g1):     [127]      91910128 b0e2f5ed a132a03e af3d8cda
-------------------------------------------------------------
```

Appendix IV - Sample Data

```
L = 5
g1:                [40]        00000000 00000000 00000100 00000039
g2:                [88]        00000000 01630632 91da50ec 55715247
Kc:                            785c915b dd25b9c6 0102ab00 b6cd2a68
Kc mod g1:         [38]        00000000 00000000 0000007f 13d44436
g2(Kc mod g1):     [126]       6fb5651c cb80c8d7 ea1ee56d f1ec5d02
-------------------------------------------------------------
L = 6
g1:                [48]        00000000 00000000 00010000 00000291
g2:                [77]        00000000 00002c93 52aa6cc0 54468311
Kc:                            5e77d19f 55ccd7d5 798f9a32 3b83e5d8
Kc mod g1:         [47]        00000000 00000000 000082eb 4af213ed
g2(Kc mod g1):     [124]       16096bcb afcf8def 1d226a1b 4d3f9a3d
-------------------------------------------------------------
```

```
L = 7
g1:                [56]        00000000 00000000 01000000 00000095
g2:                [71]        00000000 000000b3 f7fffce2 79f3a073
Kc:                            05454e03 8ddcfbe3 ed024b2d 92b7f54c
Kc mod g1:         [55]        00000000 00000000 0095b8a4 8eb816da
g2(Kc mod g1):     [126]       50f9c0d4 e3178da9 4a09fe0d 34f67b0e
------------------------------------------------------------
L = 8
g1:                [64]        00000000 00000001 00000000 0000001b
g2:                [63]        00000000 00000000 a1ab815b c7ec8025
Kc:                            7ce149fc f4b38ad7 2a5d8a41 eb15ba31
Kc mod g1:         [63]        00000000 00000000 8660806c 1865deec
g2(Kc mod g1):     [126]       532c36d4 5d0954e0 922989b6 826f78dc
------------------------------------------------------------
L = 9
g1:                [72]        00000000 00000100 00000000 00000609
g2:                [49]        00000000 00000000 0002c980 11d8b04d
Kc:                            5eeff7ca 84fc2782 9c051726 3df6f36e
Kc mod g1:         [71]        00000000 00000083 58ccb7d0 b95d3c71
g2(Kc mod g1):     [120]       016313f6 0d3771cf 7f8e4bb9 4aa6827d
------------------------------------------------------------
L = 10
g1:                [80]        00000000 00010000 00000000 00000215
g2:                [42]        00000000 00000000 0000058e 24f9a4bb
Kc:                            7b13846e 88beb4de 34e7160a fd44dc65
Kc mod g1:         [79]        00000000 0000b4de 34171767 f36981c3
g2(Kc mod g1):     [121]       023bc1ec 34a0029e f798dcfb 618ba58d
------------------------------------------------------------
L = 11
g1:                [88]        00000000 01000000 00000000 0000013b
g2:                [35]        00000000 00000000 0000000c a76024d7
Kc:                            bda6de6c 6e7d757e 8dfe2d49 9a181193
Kc mod g1:         [86]        00000000 007d757e 8dfe88aa 2fcee371
g2(Kc mod g1):     [121]       022e08a9 3aa51d8d 2f93fa78 85cc1f87
------------------------------------------------------------
L = 12
g1:                [96]        00000001 00000000 00000000 000000dd
g2:                [28]        00000000 00000000 00000000 1c9c26b9
Kc:                            e6483b1c 2cdb1040 9a658f97 c4efd90d
Kc mod g1:         [93]        00000000 2cdb1040 9a658fd7 5b562e41
g2(Kc mod g1):     [121]       030d752b 216fe29b b880275c d7e6f6f9
------------------------------------------------------------
L = 13
g1:                [104]       00000100 00000000 00000000 0000049d
g2:                [21]        00000000 00000000 00000000 0026d9e3
Kc:                            d79d281d a2266847 6b223c46 dc0ab9ee
Kc mod g1:         [100]       0000001d a2266847 6b223c45 e1fc5fa6
g2(Kc mod g1):     [121]       03f11138 9cebf919 00b93808 4ac158aa
------------------------------------------------------------
```

```
L = 14
g1:                [112]       00010000 00000000 00000000 0000014f
g2:                [14]        00000000 00000000 00000000 00004377
Kc:                            cad9a65b 9fca1c1d a2320fcf 7c4ae48e
Kc mod g1:         [111]       0000a65b 9fca1c1d a2320fcf 7cb6a909
g2(Kc mod g1):     [125]       284840fd f1305f3c 529f5703 76adf7cf
------------------------------------------------------------
L = 15
g1:                [120]       01000000 00000000 00000000 000000e7
g2:                [7]         00000000 00000000 00000000 00000089
Kc:                            21f0cc31 049b7163 d375e9e1 06029809
Kc mod g1:         [119]       00f0cc31 049b7163 d375e9e1 0602840e
g2(Kc mod g1):     [126]       7f10b53b 6df84b94 f22e566a 3754a37e
------------------------------------------------------------
L = 16
g1:                [128]       00000001 00000000 00000000 00000000 00000000
g2:                [0]         00000000 00000000 00000000 00000001
Kc:                            35ec8fc3 d50ccd32 5f2fd907 bde206de
Kc mod g1:         [125]       35ec8fc3 d50ccd32 5f2fd907 bde206de
g2(Kc mod g1):     [125]       35ec8fc3 d50ccd32 5f2fd907 bde206de
------------------------------------------------------------
```

1.2 FIRST SET OF SAMPLE DATA

```
Initial values for the key, pan address and clock

K'c1[0]   = 00   K'c1[1]   = 00    K'c1[2]   = 00    K'c1[3]  = 00
K'c1[4]   = 00   K'c1[5]   = 00    K'c1[6]   = 00    K'c1[7]  = 00
K'c1[8]   = 00   K'c1[9]   = 00    K'c1[10]  = 00    K'c1[11] = 00
K'c1[12]  = 00   K'c1[13]  = 00    K'c1[14]  = 00    K'c1[15] = 00

Addr1[0] = 00   Addr1[1] = 00    Addr1[2] = 00
Addr1[3] = 00   Addr1[4] = 00    Addr1[5] = 00


CL1[0]  = 00   CL1[1]  = 00    CL1[2]  = 00    CL1[3]  = 00


=============================================================================================
         Fill LFSRs with initial data
=============================================================================================

  t  clk#    LFSR1    LFSR2     LFSR3      LFSR4          X1 X2 X3 X4   Z  C[t+1] C[t] C[t-1]

  0     0    0000000* 00000000* 000000000* 0000000000*     0  0  0  0    0  00     00   00
  1     1    0000000* 00000001* 000000000* 0000000001*     0  0  0  0    0  00     00   00
  2     2    0000000* 00000002* 000000000* 0000000003*     0  0  0  0    0  00     00   00
  3     3    0000000* 00000004* 000000000* 0000000007*     0  0  0  0    0  00     00   00
  4     4    0000000* 00000008* 000000000* 000000000E*     0  0  0  0    0  00     00   00
  5     5    0000000* 00000010* 000000000* 000000001C*     0  0  0  0    0  00     00   00
  6     6    0000000* 00000020* 000000000* 0000000038*     0  0  0  0    0  00     00   00
```

7	7	0000000*	00000040*	000000000*	0000000070*	0	0	0	0	0	00	00	00
8	8	0000000*	00000080*	000000000*	00000000E0*	0	0	0	0	0	00	00	00
9	9	0000000*	00000100*	000000000*	00000001C0*	0	0	0	0	0	00	00	00
10	10	0000000*	00000200*	000000000*	0000000380*	0	0	0	0	0	00	00	00
11	11	0000000*	00000400*	000000000*	0000000700*	0	0	0	0	0	00	00	00
12	12	0000000*	00000800*	000000000*	0000000E00*	0	0	0	0	0	00	00	00
13	13	0000000*	00001000*	000000000*	0000001C00*	0	0	0	0	0	00	00	00
14	14	0000000*	00002000*	000000000*	0000003800*	0	0	0	0	0	00	00	00
15	15	0000000*	00004000*	000000000*	0000007000*	0	0	0	0	0	00	00	00
16	16	0000000*	00008000*	000000000*	000000E000*	0	0	0	0	0	00	00	00
17	17	0000000*	00010000*	000000000*	000001C000*	0	0	0	0	0	00	00	00
18	18	0000000*	00020000*	000000000*	0000038000*	0	0	0	0	0	00	00	00
19	19	0000000*	00040000*	000000000*	0000070000*	0	0	0	0	0	00	00	00
20	20	0000000*	00080000*	000000000*	00000E0000*	0	0	0	0	0	00	00	00
21	21	0000000*	00100000*	000000000*	00001C0000*	0	0	0	0	0	00	00	00
22	22	0000000*	00200000*	000000000*	0000380000*	0	0	0	0	0	00	00	00
23	23	0000000*	00400000*	000000000*	0000700000*	0	0	0	0	0	00	00	00
24	24	0000000*	00800000*	000000000*	0000E00000*	0	1	0	0	1	00	00	00
25	25	0000000*	01000000*	000000000*	0001C00000*	0	0	0	0	0	00	00	00
26	26	0000000	02000000*	000000000*	0003800000*	0	0	0	0	0	00	00	00
27	27	0000000	04000000*	000000000*	0007000000*	0	0	0	0	0	00	00	00
28	28	0000000	08000000*	000000000*	000E000000*	0	0	0	0	0	00	00	00
29	29	0000000	10000000*	000000000*	001C000000*	0	0	0	0	0	00	00	00
30	30	0000000	20000000*	000000000*	0038000000*	0	0	0	0	0	00	00	00
31	31	0000000	40000000*	000000000*	0070000000*	0	0	0	0	0	00	00	00
32	32	0000000	00000001	000000000*	00E0000000*	0	0	0	1	1	00	00	00
33	33	0000000	00000002	000000000*	01C0000000*	0	0	0	1	1	00	00	00
34	34	0000000	00000004	000000000	0380000000*	0	0	0	1	1	00	00	00
35	35	0000000	00000008	000000000	0700000000*	0	0	0	0	0	00	00	00
36	36	0000000	00000010	000000000	0E00000000*	0	0	0	0	0	00	00	00
37	37	0000000	00000020	000000000	1C00000000*	0	0	0	0	0	00	00	00
38	38	0000000	00000040	000000000	3800000000*	0	0	0	0	0	00	00	00
39	39	0000000	00000080	000000000	7000000000*	0	0	0	0	0	00	00	00

Start clocking Summation Combiner

40	1	0000000	00000100	000000000	6000000001	0	0	0	0	0	00	00	00
41	2	0000000	00000200	000000000	4000000003	0	0	0	0	0	00	00	00
42	3	0000000	00000400	000000000	0000000007	0	0	0	0	0	00	00	00
43	4	0000000	00000800	000000000	000000000E	0	0	0	0	0	00	00	00
44	5	0000000	00001001	000000000	000000001D	0	0	0	0	0	00	00	00
45	6	0000000	00002002	000000000	000000003B	0	0	0	0	0	00	00	00
46	7	0000000	00004004	000000000	0000000077	0	0	0	0	0	00	00	00
47	8	0000000	00008008	000000000	00000000EE	0	0	0	0	0	00	00	00
48	9	0000000	00010011	000000000	00000001DD	0	0	0	0	0	00	00	00
49	10	0000000	00020022	000000000	00000003BB	0	0	0	0	0	00	00	00
50	11	0000000	00040044	000000000	0000000777	0	0	0	0	0	00	00	00
51	12	0000000	00080088	000000000	0000000EEE	0	0	0	0	0	00	00	00
52	13	0000000	00100110	000000000	0000001DDD	0	0	0	0	0	00	00	00
53	14	0000000	00200220	000000000	0000003BBB	0	0	0	0	0	00	00	00
54	15	0000000	00400440	000000000	0000007777	0	0	0	0	0	00	00	00
55	16	0000000	00800880	000000000	000000EEEE	0	1	0	0	1	00	00	00
56	17	0000000	01001100	000000000	000001DDDD	0	0	0	0	0	00	00	00
57	18	0000000	02002200	000000000	000003BBBB	0	0	0	0	0	00	00	00
58	19	0000000	04004400	000000000	0000077777	0	0	0	0	0	00	00	00
59	20	0000000	08008800	000000000	00000EEEEE	0	0	0	0	0	00	00	00
60	21	0000000	10011000	000000000	00001DDDDD	0	0	0	0	0	00	00	00

61	22	0000000	20022000	000000000	00003BBBBB	0	0	0	0	0	00	00	00
62	23	0000000	40044000	000000000	0000777777	0	0	0	0	0	00	00	00
63	24	0000000	00088001	000000000	0000EEEEEE	0	0	0	0	0	00	00	00
64	25	0000000	00110003	000000000	0001DDDDDD	0	0	0	0	0	00	00	00
65	26	0000000	00220006	000000000	0003BBBBBB	0	0	0	0	0	00	00	00
66	27	0000000	0044000C	000000000	0007777777	0	0	0	0	0	00	00	00
67	28	0000000	00880018	000000000	000EEEEEEE	0	1	0	0	1	00	00	00
68	29	0000000	01100031	000000000	001DDDDDDC	0	0	0	0	0	00	00	00
69	30	0000000	02200062	000000000	003BBBBBB8	0	0	0	0	0	00	00	00
70	31	0000000	044000C4	000000000	0077777770	0	0	0	0	0	00	00	00
71	32	0000000	08800188	000000000	00EEEEEEE0	0	1	0	1	0	01	00	00
72	33	0000000	11000311	000000000	01DDDDDDC1	0	0	0	1	0	00	01	00
73	34	0000000	22000622	000000000	03BBBBBB83	0	0	0	1	1	11	00	01
74	35	0000000	44000C44	000000000	0777777707	0	0	0	0	1	10	11	00
75	36	0000000	08001888	000000000	0EEEEEEE0E	0	0	0	1	1	01	10	11
76	37	0000000	10003111	000000000	1DDDDDDC1D	0	0	0	1	0	01	01	10
77	38	0000000	20006222	000000000	3BBBBBB83B	0	0	0	1	0	11	01	01
78	39	0000000	4000C444	000000000	7777777077	0	0	0	0	1	01	11	01
79	40	0000000	00018888	000000000	6EEEEEE0EF	0	0	0	1	0	10	01	11
80	41	0000000	00031110	000000000	5DDDDDC1DE	0	0	0	1	1	00	10	01
81	42	0000000	00062220	000000000	3BBBBB83BC	0	0	0	1	1	01	00	10
82	43	0000000	000C4440	000000000	7777770779	0	0	0	0	1	01	01	00
83	44	0000000	00188880	000000000	6EEEEE0EF2	0	0	0	1	0	11	01	01
84	45	0000000	00311100	000000000	5DDDDC1DE5	0	0	0	1	0	10	11	01
85	46	0000000	00622200	000000000	3BBBB83BCB	0	0	0	1	1	01	10	11
86	47	0000000	00C44400	000000000	7777707797	0	1	0	0	0	01	01	10
87	48	0000000	01888801	000000000	6EEEE0EF2F	0	1	0	1	1	11	01	01
88	49	0000000	03111003	000000000	5DDDC1DE5E	0	0	0	1	0	10	11	01
89	50	0000000	06222006	000000000	3BBB83BCBC	0	0	0	1	1	01	10	11
90	51	0000000	0C44400C	000000000	7777077979	0	0	0	0	1	00	01	10
91	52	0000000	18888018	000000000	6EEE0EF2F2	0	1	0	1	0	10	00	01
92	53	0000000	31110030	000000000	5DDC1DE5E5	0	0	0	1	1	11	10	00
93	54	0000000	62220060	000000000	3BB83BCBCB	0	0	0	1	0	00	11	10
94	55	0000000	444400C1	000000000	7770779797	0	0	0	0	0	10	00	11
95	56	0000000	08880183	000000000	6EE0EF2F2F	0	1	0	1	0	00	10	00
96	57	0000000	11100307	000000000	5DC1DE5E5F	0	0	0	1	1	01	00	10
97	58	0000000	2220060E	000000000	3B83BCBCBF	0	0	0	1	0	00	01	00
98	59	0000000	44400C1C	000000000	770779797E	0	0	0	0	0	11	00	01
99	60	0000000	08801838	000000000	6E0EF2F2FC	0	1	0	0	0	01	11	00
100	61	0000000	11003070	000000000	5C1DE5E5F8	0	0	0	0	1	11	01	11
101	62	0000000	220060E0	000000000	383BCBCBF0	0	0	0	0	1	01	11	01
102	63	0000000	4400C1C0	000000000	70779797E0	0	0	0	0	1	11	01	11
103	64	0000000	08018380	000000000	60EF2F2FC1	0	0	0	1	0	10	11	01
104	65	0000000	10030701	000000000	41DE5E5F82	0	0	0	1	1	01	10	11
105	66	0000000	20060E02	000000000	03BCBCBF04	0	0	0	1	0	01	01	10
106	67	0000000	400C1C05	000000000	0779797E09	0	0	0	0	1	10	01	01
107	68	0000000	0018380A	000000000	0EF2F2FC12	0	0	0	1	1	00	10	01
108	69	0000000	00307015	000000000	1DE5E5F825	0	0	0	1	1	01	00	10
109	70	0000000	0060E02A	000000000	3BCBCBF04B	0	0	0	1	0	00	01	00
110	71	0000000	00C1C055	000000000	779797E097	0	1	0	1	0	10	00	01
111	72	0000000	018380AA	000000000	6F2F2FC12F	0	1	0	0	1	11	10	00
112	73	0000000	03070154	000000000	5E5E5F825E	0	0	0	0	1	11	11	10
113	74	0000000	060E02A8	000000000	3CBCBF04BC	0	0	0	1	0	11	11	11
114	75	0000000	0C1C0550	000000000	79797E0979	0	0	0	0	1	00	11	11
115	76	0000000	18380AA0	000000000	72F2FC12F2	0	0	0	1	1	10	00	11
116	77	0000000	30701541	000000000	65E5F825E5	0	0	0	1	1	11	10	00
117	78	0000000	60E02A82	000000000	4BCBF04BCB	0	1	0	1	1	00	11	10

118	79	0000000	41C05505	000000000	1797E09796	0	1	0	1	0	11	00	11
119	80	0000000	0380AA0A	000000000	2F2FC12F2C	0	1	0	0	0	01	11	00
120	81	0000000	07015415	000000000	5E5F825E59	0	0	0	0	1	11	01	11
121	82	0000000	0E02A82A	000000000	3CBF04BCB2	0	0	0	1	0	10	11	01
122	83	0000000	1C055054	000000000	797E097964	0	0	0	0	0	01	10	11
123	84	0000000	380AA0A8	000000000	72FC12F2C9	0	0	0	1	0	01	01	10
124	85	0000000	70154151	000000000	65F825E593	0	0	0	1	0	11	01	01
125	86	0000000	602A82A3	000000000	4BF04BCB26	0	0	0	1	0	10	11	01
126	87	0000000	40550546	000000000	17E097964C	0	0	0	1	1	01	10	11
127	88	0000000	00AA0A8D	000000000	2FC12F2C99	0	1	0	1	1	01	01	10
128	89	0000000	0154151A	000000000	5F825E5932	0	0	0	1	0	11	01	01
129	90	0000000	02A82A34	000000000	3F04BCB264	0	1	0	0	0	10	11	01
130	91	0000000	05505468	000000000	7E097964C9	0	0	0	0	0	01	10	11
131	92	0000000	0AA0A8D0	000000000	7C12F2C992	0	1	0	0	0	01	01	10
132	93	0000000	154151A1	000000000	7825E59324	0	0	0	0	1	10	01	01
133	94	0000000	2A82A342	000000000	704BCB2648	0	1	0	0	1	00	10	01
134	95	0000000	55054684	000000000	6097964C91	0	0	0	1	1	01	00	10
135	96	0000000	2A0A8D09	000000000	412F2C9923	0	0	0	0	1	01	01	00
136	97	0000000	54151A12	000000000	025E593246	0	0	0	0	1	10	01	01
137	98	0000000	282A3424	000000000	04BCB2648D	0	0	0	1	1	00	10	01
138	99	0000000	50546848	000000000	097964C91A	0	0	0	0	0	01	00	10
139	100	0000000	20A8D090	000000000	12F2C99235	0	1	0	1	1	00	01	00
140	101	0000000	4151A120	000000000	25E593246A	0	0	0	1	1	11	00	01
141	102	0000000	02A34240	000000000	4BCB2648D5	0	1	0	1	1	01	11	00
142	103	0000000	05468481	000000000	17964C91AB	0	0	0	1	0	10	01	11
143	104	0000000	0A8D0903	000000000	2F2C992357	0	1	0	0	1	00	10	01
144	105	0000000	151A1206	000000000	5E593246AE	0	0	0	0	0	01	00	10
145	106	0000000	2A34240C	000000000	3CB2648D5C	0	0	0	1	0	00	01	00
146	107	0000000	54684818	000000000	7964C91AB8	0	0	0	0	0	11	00	01
147	108	0000000	28D09030	000000000	72C9923571	0	1	0	1	1	01	11	00
148	109	0000000	51A12060	000000000	6593246AE2	0	1	0	1	1	10	01	11
149	110	0000000	234240C0	000000000	4B2648D5C5	0	0	0	0	0	00	10	01
150	111	0000000	46848180	000000000	164C91AB8A	0	1	0	0	1	01	00	10
151	112	0000000	0D090301	000000000	2C99235714	0	0	0	1	0	00	01	00
152	113	0000000	1A120602	000000000	593246AE28	0	0	0	0	0	11	00	01
153	114	0000000	34240C04	000000000	32648D5C51	0	0	0	0	1	10	11	00
154	115	0000000	68481809	000000000	64C91AB8A2	0	0	0	1	1	01	10	11
155	116	0000000	50903012	000000000	4992357144	0	1	0	1	1	01	01	10
156	117	0000000	21206024	000000000	13246AE288	0	0	0	0	1	10	01	01
157	118	0000000	4240C048	000000000	2648D5C511	0	0	0	0	0	00	10	01
158	119	0000000	04818090	000000000	4C91AB8A23	0	1	0	1	0	00	00	10
159	120	0000000	09030120	000000000	1923571446	0	0	0	0	0	00	00	00
160	121	0000000	12060240	000000000	3246AE288D	0	0	0	0	0	00	00	00
161	122	0000000	240C0480	000000000	648D5C511B	0	0	0	1	1	00	00	00
162	123	0000000	48180900	000000000	491AB8A237	0	0	0	0	0	00	00	00
163	124	0000000	10301200	000000000	123571446F	0	0	0	0	0	00	00	00
164	125	0000000	20602400	000000000	246AE288DF	0	0	0	0	0	00	00	00
165	126	0000000	40C04800	000000000	48D5C511BE	0	1	0	1	0	01	00	00
166	127	0000000	01809001	000000000	11AB8A237D	0	1	0	1	1	00	01	00
167	128	0000000	03012002	000000000	23571446FA	0	0	0	0	0	11	00	01
168	129	0000000	06024004	000000000	46AE288DF5	0	0	0	1	0	01	11	00
169	130	0000000	0C048008	000000000	0D5C511BEA	0	0	0	0	1	11	01	11
170	131	0000000	18090011	000000000	1AB8A237D5	0	0	0	1	0	10	11	01
171	132	0000000	30120022	000000000	3571446FAA	0	0	0	0	0	01	10	11
172	133	0000000	60240044	000000000	6AE288DF55	0	0	0	1	0	01	01	10
173	134	0000000	40480089	000000000	55C511BEAA	0	0	0	1	0	11	01	01
174	135	0000000	00900113	000000000	2B8A237D54	0	1	0	1	1	10	11	01

175	136	0000000	01200227	000000000	571446FAA8	0	0	0	0	0	01	10	11
176	137	0000000	0240044E	000000000	2E288DF550	0	0	0	0	1	00	01	10
177	138	0000000	0480089C	000000000	5C511BEAA0	0	1	0	0	1	11	00	01
178	139	0000000	09001138	000000000	38A237D540	0	0	0	1	0	01	11	00
179	140	0000000	12002270	000000000	71446FAA81	0	0	0	0	1	11	01	11
180	141	0000000	240044E0	000000000	6288DF5503	0	0	0	1	0	10	11	01
181	142	0000000	480089C0	000000000	4511BEAA06	0	0	0	0	0	01	10	11
182	143	0000000	10011381	000000000	0A237D540D	0	0	0	0	1	00	01	10
183	144	0000000	20022702	000000000	1446FAA81A	0	0	0	0	0	11	00	01
184	145	0000000	40044E04	000000000	288DF55035	0	0	0	1	0	01	11	00
185	146	0000000	00089C08	000000000	511BEAA06A	0	0	0	0	1	11	01	11
186	147	0000000	00113810	000000000	2237D540D5	0	0	0	0	1	01	11	01
187	148	0000000	00227021	000000000	446FAA81AA	0	0	0	0	1	11	01	11
188	149	0000000	0044E042	000000000	08DF550355	0	0	0	1	0	10	11	01
189	150	0000000	0089C085	000000000	11BEAA06AA	0	1	0	1	0	10	10	11
190	151	0000000	0113810A	000000000	237D540D54	0	0	0	0	0	10	10	10
191	152	0000000	02270215	000000000	46FAA81AA9	0	0	0	1	1	10	10	10
192	153	0000000	044E042A	000000000	0DF5503553	0	0	0	1	1	10	10	10
193	154	0000000	089C0854	000000000	1BEAA06AA7	0	1	0	1	0	01	10	10
194	155	0000000	113810A8	000000000	37D540D54E	0	0	0	1	0	01	01	10
195	156	0000000	22702150	000000000	6FAA81AA9D	0	0	0	1	0	11	01	01
196	157	0000000	44E042A0	000000000	5F5503553A	0	1	0	0	0	10	11	01
197	158	0000000	09C08540	000000000	3EAA06AA75	0	1	0	1	0	10	10	11
198	159	0000000	13810A80	000000000	7D540D54EA	0	1	0	0	1	10	10	10
199	160	0000000	27021500	000000000	7AA81AA9D5	0	0	0	1	1	10	10	10
200	161	0000000	4E042A00	000000000	75503553AB	0	0	0	0	0	10	10	10
201	162	0000000	1C085400	000000000	6AA06AA756	0	0	0	1	1	10	10	10
202	163	0000000	3810A800	000000000	5540D54EAC	0	0	0	0	0	10	10	10
203	164	0000000	70215000	000000000	2A81AA9D58	0	0	0	1	1	10	10	10
204	165	0000000	6042A001	000000000	5503553AB0	0	0	0	0	0	10	10	10
205	166	0000000	40854002	000000000	2A06AA7561	0	1	0	0	1	10	10	10
206	167	0000000	010A8004	000000000	540D54EAC3	0	0	0	0	0	10	10	10
207	168	0000000	02150009	000000000	281AA9D586	0	0	0	0	0	10	10	10
208	169	0000000	042A0012	000000000	503553AB0C	0	0	0	0	0	10	10	10
209	170	0000000	08540024	000000000	206AA75618	0	0	0	0	0	10	10	10
210	171	0000000	10A80048	000000000	40D54EAC30	0	1	0	1	0	01	10	10
211	172	0000000	21500091	000000000	01AA9D5861	0	0	0	1	0	01	01	10
212	173	0000000	42A00122	000000000	03553AB0C3	0	1	0	0	0	11	01	01
213	174	0000000	05400244	000000000	06AA756186	0	0	0	1	0	10	11	01
214	175	0000000	0A800488	000000000	0D54EAC30D	0	1	0	0	1	01	10	11
215	176	0000000	15000911	000000000	1AA9D5861A	0	0	0	1	0	01	01	10
216	177	0000000	2A001223	000000000	3553AB0C35	0	0	0	0	1	10	01	01
217	178	0000000	54002446	000000000	6AA756186A	0	0	0	1	1	00	10	01
218	179	0000000	2800488D	000000000	554EAC30D5	0	0	0	0	0	01	00	10
219	180	0000000	5000911B	000000000	2A9D5861AA	0	0	0	1	0	00	01	00
220	181	0000000	20012236	000000000	553AB0C355	0	0	0	0	0	11	00	01
221	182	0000000	4002446C	000000000	2A756186AA	0	0	0	0	1	10	11	00
222	183	0000000	000488D9	000000000	54EAC30D54	0	0	0	1	1	01	10	11
223	184	0000000	000911B2	000000000	29D5861AA8	0	0	0	1	0	01	01	10
224	185	0000000	00122364	000000000	53AB0C3550	0	0	0	1	0	11	01	01
225	186	0000000	002446C8	000000000	2756186AA0	0	0	0	0	1	01	11	01
226	187	0000000	00488D90	000000000	4EAC30D540	0	0	0	1	0	10	01	11
227	188	0000000	00911B20	000000000	1D5861AA81	0	1	0	0	1	00	10	01
228	189	0000000	01223640	000000000	3AB0C35502	0	0	0	1	1	01	00	10
229	190	0000000	02446C80	000000000	756186AA05	0	0	0	0	1	01	01	00
230	191	0000000	0488D901	000000000	6AC30D540B	0	1	0	1	1	11	01	01
231	192	0000000	0911B203	000000000	55861AA817	0	0	0	1	0	10	11	01

```
232   193    0000000  12236407  000000000  2B0C35502F      0  0  0  0    0  01     10   11
233   194    0000000  2446C80E  000000000  56186AA05F      0  0  0  0    1  00     01   10
234   195    0000000  488D901C  000000000  2C30D540BF      0  1  0  0    1  11     00   01
235   196    0000000  111B2039  000000000  5861AA817E      0  0  0  0    1  10     11   00
236   197    0000000  22364072  000000000  30C35502FD      0  0  0  1    1  01     10   11
237   198    0000000  446C80E4  000000000  6186AA05FB      0  0  0  1    0  01     01   10
238   199    0000000  08D901C8  000000000  430D540BF6      0  1  0  0    0  11     01   01
239   200    0000000  11B20391  000000000  061AA817EC      0  1  0  0    0  10     11   01

Z[0]  = 3D
Z[1]  = C1
Z[2]  = F0
Z[3]  = BB
Z[4]  = 58
Z[5]  = 1E
Z[6]  = 42
Z[7]  = 42
Z[8]  = 4B
Z[9]  = 8E
Z[10] = C1
Z[11] = 2A
Z[12] = 40
Z[13] = 63
Z[14] = 7A
Z[15] = 1E

================================================================================================
         Reload this pattern into the LFSRs
         Hold content of Summation Combiner regs and calculate new C[t+1] and Z values
================================================================================================
LFSR1  <= 04B583D
LFSR2  <= 208E1EC1
LFSR3  <= 063C142F0
LFSR4  <= 0F7A2A42BB
C[t+1] <= 10

================================================================================================
         Generating 125 key symbols (encryption/decryption sequence)
================================================================================================
240     1    04B583D  208E1EC1  063C142F0  0F7A2A42BB      0  1  0  0    0  10     11   01
241     2    096B07A  411C3D82  0C78285E1  1EF4548577      1  0  1  1    1  10     10   11
242     3    12D60F4  02387B04  18F050BC3  3DE8A90AEF      0  0  1  1    0  01     10   10
243     4    05AC1E9  0470F609  11E0A1786  7BD15215DF      0  0  0  1    0  01     01   10
244     5    0B583D2  08E1EC13  03C142F0C  77A2A42BBF      1  1  0  1    0  00     01   01
245     6    16B07A5  11C3D827  078285E18  6F4548577E      0  1  0  0    1  11     00   01
246     7    0D60F4B  2387B04F  0F050BC30  5E8A90AEFD      1  1  1  1    1  00     11   00
247     8    1AC1E97  470F609E  1E0A17860  3D15215DFA      1  0  1  0    0  11     00   11
248     9    1583D2E  0E1EC13D  1C142F0C0  7A2A42BBF4      0  0  1  0    0  01     11   00
249    10    0B07A5D  1C3D827B  18285E181  74548577E9      1  0  1  0    1  10     01   11
250    11    160F4BB  387B04F7  1050BC302  68A90AEFD2      0  0  0  1    1  00     10   01
251    12    0C1E976  70F609EE  00A178605  515215DFA5      1  1  0  0    0  00     00   10
252    13    183D2ED  61EC13DD  0142F0C0B  22A42BBF4B      1  1  0  1    1  01     00   00
253    14    107A5DA  43D827BA  0285E1817  4548577E97      0  1  0  0    0  00     01   00
254    15    00F4BB4  07B04F74  050BC302F  0A90AEFD2E      0  1  0  1    0  10     00   01
255    16    01E9769  0F609EE8  0A178605E  15215DFA5C      0  0  1  0    1  11     10   00
256    17    03D2ED3  1EC13DD0  142F0C0BD  2A42BBF4B9      0  1  0  0    0  00     11   10
257    18    07A5DA7  3D827BA0  085E1817B  548577E972      0  1  1  1    1  11     00   11
```

258	19	0F4BB4F	7B04F740	10BC302F6	290AEFD2E5	1	0	0	0	0	01	11	00
259	20	1E9769F	7609EE80	0178605ED	5215DFA5CA	1	0	0	0	0	10	01	11
260	21	1D2ED3F	6C13DD01	02F0C0BDA	242BBF4B94	1	0	0	0	1	00	10	01
261	22	1A5DA7E	5827BA03	05E1817B4	48577E9729	1	0	0	0	1	01	00	10
262	23	14BB4FC	304F7407	0BC302F69	10AEFD2E53	0	0	1	1	1	00	01	00
263	24	09769F9	609EE80E	178605ED2	215DFA5CA7	1	1	0	0	0	10	00	01
264	25	12ED3F2	413DD01C	0F0C0BDA4	42BBF4B94F	0	0	1	1	0	00	10	00
265	26	05DA7E5	027BA038	1E1817B49	0577E9729F	0	0	1	0	1	01	00	10
266	27	0BB4FCA	04F74071	1C302F693	0AEFD2E53F	1	1	1	1	1	11	01	00
267	28	1769F95	09EE80E3	18605ED27	15DFA5CA7F	0	1	1	1	0	11	11	01
268	29	0ED3F2B	13DD01C6	10C0BDA4F	2BBF4B94FE	1	1	0	1	0	10	11	11
269	30	1DA7E56	27BA038D	01817B49F	577E9729FD	1	1	0	0	0	10	10	11
270	31	1B4FCAD	4F74071B	0302F693E	2EFD2E53FB	1	0	0	1	0	01	10	10
271	32	169F95B	1EE80E37	0605ED27D	5DFA5CA7F7	0	1	0	1	1	01	01	10
272	33	0D3F2B7	3DD01C6E	0C0BDA4FB	3BF4B94FEF	1	1	1	1	1	00	01	01
273	34	1A7E56F	7BA038DC	1817B49F6	77E9729FDE	1	1	1	1	0	01	00	01
274	35	14FCADF	774071B9	102F693ED	6FD2E53FBD	0	0	0	1	0	00	01	00
275	36	09F95BE	6E80E373	005ED27DB	5FA5CA7F7B	1	1	0	1	1	10	00	01
276	37	13F2B7C	5D01C6E7	00BDA4FB6	3F4B94FEF7	0	0	0	0	0	11	10	00
277	38	07E56F9	3A038DCE	017B49F6C	7E9729FDEE	0	0	0	1	0	00	11	10
278	39	0FCADF2	74071B9C	02F693ED8	7D2E53FBDD	1	0	0	0	1	10	00	11
279	40	1F95BE5	680E3738	05ED27DB0	7A5CA7F7BA	1	0	0	0	1	11	10	00
280	41	1F2B7CA	501C6E71	0BDA4FB60	74B94FEF74	1	0	1	1	0	01	11	10
281	42	1E56F94	2038DCE2	17B49F6C0	69729FDEE8	1	0	0	0	0	10	01	11
282	43	1CADF29	4071B9C4	0F693ED80	52E53FBDD1	1	0	1	1	1	11	10	01
283	44	195BE53	00E37389	1ED27DB01	25CA7F7BA3	1	1	1	1	1	01	11	10
284	45	12B7CA6	01C6E713	1DA4FB602	4B94FEF747	0	1	1	1	0	01	01	11
285	46	056F94C	038DCE26	1B49F6C04	1729FDEE8E	0	1	1	0	1	11	01	01
286	47	0ADF299	071B9C4D	1693ED808	2E53FBDD1C	1	0	0	0	0	10	11	01
287	48	15BE532	0E37389A	0D27DB011	5CA7F7BA38	0	0	1	1	0	10	10	11
288	49	0B7CA64	1C6E7135	1A4FB6022	394FEF7471	1	0	1	0	0	01	10	10
289	50	16F94C9	38DCE26A	149F6C044	729FDEE8E2	0	1	0	1	1	01	01	10
290	51	0DF2993	71B9C4D4	093ED8089	653FBDD1C4	1	1	1	0	0	00	01	01
291	52	1BE5327	637389A9	127DB0112	4A7F7BA388	1	0	0	0	1	11	00	01
292	53	17CA64E	46E71353	04FB60224	14FEF74710	0	1	0	1	1	01	11	00
293	54	0F94C9C	0DCE26A6	09F6C0448	29FDEE8E21	1	1	1	1	1	01	01	11
294	55	1F29939	1B9C4D4D	13ED80890	53FBDD1C42	1	1	0	1	0	00	01	01
295	56	1E53272	37389A9A	07DB01121	27F7BA3884	1	0	0	1	0	10	00	01
296	57	1CA64E5	6E713534	0FB602242	4FEF747108	1	0	1	1	1	00	10	00
297	58	194C9CB	5CE26A69	1F6C04485	1FDEE8E210	1	1	1	1	0	11	00	10
298	59	1299397	39C4D4D3	1ED80890A	3FBDD1C420	0	1	1	1	0	00	11	00
299	60	053272F	7389A9A6	1DB011214	7F7BA38840	0	1	1	0	0	11	00	11
300	61	0A64E5E	6713534C	1B6022428	7EF7471081	1	0	1	1	0	00	11	00
301	62	14C9CBD	4E26A699	16C044850	7DEE8E2102	0	0	0	1	1	10	00	11
302	63	099397A	1C4D4D32	0D80890A0	7BDD1C4205	1	0	1	1	1	00	10	00
303	64	13272F4	389A9A65	1B0112141	77BA38840B	0	1	1	1	1	00	00	10
304	65	064E5E8	713534CB	160224283	6F74710817	0	0	0	0	0	00	00	00
305	66	0C9CBD1	626A6997	0C0448507	5EE8E2102E	1	0	1	1	1	01	00	00
306	67	19397A3	44D4D32E	180890A0E	3DD1C4205C	1	1	1	1	1	11	01	00
307	68	1272F46	09A9A65D	10112141D	7BA38840B8	0	1	0	1	1	10	11	01
308	69	04E5E8C	13534CBA	00224283A	7747108171	0	0	0	0	0	01	10	11
309	70	09CBD19	26A69975	004485075	6E8E2102E3	1	1	0	1	0	10	01	10
310	71	1397A32	4D4D32EB	00890A0EA	5D1C4205C7	0	0	0	0	0	00	10	01
311	72	072F465	1A9A65D7	0112141D5	3A38840B8F	0	1	0	0	1	01	00	10
312	73	0E5E8CA	3534CBAF	0224283AA	747108171F	1	0	0	0	0	00	01	00
313	74	1CBD194	6A69975E	044850755	68E2102E3E	1	0	0	1	0	10	00	01
314	75	197A329	54D32EBC	0890A0EAB	51C4205C7D	1	1	1	1	0	01	10	00

315	76	12F4653	29A65D79	112141D56	238840B8FA	0	1	0	1	1	01	01	10
316	77	05E8CA6	534CBAF2	024283AAD	47108171F4	0	0	0	0	1	10	01	01
317	78	0BD194D	269975E5	04850755B	0E2102E3E9	1	1	0	0	0	11	10	01
318	79	17A329A	4D32EBCB	090A0EAB6	1C4205C7D2	0	0	1	0	0	00	11	10
319	80	0F46535	1A65D797	12141D56D	38840B8FA5	1	0	0	1	0	11	00	11
320	81	1E8CA6A	34CBAF2F	04283AADA	7108171F4B	1	1	0	0	1	01	11	00
321	82	1D194D5	69975E5F	0850755B4	62102E3E97	1	1	1	0	0	01	01	11
322	83	1A329AA	532EBCBF	10A0EAB68	44205C7D2F	1	0	0	0	0	11	01	01
323	84	1465355	265D797F	0141D56D1	0840B8FA5E	0	0	0	0	1	01	11	01
324	85	08CA6AB	4CBAF2FF	0283AADA2	108171F4BC	1	1	0	1	0	01	01	11
325	86	1194D56	1975E5FF	050755B45	2102E3E979	0	0	0	0	1	10	01	01
326	87	0329AAD	32EBCBFF	0A0EAB68A	4205C7D2F3	0	1	1	0	0	11	10	01
327	88	065355A	65D797FF	141D56D14	040B8FA5E7	0	1	0	0	0	00	11	10
328	89	0CA6AB4	4BAF2FFF	083AADA28	08171F4BCF	1	1	1	0	1	11	00	11
329	90	194D569	175E5FFF	10755B450	102E3E979E	1	0	0	0	0	01	11	00
330	91	129AAD3	2EBCBFFF	00EAB68A1	205C7D2F3C	0	1	0	0	0	10	01	11
331	92	05355A6	5D797FFF	01D56D142	40B8FA5E78	0	0	0	1	1	00	10	01
332	93	0A6AB4D	3AF2FFFE	03AADA285	0171F4BCF1	1	1	0	0	0	00	00	10
333	94	14D569B	75E5FFFD	0755B450A	02E3E979E2	0	1	0	1	0	01	00	00
334	95	09AAD37	6BCBFFFA	0EAB68A15	05C7D2F3C4	1	1	1	1	1	11	01	00
335	96	1355A6E	5797FFF4	1D56D142A	0B8FA5E788	0	1	1	1	0	11	11	01
336	97	06AB4DC	2F2FFFE8	1AADA2854	171F4BCF11	0	0	1	0	0	11	11	11
337	98	0D569B8	5E5FFFD0	155B450A9	2E3E979E23	1	0	0	0	0	11	11	11
338	99	1AAD370	3CBFFFA1	0AB68A153	5C7D2F3C46	1	1	1	0	0	10	11	11
339	100	155A6E0	797FFF43	156D142A7	38FA5E788D	0	0	0	1	1	01	10	11
340	101	0AB4DC0	72FFFE87	0ADA2854E	71F4BCF11B	1	1	1	1	1	10	01	10
341	102	1569B81	65FFFD0E	15B450A9D	63E979E236	0	1	0	1	0	11	10	01
342	103	0AD3703	4BFFFA1C	0B68A153B	47D2F3C46C	1	1	1	1	1	01	11	10
343	104	15A6E07	17FFF438	16D142A76	0FA5E788D8	0	1	0	1	1	10	01	11
344	105	0B4DC0F	2FFFE870	0DA2854EC	1F4BCF11B0	1	1	1	0	1	11	10	01
345	106	169B81F	5FFFD0E1	1B450A9D8	3E979E2360	0	1	1	1	0	01	11	10
346	107	0D3703F	3FFFA1C3	168A153B0	7D2F3C46C1	1	1	0	0	1	10	01	11
347	108	1A6E07E	7FFF4386	0D142A761	7A5E788D83	1	1	1	0	1	11	10	01
348	109	14DC0FD	7FFE870C	1A2854EC2	74BCF11B07	0	1	1	1	0	01	11	10
349	110	09B81FB	7FFD0E19	1450A9D84	6979E2360E	1	1	0	0	1	10	01	11
350	111	13703F6	7FFA1C33	08A153B09	52F3C46C1C	0	1	1	1	1	11	10	01
351	112	06E07EC	7FF43867	1142A7612	25E788D838	0	1	0	1	1	00	11	10
352	113	0DC0FD8	7FE870CF	02854EC25	4BCF11B071	1	1	0	1	1	11	00	11
353	114	1B81FB1	7FD0E19E	050A9D84B	179E2360E3	1	1	0	1	0	00	11	00
354	115	1703F62	7FA1C33D	0A153B096	2F3C46C1C7	0	1	1	0	0	11	00	11
355	116	0E07EC4	7F43867B	142A7612C	5E788D838E	1	0	0	0	0	01	11	00
356	117	1C0FD88	7E870CF6	0854EC259	3CF11B071C	1	1	1	1	1	01	01	11
357	118	181FB11	7D0E19ED	10A9D84B3	79E2360E38	1	0	0	1	1	11	01	01
358	119	103F622	7A1C33DA	0153B0967	73C46C1C71	0	0	0	1	0	10	11	01
359	120	007EC45	743867B5	02A7612CE	6788D838E3	0	0	0	1	1	01	10	11
360	121	00FD88B	6870CF6B	054EC259C	4F11B071C6	0	0	0	0	1	00	01	10
361	122	01FB117	50E19ED7	0A9D84B38	1E2360E38C	0	1	1	0	0	10	00	01
362	123	03F622F	21C33DAE	153B09671	3C46C1C718	0	1	0	0	1	11	10	00
363	124	07EC45F	43867B5C	0A7612CE2	788D838E30	0	1	1	1	0	01	11	10
364	125	0FD88BF	070CF6B9	14EC259C4	711B071C61	1	0	0	0	0	10	01	11

1.3 SECOND SET OF SAMPLE DATA

Initial values for the key, BD_ADDR and clock

K'c2[0] = 00 K'c2[1] = 00 K'c2[2] = 00 K'c2[3] = 00
K'c2[4] = 00 K'c2[5] = 00 K'c2[6] = 00 K'c2[7] = 00
K'c2[8] = 00 K'c2[9] = 00 K'c2[10] = 00 K'c2[11] = 00
K'c2[12] = 00 K'c2[13] = 00 K'c2[14] = 00 K'c2[15] = 00

Addr2[0] = 00 Addr2[1] = 00 Addr2[2] = 00
Addr2[3] = 00 Addr2[4] = 00 Addr2[5] = 00

CL2[0] = 00 CL2[1] = 00 CL2[2] = 00 CL2[3] = 03

Fill LFSRs with initial data

t	clk#	LFSR1	LFSR2	LFSR3	LFSR4	X1	X2	X3	X4	Z	C[t+1]	C[t]	C[t-1]
0	0	0000000*	00000000*	000000000*	0000000000*	0	0	0	0	0	00	00	00
1	1	0000001*	00000001*	000000001*	0000000001*	0	0	0	0	0	00	00	00
2	2	0000002*	00000002*	000000002*	0000000003*	0	0	0	0	0	00	00	00
3	3	0000004*	00000004*	000000004*	0000000007*	0	0	0	0	0	00	00	00
4	4	0000008*	00000008*	000000008*	000000000E*	0	0	0	0	0	00	00	00
5	5	0000010*	00000010*	000000010*	000000001C*	0	0	0	0	0	00	00	00
6	6	0000020*	00000020*	000000020*	0000000038*	0	0	0	0	0	00	00	00
7	7	0000040*	00000040*	000000040*	0000000070*	0	0	0	0	0	00	00	00
8	8	0000080*	00000080*	000000080*	00000000E0*	0	0	0	0	0	00	00	00
9	9	0000100*	00000100*	000000100*	00000001C0*	0	0	0	0	0	00	00	00
10	10	0000200*	00000200*	000000200*	0000000380*	0	0	0	0	0	00	00	00
11	11	0000400*	00000400*	000000400*	0000000700*	0	0	0	0	0	00	00	00
12	12	0000800*	00000800*	000000800*	0000000E00*	0	0	0	0	0	00	00	00
13	13	0001000*	00001000*	000001000*	0000001C00*	0	0	0	0	0	00	00	00
14	14	0002000*	00002000*	000002000*	0000003800*	0	0	0	0	0	00	00	00
15	15	0004000*	00004000*	000004000*	0000007000*	0	0	0	0	0	00	00	00
16	16	0008000*	00008000*	000008000*	000000E000*	0	0	0	0	0	00	00	00
17	17	0010000*	00010000*	000010000*	000001C000*	0	0	0	0	0	00	00	00
18	18	0020000*	00020000*	000020000*	0000038000*	0	0	0	0	0	00	00	00
19	19	0040000*	00040000*	000040000*	0000070000*	0	0	0	0	0	00	00	00
20	20	0080000*	00080000*	000080000*	00000E0000*	0	0	0	0	0	00	00	00
21	21	0100000*	00100000*	000100000*	00001C0000*	0	0	0	0	0	00	00	00
22	22	0200000*	00200000*	000200000*	0000380000*	0	0	0	0	0	00	00	00
23	23	0400000*	00400000*	000400000*	0000700000*	0	0	0	0	0	00	00	00
24	24	0800000*	00800000*	000800000*	0000E00000*	1	1	0	0	0	01	00	00
25	25	1000000*	01000000*	001000000*	0001C00000*	0	0	0	0	0	00	00	00
26	26	0000001	02000000*	002000000*	0003800000*	0	0	0	0	0	00	00	00
27	27	0000002	04000000*	004000000*	0007000000*	0	0	0	0	0	00	00	00
28	28	0000004	08000000*	008000000*	000E000000*	0	0	0	0	0	00	00	00
29	29	0000008	10000000*	010000000*	001C000000*	0	0	0	0	0	00	00	00
30	30	0000010	20000000*	020000000*	0038000000*	0	0	0	0	0	00	00	00
31	31	0000020	40000000*	040000000*	0070000000*	0	0	0	0	0	00	00	00
32	32	0000040	00000001	080000000*	00E0000000*	0	0	1	1	0	01	00	00
33	33	0000080	00000002	100000000*	01C0000000*	0	0	0	1	1	00	00	00
34	34	0000101	00000004	000000001	0380000000*	0	0	0	1	1	00	00	00

35	35	0000202	00000008	000000002	0700000000*	0	0	0	0	0	00	00	00
36	36	0000404	00000010	000000004	0E00000000*	0	0	0	0	0	00	00	00
37	37	0000808	00000020	000000008	1C00000000*	0	0	0	0	0	00	00	00
38	38	0001011	00000040	000000011	3800000000*	0	0	0	0	0	00	00	00
39	39	0002022	00000080	000000022	7000000000*	0	0	0	0	0	00	00	00

Start clocking Summation Combiner

40	1	0004044	00000100	000000044	6000000001	0	0	0	0	0	00	00	00
41	2	0008088	00000200	000000088	4000000003	0	0	0	0	0	00	00	00
42	3	0010111	00000400	000000111	0000000007	0	0	0	0	0	00	00	00
43	4	0020222	00000800	000000222	000000000E	0	0	0	0	0	00	00	00
44	5	0040444	00001001	000000444	000000001D	0	0	0	0	0	00	00	00
45	6	0080888	00002002	000000888	000000003B	0	0	0	0	0	00	00	00
46	7	0101111	00004004	000001111	0000000077	0	0	0	0	0	00	00	00
47	8	0202222	00008008	000002222	00000000EE	0	0	0	0	0	00	00	00
48	9	0404444	00010011	000004444	00000001DD	0	0	0	0	0	00	00	00
49	10	0808888	00020022	000008888	00000003BB	1	0	0	0	1	00	00	00
50	11	1011110	00040044	000011111	0000000777	0	0	0	0	0	00	00	00
51	12	0022221	00080088	000022222	0000000EEE	0	0	0	0	0	00	00	00
52	13	0044442	00100110	000044444	0000001DDD	0	0	0	0	0	00	00	00
53	14	0088884	00200220	000088888	0000003BBB	0	0	0	0	0	00	00	00
54	15	0111109	00400440	000111111	0000007777	0	0	0	0	0	00	00	00
55	16	0222212	00800880	000222222	000000EEEE	0	1	0	0	1	00	00	00
56	17	0444424	01001100	000444444	000001DDDD	0	0	0	0	0	00	00	00
57	18	0888848	02002200	000888888	000003BBBB	1	0	0	0	1	00	00	00
58	19	1111090	04004400	001111110	0000077777	0	0	0	0	0	00	00	00
59	20	0222120	08008800	002222220	00000EEEEE	0	0	0	0	0	00	00	00
60	21	0444240	10011000	004444440	00001DDDDD	0	0	0	0	0	00	00	00
61	22	0888480	20022000	008888880	00003BBBBB	1	0	0	0	1	00	00	00
62	23	1110900	40044000	011111100	0000777777	0	0	0	0	0	00	00	00
63	24	0221200	00088001	022222200	0000EEEEEE	0	0	0	0	0	00	00	00
64	25	0442400	00110003	044444400	0001DDDDDD	0	0	0	0	0	00	00	00
65	26	0884800	00220006	088888800	0003BBBBBB	1	0	1	0	0	01	00	00
66	27	1109000	0044000C	111111000	0007777777	0	0	0	0	1	01	01	00
67	28	0212001	00880018	022222001	000EEEEEEE	0	1	0	0	0	11	01	01
68	29	0424002	01100031	044444002	001DDDDDDC	0	0	0	0	1	01	11	01
69	30	0848004	02200062	088888004	003BBBBBB8	1	0	1	0	1	10	01	11
70	31	1090008	044000C4	111110008	0077777770	0	0	0	0	0	00	10	01
71	32	0120010	08800188	022220010	00EEEEEEE0	0	1	0	1	0	00	00	10
72	33	0240020	11000311	044440020	01DDDDDDC1	0	0	0	1	1	00	00	00
73	34	0480040	22000622	088880040	03BBBBBB83	0	0	1	1	0	01	00	00
74	35	0900081	44000C44	111100080	0777777707	1	0	0	0	0	00	01	00
75	36	1200103	08001888	022200101	0EEEEEEE0E	0	0	0	1	1	11	00	01
76	37	0400207	10003111	044400202	1DDDDDDC1D	0	0	0	1	0	01	11	00
77	38	080040E	20006222	088800404	3BBBBBB83B	1	0	1	1	0	01	01	11
78	39	100081C	4000C444	111000808	7777777077	0	0	0	0	1	10	01	01
79	40	0001038	00018888	022001010	6EEEEEE0EF	0	0	0	1	1	00	10	01
80	41	0002070	00031110	044002020	5DDDDDC1DE	0	0	0	1	1	01	00	10
81	42	00040E0	00062220	088004040	3BBBBB83BC	0	0	1	1	1	00	01	00
82	43	00081C1	000C4440	110008081	7777770779	0	0	0	0	0	11	00	01
83	44	0010383	00188880	020010103	6EEEEE0EF2	0	0	0	1	0	01	11	00
84	45	0020707	00311100	040020206	5DDDDC1DE5	0	0	0	1	0	10	01	11
85	46	0040E0E	00622200	08004040C	3BBBB83BCB	0	0	1	1	0	11	10	01
86	47	0081C1D	00C44400	100080819	7777707797	0	1	0	0	0	00	11	10
87	48	010383A	01888801	000101032	6EEEE0EF2F	0	1	0	1	0	11	00	11
88	49	0207075	03111003	000202064	5DDDC1DE5E	0	0	0	1	0	01	11	00

89	50	040E0EA	06222006	0004040C8	3BBB83BCBC	0	0	0	1	0	10	01	11
90	51	081C1D5	0C44400C	000808191	7777077979	1	0	0	0	1	00	10	01
91	52	10383AB	18888018	001010323	6EEE0EF2F2	0	1	0	1	0	00	00	10
92	53	0070756	31110030	002020646	5DDC1DE5E5	0	0	0	1	1	00	00	00
93	54	00E0EAC	62220060	004040C8C	3BB83BCBCB	0	0	0	1	1	00	00	00
94	55	01C1D59	444400C1	008081919	7770779797	0	0	0	0	0	00	00	00
95	56	0383AB2	08880183	010103232	6EE0EF2F2F	0	1	0	1	0	01	00	00
96	57	0707565	11100307	020206464	5DC1DE5E5F	0	0	0	1	0	00	01	00
97	58	0E0EACA	2220060E	04040C8C8	3B83BCBCBF	1	0	0	1	0	10	00	01
98	59	1C1D594	44400C1C	080819191	770779797E	1	0	1	0	0	00	10	00
99	60	183AB28	08801838	101032323	6E0EF2F2FC	1	1	0	0	0	00	00	10
100	61	1075650	11003070	002064647	5C1DE5E5F8	0	0	0	0	0	00	00	00
101	62	00EACA1	220060E0	0040C8C8E	383BCBCBF0	0	0	0	0	0	00	00	00
102	63	01D5943	4400C1C0	00819191D	70779797E0	0	0	0	0	0	00	00	00
103	64	03AB286	08018380	01032323A	60EF2F2FC1	0	0	0	1	1	00	00	00
104	65	075650C	10030701	020646475	41DE5E5F82	0	0	0	1	1	00	00	00
105	66	0EACA18	20060E02	040C8C8EA	03BCBCBF04	1	0	0	1	0	01	00	00
106	67	1D59430	400C1C05	0819191D4	0779797E09	1	0	1	0	1	00	01	00
107	68	1AB2861	0018380A	1032323A9	0EF2F2FC12	1	0	0	1	0	10	00	01
108	69	15650C3	00307015	006464752	1DE5E5F825	0	0	0	1	1	11	10	00
109	70	0ACA186	0060E02A	00C8C8EA4	3BCBCBF04B	1	0	0	1	1	00	11	10
110	71	159430C	00C1C055	019191D48	779797E097	0	1	0	1	0	11	00	11
111	72	0B28618	018380AA	032323A90	6F2F2FC12F	1	1	0	0	1	01	11	00
112	73	1650C30	03070154	064647520	5E5E5F825E	0	0	0	0	1	11	01	11
113	74	0CA1860	060E02A8	0C8C8EA40	3CBCBF04BC	1	0	1	1	0	11	11	01
114	75	19430C0	0C1C0550	19191D480	79797E0979	1	0	1	0	1	11	11	11
115	76	1286180	18380AA0	12323A900	72F2FC12F2	0	0	0	1	0	11	11	11
116	77	050C301	30701541	046475201	65E5F825E5	0	0	0	1	0	11	11	11
117	78	0A18602	60E02A82	08C8EA402	4BCBF04BCB	1	1	1	1	1	10	11	11
118	79	1430C04	41C05505	1191D4804	1797E09796	0	1	0	1	0	10	10	11
119	80	0861808	0380AA0A	0323A9008	2F2FC12F2C	1	1	0	0	0	01	10	10
120	81	10C3011	07015415	064752011	5E5F825E59	0	0	0	0	1	00	01	10
121	82	0186022	0E02A82A	0C8EA4022	3CBF04BCB2	0	0	1	1	0	10	00	01
122	83	030C045	1C055054	191D48044	797E097964	0	0	1	0	1	11	10	00
123	84	061808A	380AA0A8	123A90088	72FC12F2C9	0	0	0	1	0	00	11	10
124	85	0C30115	70154151	047520111	65F825E593	1	0	0	1	0	11	00	11
125	86	186022A	602A82A3	08EA40222	4BF04BCB26	1	0	1	1	0	00	11	00
126	87	10C0455	40550546	11D480444	17E097964C	0	0	0	1	1	10	00	11
127	88	01808AA	00AA0A8D	03A900888	2FC12F2C99	0	1	0	1	0	00	10	00
128	89	0301155	0154151A	075201111	5F825E5932	0	0	0	1	1	01	00	10
129	90	06022AA	02A82A34	0EA402222	3F04BCB264	0	1	1	0	1	00	01	00
130	91	0C04555	05505468	1D4804445	7E097964C9	1	0	1	0	0	10	00	01
131	92	1808AAA	0AA0A8D0	1A900888A	7C12F2C992	1	1	1	0	1	00	10	00
132	93	1011555	154151A1	152011115	7825E59324	0	0	0	0	0	01	00	10
133	94	0022AAB	2A82A342	0A402222B	704BCB2648	0	1	1	0	1	00	01	00
134	95	0045556	55054684	148044457	6097964C91	0	0	0	1	1	11	00	01
135	96	008AAAC	2A0A8D09	0900888AE	412F2C9923	0	0	1	0	0	01	11	00
136	97	0115559	54151A12	12011115D	025E593246	0	0	0	0	1	11	01	11
137	98	022AAB2	282A3424	0402222BA	04BCB2648D	0	0	0	1	0	10	11	01
138	99	0455564	50546848	080444575	097964C91A	0	0	1	0	1	01	10	11
139	100	08AAAC8	20A8D090	100888AEA	12F2C99235	1	1	0	1	0	10	01	10
140	101	1155591	4151A120	0011115D5	25E593246A	0	0	0	1	1	00	10	01
141	102	02AAB22	02A34240	002222BAA	4BCB2648D5	0	1	0	1	0	00	00	10
142	103	0555644	05468481	004445755	17964C91AB	0	0	0	1	1	00	00	00
143	104	0AAAC88	0A8D0903	00888AEAA	2F2C992357	1	1	0	0	0	01	00	00
144	105	1555911	151A1206	011115D55	5E593246AE	0	0	0	0	1	01	01	00
145	106	0AAB222	2A34240C	02222BAAA	3CB2648D5C	1	0	0	1	1	11	01	01

146	107	1556445	54684818	044457555	7964C91AB8	0	0	0	0	1	01	11	01
147	108	0AAC88B	28D09030	0888AEAAA	72C9923571	1	1	1	1	1	01	01	11
148	109	1559117	51A12060	11115D555	6593246AE2	0	1	0	1	1	11	01	01
149	110	0AB222F	234240C0	0222BAAAB	4B2648D5C5	1	0	0	0	0	10	11	01
150	111	156445F	46848180	044575557	164C91AB8A	0	1	0	0	1	01	10	11
151	112	0AC88BF	0D090301	088AEAAAE	2C99235714	1	0	1	1	0	10	01	10
152	113	159117F	1A120602	1115D555D	593246AE28	0	0	0	0	0	00	10	01
153	114	0B222FE	34240C04	022BAAABA	32648D5C51	1	0	0	0	1	01	00	10
154	115	16445FD	68481809	045755574	64C91AB8A2	0	0	0	1	0	00	01	00
155	116	0C88BFA	50903012	08AEAAAE8	4992357144	1	1	1	1	0	01	00	01
156	117	19117F5	21206024	115D555D1	13246AE288	1	0	0	0	0	00	01	00
157	118	1222FEA	4240C048	02BAAABA2	2648D5C511	0	0	0	0	0	11	00	01
158	119	0445FD5	04818090	057555744	4C91AB8A23	0	1	0	1	1	01	11	00
159	120	088BFAA	09030120	0AEAAAE88	1923571446	1	0	1	0	1	10	01	11
160	121	1117F55	12060240	15D555D11	3246AE288D	0	0	0	0	0	00	10	01
161	122	022FEAA	240C0480	0BAAABA22	648D5C511B	0	0	1	1	0	00	00	10
162	123	045FD54	48180900	175557444	491AB8A237	0	0	0	0	0	00	00	00
163	124	08BFAA9	10301200	0EAAAE889	123571446F	1	0	1	0	0	01	00	00
164	125	117F553	20602400	1D555D113	246AE288DF	0	0	1	0	0	00	01	00
165	126	02FEAA7	40C04800	1AAABA227	48D5C511BE	0	1	1	1	1	10	00	01
166	127	05FD54F	01809001	15557444F	11AB8A237D	0	1	0	1	0	00	10	00
167	128	0BFAA9F	03012002	0AAAE889E	23571446FA	1	0	1	0	0	00	00	10
168	129	17F553F	06024004	1555D113D	46AE288DF5	0	0	0	1	1	00	00	00
169	130	0FEAA7E	0C048008	0AABA227A	0D5C511BEA	1	0	1	0	0	01	00	00
170	131	1FD54FC	18090011	1557444F5	1AB8A237D5	1	0	0	1	1	00	01	00
171	132	1FAA9F9	30120022	0AAE889EB	3571446FAA	1	0	1	0	0	10	00	01
172	133	1F553F2	60240044	155D113D7	6AE288DF55	1	0	0	1	0	00	10	00
173	134	1EAA7E4	40480089	0ABA227AE	55C511BEAA	1	0	1	1	1	00	00	10
174	135	1D54FC9	00900113	157444F5D	2B8A237D54	1	1	0	1	1	01	00	00
175	136	1AA9F93	01200227	0AE889EBA	571446FAA8	1	0	1	0	1	00	01	00
176	137	1553F26	0240044E	15D113D75	2E288DF550	0	0	0	0	0	11	00	01
177	138	0AA7E4C	0480089C	0BA227AEA	5C511BEAA0	1	1	1	0	0	00	11	00
178	139	154FC98	09001138	17444F5D4	38A237D540	0	0	0	1	1	10	00	11
179	140	0A9F931	12002270	0E889EBA9	71446FAA81	1	0	1	0	0	00	10	00
180	141	153F262	240044E0	1D113D753	6288DF5503	0	0	1	1	0	00	00	10
181	142	0A7E4C5	480089C0	1A227AEA7	4511BEAA06	1	0	1	0	0	01	00	00
182	143	14FC98B	10011381	1444F5D4F	0A237D540D	0	0	0	0	1	01	01	00
183	144	09F9316	20022702	0889EBA9E	1446FAA81A	1	0	1	0	1	11	01	01
184	145	13F262D	40044E04	1113D753D	288DF55035	0	0	0	1	0	10	11	01
185	146	07E4C5A	00089C08	0227AEA7A	511BEAA06A	0	0	0	0	0	01	10	11
186	147	0FC98B4	00113810	044F5D4F5	2237D540D5	1	0	0	0	0	01	01	10
187	148	1F93169	00227021	089EBA9EB	446FAA81AA	1	0	1	0	1	11	01	01
188	149	1F262D2	0044E042	113D753D7	08DF550355	1	0	0	1	1	10	11	01
189	150	1E4C5A4	0089C085	027AEA7AE	11BEAA06AA	1	1	0	1	1	10	10	11
190	151	1C98B48	0113810A	04F5D4F5C	237D540D54	1	0	0	0	1	10	10	10
191	152	1931691	02270215	09EBA9EB8	46FAA81AA9	1	0	1	1	1	01	10	10
192	153	1262D22	044E042A	13D753D71	0DF5503553	0	0	0	1	0	01	01	10
193	154	04C5A44	089C0854	07AEA7AE2	1BEAA06AA7	0	1	0	1	1	11	01	01
194	155	098B488	113810A8	0F5D4F5C4	37D540D54E	1	0	1	1	0	11	11	01
195	156	1316910	22702150	1EBA9EB89	6FAA81AA9D	0	0	1	1	1	11	11	11
196	157	062D220	44E042A0	1D753D712	5F5503553A	0	1	1	0	1	11	11	11
197	158	0C5A440	09C08540	1AEA7AE25	3EAA06AA75	1	1	1	1	1	10	11	11
198	159	18B4880	13810A80	15D4F5C4B	7D540D54EA	1	1	0	0	0	10	10	11
199	160	1169100	27021500	0BA9EB897	7AA81AA9D5	0	0	1	1	0	01	10	10
200	161	02D2201	4E042A00	1753D712E	75503553AB	0	0	0	0	1	00	01	10
201	162	05A4403	1C085400	0EA7AE25C	6AA06AA756	0	0	1	1	0	10	00	01
202	163	0B48807	3810A800	1D4F5C4B8	5540D54EAC	1	0	1	0	0	00	10	00

203	164	169100F	70215000	1A9EB8971	2A81AA9D58	0	0	1	1	0	00	00	10
204	165	0D2201E	6042A001	153D712E3	5503553AB0	1	0	0	0	1	00	00	00
205	166	1A4403C	40854002	0A7AE25C6	2A06AA7561	1	1	1	0	1	01	00	00
206	167	1488079	010A8004	14F5C4B8D	540D54EAC3	0	0	0	0	1	01	01	00
207	168	09100F2	02150009	09EB8971B	281AA9D586	1	0	1	0	1	11	01	01
208	169	12201E5	042A0012	13D712E37	503553AB0C	0	0	0	0	1	01	11	01
209	170	04403CA	08540024	07AE25C6E	206AA75618	0	0	0	0	1	11	01	11
210	171	0880795	10A80048	0F5C4B8DD	40D54EAC30	1	1	1	1	1	11	11	01
211	172	1100F2A	21500091	1EB8971BA	01AA9D5861	0	0	1	1	1	11	11	11
212	173	0201E54	42A00122	1D712E374	03553AB0C3	0	1	1	0	1	11	11	11
213	174	0403CA9	05400244	1AE25C6E9	06AA756186	0	0	1	1	1	11	11	11
214	175	0807952	0A800488	15C4B8DD3	0D54EAC30D	1	1	0	0	1	11	11	11
215	176	100F2A5	15000911	0B8971BA6	1AA9D5861A	0	0	1	1	1	11	11	11
216	177	001E54A	2A001223	1712E374C	3553AB0C35	0	0	0	0	1	00	11	11
217	178	003CA94	54002446	0E25C6E98	6AA756186A	0	0	1	1	0	11	00	11
218	179	0079528	2800488D	1C4B8DD31	554EAC30D5	0	0	1	0	0	01	11	00
219	180	00F2A50	5000911B	18971BA62	2A9D5861AA	0	0	1	1	1	10	01	11
220	181	01E54A0	20012236	112E374C4	553AB0C355	0	0	0	0	0	00	10	01
221	182	03CA940	4002446C	025C6E988	2A756186AA	0	0	0	0	0	01	00	10
222	183	0795280	000488D9	04B8DD310	54EAC30D54	0	0	0	1	0	00	01	00
223	184	0F2A500	000911B2	0971BA620	29D5861AA8	1	0	1	1	1	10	00	01
224	185	1E54A00	00122364	12E374C40	53AB0C3550	1	0	0	1	0	00	10	00
225	186	1CA9400	002446C8	05C6E9880	2756186AA0	1	0	0	0	1	01	00	10
226	187	1952800	00488D90	0B8DD3101	4EAC30D540	1	0	1	1	0	11	01	00
227	188	12A5000	00911B20	171BA6202	1D5861AA81	0	1	0	0	0	10	11	01
228	189	054A000	01223640	0E374C404	3AB0C35502	0	0	1	1	0	10	10	11
229	190	0A94000	02446C80	1C6E98808	756186AA05	1	0	1	0	0	01	10	10
230	191	1528001	0488D901	18DD31011	6AC30D540B	0	1	1	1	0	10	01	10
231	192	0A50003	0911B203	11BA62023	55861AA817	1	0	0	1	0	11	10	01
232	193	14A0006	12236407	0374C4047	2B0C35502F	0	0	0	0	1	11	11	10
233	194	094000C	2446C80E	06E98808E	56186AA05F	1	0	0	0	0	11	11	11
234	195	1280018	488D901C	0DD31011D	2C30D540BF	0	1	1	0	1	11	11	11
235	196	0500030	111B2039	1BA62023A	5861AA817E	0	0	1	0	0	11	11	11
236	197	0A00060	22364072	174C40475	30C35502FD	1	0	0	1	1	11	11	11
237	198	14000C0	446C80E4	0E98808EA	6186AA05FB	0	0	1	1	1	11	11	11
238	199	0800180	08D901C8	1D31011D5	430D540BF6	1	1	1	0	0	10	11	11
239	200	1000301	11B20391	1A62023AB	061AA817EC	0	1	1	0	0	10	10	11

```
Z[0]  = 25
Z[1]  = 45
Z[2]  = 6B
Z[3]  = 55
Z[4]  = 5F
Z[5]  = C2
Z[6]  = 20
Z[7]  = E5
Z[8]  = C4
Z[9]  = F8
Z[10] = 3A
Z[11] = F1
Z[12] = FF
Z[13] = 89
Z[14] = 02
Z[15] = 35

================================================================================
         Reload this pattern into the LFSRs
```

Appendix IV - Sample Data

```
         Hold content of Summation Combiner regs and calculate new C[t+1] and Z values
================================================================================================
LFSR1  <= 1C45F25
LFSR2  <= 7FF8C245
LFSR3  <= 1893A206B
LFSR4  <= 1A02F1E555
C[t+1] <= 10

================================================================================================
         Generating 125 key symbols (encryption/decryption sequence)
================================================================================================
240     1    1C45F25  7FF8C245  1893A206B  1A02F1E555      1  1  1  0    1  10     10   11
241     2    188BE4A  7FF1848B  1127440D7  3405E3CAAB      1  1  0  0    0  01     10   10
242     3    1117C95  7FE30917  024E881AF  680BC79557      0  1  0  0    0  01     01   10
243     4    022F92B  7FC6122F  049D1035E  50178F2AAF      0  1  0  0    0  11     01   01
244     5    045F257  7F8C245E  093A206BD  202F1E555E      0  1  1  0    1  10     11   01
245     6    08BE4AE  7F1848BC  127440D7A  405E3CAABC      1  0  0  0    1  01     10   11
246     7    117C95C  7E309178  04E881AF4  00BC795579      0  0  0  1    0  01     01   10
247     8    02F92B8  7C6122F0  09D1035E8  0178F2AAF2      0  0  1  0    0  11     01   01
248     9    05F2570  78C245E1  13A206BD0  02F1E555E5      0  1  0  1    1  10     11   01
249    10    0BE4AE1  71848BC2  07440D7A0  05E3CAABCA      1  1  0  1    1  10     10   11
250    11    17C95C3  63091784  0E881AF40  0BC7955795      0  0  1  1    0  01     10   10
251    12    0F92B87  46122F09  1D1035E80  178F2AAF2B      1  0  1  1    0  10     01   10
252    13    1F2570F  0C245E12  1A206BD01  2F1E555E56      1  0  1  0    0  11     10   01
253    14    1E4AE1F  1848BC25  1440D7A03  5E3CAABCAC      1  0  0  0    0  00     11   10
254    15    1C95C3E  3091784A  0881AF407  3C79557958      1  1  1  0    1  11     00   11
255    16    192B87D  6122F094  11035E80F  78F2AAF2B1      1  0  0  1    1  01     11   00
256    17    12570FA  4245E128  0206BD01E  71E555E562      0  0  0  1    0  10     01   11
257    18    04AE1F4  048BC250  040D7A03D  63CAABCAC5      0  1  0  1    0  11     10   01
258    19    095C3E8  091784A0  081AF407A  479557958A      1  0  1  1    0  01     11   10
259    20    12B87D1  122F0941  1035E80F4  0F2AAF2B14      0  0  0  0    1  11     01   11
260    21    0570FA3  245E1283  006BD01E9  1E555E5628      0  0  0  0    1  01     11   01
261    22    0AE1F46  48BC2506  00D7A03D2  3CAABCAC50      1  1  0  1    0  01     01   11
262    23    15C3E8C  11784A0C  01AF407A5  79557958A0      0  0  0  0    1  10     01   01
263    24    0B87D18  22F09419  035E80F4A  72AAF2B140      1  1  0  1    1  11     10   01
264    25    170FA30  45E12832  06BD01E94  6555E56280      0  1  0  0    0  00     11   10
265    26    0E1F460  0BC25065  0D7A03D28  4AABCAC501      1  1  1  1    0  00     00   11
266    27    1C3E8C0  1784A0CB  1AF407A50  1557958A03      1  1  1  0    1  01     00   00
267    28    187D181  2F094196  15E80F4A0  2AAF2B1406      1  0  0  1    1  00     01   00
268    29    10FA302  5E12832C  0BD01E941  555E56280C      0  0  1  0    1  11     00   01
269    30    01F4604  3C250658  17A03D283  2ABCAC5019      0  0  0  1    0  01     11   00
270    31    03E8C09  784A0CB0  0F407A506  557958A033      0  0  1  0    0  10     01   11
271    32    07D1812  70941960  1E80F4A0C  2AF2B14066      0  1  1  1    1  11     10   01
272    33    0FA3024  612832C1  1D01E9419  55E56280CD      1  0  1  1    0  01     11   10
273    34    1F46049  42506583  1A03D2832  2BCAC5019A      1  0  1  1    0  01     01   11
274    35    1E8C093  04A0CB07  1407A5065  57958A0335      1  1  0  1    0  00     01   01
275    36    1D18127  0941960F  080F4A0CB  2F2B14066B      1  0  1  0    0  10     00   01
276    37    1A3024F  12832C1F  101E94196  5E56280CD7      1  1  0  0    0  00     10   00
277    38    146049F  2506583E  003D2832C  3CAC5019AE      0  0  0  1    1  01     00   10
278    39    08C093E  4A0CB07D  007A50658  7958A0335D      1  0  0  0    0  00     01   00
279    40    118127C  141960FA  00F4A0CB0  72B14066BA      0  0  0  1    1  11     00   01
280    41    03024F8  2832C1F4  01E941961  656280CD74      0  0  0  0    1  10     11   00
281    42    06049F1  506583E9  03D2832C2  4AC5019AE9      0  0  0  1    1  01     10   11
282    43    0C093E2  20CB07D2  07A506585  158A0335D3      1  1  0  1    0  10     01   10
283    44    18127C5  41960FA5  0F4A0CB0B  2B14066BA7      1  1  1  0    1  11     10   01
284    45    1024F8A  032C1F4B  1E9419616  56280CD74F      0  0  1  0    0  00     11   10
285    46    0049F15  06583E97  1D2832C2C  2C5019AE9F      0  0  1  0    1  10     00   11
```

286	47	0093E2B	0CB07D2F	1A5065859	58A0335D3E	0	1	1	1	1	00	10	00
287	48	0127C56	1960FA5E	14A0CB0B2	314066BA7D	0	0	0	0	0	01	00	10
288	49	024F8AD	32C1F4BC	094196164	6280CD74FB	0	1	1	1	0	11	01	00
289	50	049F15A	6583E978	12832C2C8	45019AE9F6	0	1	0	0	0	10	11	01
290	51	093E2B5	4B07D2F0	050658591	0A0335D3ED	1	0	0	0	1	01	10	11
291	52	127C56B	160FA5E0	0A0CB0B22	14066BA7DA	0	0	1	0	0	01	01	10
292	53	04F8AD7	2C1F4BC1	141961645	280CD74FB5	0	0	0	0	1	10	01	01
293	54	09F15AF	583E9783	0832C2C8A	5019AE9F6A	1	0	1	0	0	11	10	01
294	55	13E2B5E	307D2F06	106585915	20335D3ED5	0	0	0	0	1	11	11	10
295	56	07C56BD	60FA5E0D	00CB0B22B	4066BA7DAA	0	1	0	0	0	11	11	11
296	57	0F8AD7A	41F4BC1B	019616457	00CD74FB54	1	1	0	1	0	10	11	11
297	58	1F15AF4	03E97836	032C2C8AF	019AE9F6A9	1	1	0	1	1	10	10	11
298	59	1E2B5E9	07D2F06C	06585915E	0335D3ED52	1	1	0	0	0	01	10	10
299	60	1C56BD2	0FA5E0D8	0CB0B22BC	066BA7DAA4	1	1	1	0	0	10	01	10
300	61	18AD7A5	1F4BC1B0	196164578	0CD74FB549	1	0	1	1	1	11	10	01
301	62	115AF4B	3E978361	12C2C8AF0	19AE9F6A92	0	1	0	1	1	00	11	10
302	63	02B5E96	7D2F06C2	0585915E0	335D3ED524	0	0	0	0	0	10	00	11
303	64	056BD2D	7A5E0D85	0B0B22BC1	66BA7DAA49	0	0	1	1	0	00	10	00
304	65	0AD7A5B	74BC1B0A	161645783	4D74FB5493	1	1	0	0	0	00	00	10
305	66	15AF4B6	69783615	0C2C8AF07	1AE9F6A926	0	0	1	1	0	01	00	00
306	67	0B5E96D	52F06C2B	185915E0F	35D3ED524C	1	1	1	1	1	11	01	00
307	68	16BD2DB	25E0D857	10B22BC1F	6BA7DAA499	0	1	0	1	1	10	11	01
308	69	0D7A5B7	4BC1B0AF	01645783F	574FB54933	1	1	0	0	0	10	10	11
309	70	1AF4B6F	1783615F	02C8AF07F	2E9F6A9266	1	1	0	1	1	01	10	10
310	71	15E96DF	2F06C2BF	05915E0FF	5D3ED524CC	0	0	0	0	1	00	01	10
311	72	0BD2DBF	5E0D857F	0B22BC1FE	3A7DAA4998	1	0	1	0	0	10	00	01
312	73	17A5B7F	3C1B0AFE	1645783FD	74FB549331	0	0	0	1	1	11	10	00
313	74	0F4B6FF	783615FD	0C8AF07FA	69F6A92662	1	0	1	1	0	01	11	10
314	75	1E96DFF	706C2BFB	1915E0FF5	53ED524CC4	1	0	1	1	0	01	01	11
315	76	1D2DBFE	60D857F6	122BC1FEB	27DAA49988	1	1	0	1	0	00	01	01
316	77	1A5B7FD	41B0AFEC	045783FD7	4FB5493310	1	1	0	1	1	10	00	01
317	78	14B6FFA	03615FD8	08AF07FAE	1F6A926620	0	0	1	0	1	11	10	00
318	79	096DFF4	06C2BFB1	115E0FF5D	3ED524CC40	1	1	0	1	0	01	11	10
319	80	12DBFE8	0D857F63	02BC1FEBB	7DAA499881	0	1	0	1	1	10	01	11
320	81	05B7FD0	1B0AFEC6	05783FD77	7B54933103	0	0	0	0	0	00	10	01
321	82	0B6FFA1	3615FD8C	0AF07FAEF	76A9266206	1	0	1	1	1	00	00	10
322	83	16DFF42	6C2BFB18	15E0FF5DE	6D524CC40C	0	0	0	0	0	00	00	00
323	84	0DBFE85	5857F631	0BC1FEBBD	5AA4998819	1	0	1	1	1	01	00	00
324	85	1B7FD0B	30AFEC62	1783FD77A	3549331033	1	1	0	0	1	00	01	00
325	86	16FFA16	615FD8C5	0F07FAEF5	6A92662067	0	0	1	1	0	10	00	01
326	87	0DFF42D	42BFB18B	1E0FF5DEA	5524CC40CE	1	1	1	0	1	00	10	00
327	88	1BFE85B	057F6317	1C1FEBBD5	2A4998819C	1	0	1	0	0	00	00	10
328	89	17FD0B7	0AFEC62E	183FD77AA	5493310339	0	1	1	1	1	01	00	00
329	90	0FFA16F	15FD8C5C	107FAEF55	2926620672	1	1	0	0	1	00	01	00
330	91	1FF42DF	2BFB18B9	00FF5DEAA	524CC40CE5	1	1	0	0	0	10	00	01
331	92	1FE85BF	57F63172	01FEBBD55	24998819CA	1	1	0	1	1	00	10	00
332	93	1FD0B7F	2FEC62E4	03FD77AAA	4933103394	1	1	0	0	0	00	00	10
333	94	1FA16FF	5FD8C5C9	07FAEF555	1266206728	1	1	0	0	0	01	00	00
334	95	1F42DFF	3FB18B93	0FF5DEAAA	24CC40CE51	1	1	1	1	1	11	01	00
335	96	1E85BFF	7F631727	1FEBBD554	4998819CA3	1	0	1	1	0	11	11	01
336	97	1D0B7FE	7EC62E4F	1FD77AAA9	1331033947	1	1	1	0	0	10	11	11
337	98	1A16FFC	7D8C5C9F	1FAEF5553	266206728E	1	1	1	0	1	10	10	11
338	99	142DFF9	7B18B93F	1F5DEAAA7	4CC40CE51D	0	0	1	1	0	01	10	10
339	100	085BFF3	7631727F	1EBBD554E	198819CA3B	1	0	1	1	0	10	01	10
340	101	10B7FE6	6C62E4FF	1D77AAA9C	3310339477	0	0	1	0	1	00	10	01
341	102	016FFCC	58C5C9FE	1AEF55538	66206728EE	0	1	1	0	0	00	00	10
342	103	02DFF98	318B93FC	15DEAAA70	4C40CE51DC	0	1	0	0	1	00	00	00

343	104	05BFF31	631727F8	0BBD554E1	18819CA3B9	0	0	1	1	0	01	00	00
344	105	0B7FE62	462E4FF1	177AAA9C2	3103394772	1	0	0	0	0	00	01	00
345	106	16FFCC5	0C5C9FE2	0EF555384	6206728EE4	0	0	1	0	1	11	00	01
346	107	0DFF98A	18B93FC4	1DEAAA709	440CE51DC9	1	1	1	0	0	00	11	00
347	108	1BFF315	31727F88	1BD554E12	0819CA3B93	1	0	1	0	0	11	00	11
348	109	17FE62A	62E4FF11	17AAA9C24	1033947726	0	1	0	0	0	01	11	00
349	110	0FFCC54	45C9FE22	0F5553849	206728EE4C	1	1	1	0	0	01	01	11
350	111	1FF98A8	0B93FC44	1EAAA7093	40CE51DC99	1	1	1	1	1	00	01	01
351	112	1FF3150	1727F889	1D554E127	019CA3B933	1	0	1	1	1	10	00	01
352	113	1FE62A0	2E4FF112	1AAA9C24F	0339477267	1	0	1	0	0	00	10	00
353	114	1FCC541	5C9FE225	15553849E	06728EE4CF	1	1	0	0	0	00	00	10
354	115	1F98A82	393FC44B	0AAA7093C	0CE51DC99F	1	0	1	1	1	01	00	00
355	116	1F31504	727F8897	1554E1279	19CA3B933E	1	0	0	1	1	00	01	00
356	117	1E62A09	64FF112F	0AA9C24F2	339477267D	1	1	1	1	0	01	00	01
357	118	1CC5412	49FE225E	1553849E4	6728EE4CFB	1	1	0	0	1	00	01	00
358	119	198A824	13FC44BC	0AA7093C9	4E51DC99F7	1	1	1	0	1	10	00	01
359	120	1315049	27F88979	154E12792	1CA3B933EE	0	1	0	1	0	00	10	00
360	121	062A093	4FF112F3	0A9C24F24	39477267DC	0	1	1	0	0	00	00	10
361	122	0C54127	1FE225E6	153849E48	728EE4CFB8	1	1	0	1	1	01	00	00
362	123	18A824E	3FC44BCD	0A7093C91	651DC99F71	1	1	1	0	0	11	01	00
363	124	115049C	7F88979A	14E127922	4A3B933EE2	0	1	0	0	0	10	11	01
364	125	02A0938	7F112F35	09C24F244	1477267DC5	0	0	1	0	1	01	10	11

1.4 THIRD SET OF SAMPLES

Initial values for the key, pan address and clock

K'c3[0] = FF K'c3[1] = FF K'c3[2] = FF K'c3[3] = FF
K'c3[4] = FF K'c3[5] = FF K'c3[6] = FF K'c3[7] = FF
K'c3[8] = FF K'c3[9] = FF K'c3[10] = FF K'c3[11] = FF
K'c3[12] = FF K'c3[13] = FF K'c3[14] = FF K'c3[15] = FF

Addr3[0] = FF Addr3[1] = FF Addr3[2] = FF
Addr3[3] = FF Addr3[4] = FF Addr3[5] = FF

CL3[0] = FF CL3[1] = FF CL3[2] = FF CL3[3] = 03

Fill LFSRs with initial data

t	clk#	LFSR1	LFSR2	LFSR3	LFSR4	X1	X2	X3	X4	Z	C[t+1]	C[t]	C[t-1]
0	0	0000000*	00000000*	000000000*	0000000000*	0	0	0	0	0	00	00	00
1	1	0000001*	00000001*	000000001*	0000000001*	0	0	0	0	0	00	00	00
2	2	0000003*	00000002*	000000003*	0000000003*	0	0	0	0	0	00	00	00
3	3	0000007*	00000004*	000000007*	0000000007*	0	0	0	0	0	00	00	00
4	4	000000F*	00000009*	00000000F*	000000000F*	0	0	0	0	0	00	00	00
5	5	000001F*	00000013*	00000001F*	000000001F*	0	0	0	0	0	00	00	00
6	6	000003F*	00000027*	00000003F*	000000003F*	0	0	0	0	0	00	00	00
7	7	000007F*	0000004F*	00000007F*	000000007F*	0	0	0	0	0	00	00	00
8	8	00000FF*	0000009F*	0000000FF*	00000000FF*	0	0	0	0	0	00	00	00
9	9	00001FF*	0000013F*	0000001FF*	00000001FF*	0	0	0	0	0	00	00	00
10	10	00003FF*	0000027F*	0000003FF*	00000003FF*	0	0	0	0	0	00	00	00
11	11	00007FF*	000004FF*	0000007FF*	00000007FF*	0	0	0	0	0	00	00	00
12	12	0000FFF*	000009FF*	000000FFF*	0000000FFF*	0	0	0	0	0	00	00	00
13	13	0001FFF*	000013FF*	000001FFF*	0000001FFF*	0	0	0	0	0	00	00	00
14	14	0003FFF*	000027FF*	000003FFF*	0000003FFF*	0	0	0	0	0	00	00	00
15	15	0007FFF*	00004FFF*	000007FFF*	0000007FFF*	0	0	0	0	0	00	00	00
16	16	000FFFF*	00009FFF*	00000FFFF*	000000FFFF*	0	0	0	0	0	00	00	00
17	17	001FFFF*	00013FFF*	00001FFFF*	000001FFFF*	0	0	0	0	0	00	00	00
18	18	003FFFF*	00027FFF*	00003FFFF*	000003FFFF*	0	0	0	0	0	00	00	00
19	19	007FFFF*	0004FFFF*	00007FFFF*	000007FFFF*	0	0	0	0	0	00	00	00
20	20	00FFFFF*	0009FFFF*	0000FFFFF*	00000FFFFF*	0	0	0	0	0	00	00	00
21	21	01FFFFF*	0013FFFF*	0001FFFFF*	00001FFFFF*	0	0	0	0	0	00	00	00
22	22	03FFFFF*	0027FFFF*	0003FFFFF*	00003FFFFF*	0	0	0	0	0	00	00	00
23	23	07FFFFF*	004FFFFF*	0007FFFFF*	00007FFFFF*	0	0	0	0	0	00	00	00
24	24	0FFFFFF*	009FFFFF*	000FFFFFF*	0000FFFFFF*	1	1	0	0	0	01	00	00
25	25	1FFFFFF*	013FFFFF*	001FFFFFF*	0001FFFFFF*	1	0	0	0	1	00	00	00
26	26	1FFFFFF	027FFFFF*	003FFFFFF*	0003FFFFFF*	1	0	0	0	1	00	00	00
27	27	1FFFFFF	04FFFFFF*	007FFFFFF*	0007FFFFFF*	1	1	0	0	0	01	00	00
28	28	1FFFFFF	09FFFFFF*	00FFFFFFF*	000FFFFFFF*	1	1	0	0	0	01	00	00
29	29	1FFFFFF	13FFFFFF*	01FFFFFFF*	001FFFFFFF*	1	1	0	0	0	01	00	00
30	30	1FFFFFF	27FFFFFF*	03FFFFFFF*	003FFFFFFF*	1	1	0	0	0	01	00	00
31	31	1FFFFFF	4FFFFFFF*	07FFFFFFF*	007FFFFFFF*	1	1	0	0	0	01	00	00
32	32	1FFFFFF	1FFFFFFF	0FFFFFFFF*	00FFFFFFFF*	1	1	1	1	0	10	00	00
33	33	1FFFFFF	3FFFFFFE	1FFFFFFFF*	01FFFFFFFF*	1	1	1	1	0	10	00	00
34	34	1FFFFFF	7FFFFFFC	1FFFFFFFF	03FFFFFFFF*	1	1	1	1	0	10	00	00

35	35	1FFFFF	7FFFFF9	1FFFFFFF	07FFFFFFF*	1	1	1	1	0	10	00	00
36	36	1FFFFF	7FFFFF3	1FFFFFFF	0FFFFFFFF*	1	1	1	1	0	10	00	00
37	37	1FFFFF	7FFFFE7	1FFFFFFF	1FFFFFFFF*	1	1	1	1	0	10	00	00
38	38	1FFFFF	7FFFFCF	1FFFFFFF	3FFFFFFFF*	1	1	1	1	0	10	00	00
39	39	1FFFFF	7FFFF9F	1FFFFFFF	7FFFFFFFF*	1	1	1	1	0	10	00	00

Start clocking Summation Combiner

40	1	1FFFFF	7FFFF3F	1FFFFFFF	7FFFFFFFF	1	1	1	1	0	01	10	00
41	2	1FFFFF	7FFFE7F	1FFFFFFF	7FFFFFFFF	1	1	1	1	1	10	01	10
42	3	1FFFFF	7FFFCFF	1FFFFFFF	7FFFFFFFF	1	1	1	1	0	10	10	01
43	4	1FFFFF	7FFF9FF	1FFFFFFF	7FFFFFFFF	1	1	1	1	0	00	10	10
44	5	1FFFFF	7FFF3FF	1FFFFFFF	7FFFFFFFF	1	1	1	1	0	11	00	10
45	6	1FFFFF	7FFE7FE	1FFFFFFF	7FFFFFFFF	1	1	1	1	1	00	11	00
46	7	1FFFFF	7FFCFFC	1FFFFFFF	7FFFFFFFF	1	1	1	1	0	00	00	11
47	8	1FFFFF	7FFF9FF9	1FFFFFFF	7FFFFFFFF	1	1	1	1	0	10	00	00
48	9	1FFFFF	7FFF3FF3	1FFFFFFF	7FFFFFFFF	1	1	1	1	0	01	10	00
49	10	1FFFFF	7FFE7FE6	1FFFFFFF	7FFFFFFFF	1	1	1	1	1	10	01	10
50	11	1FFFFE	7FFCFFCC	1FFFFFFE	7FFFFFFFF	1	1	1	1	0	10	10	01
51	12	1FFFFC	7FF9FF99	1FFFFFFC	7FFFFFFFF	1	1	1	1	0	00	10	10
52	13	1FFFF8	7FF3FF33	1FFFFFF8	7FFFFFFFF	1	1	1	1	0	11	00	10
53	14	1FFFF0	7FE7FE67	1FFFFFF0	7FFFFFFFF	1	1	1	1	1	00	11	00
54	15	1FFFE0	7FCFFCCF	1FFFFFE1	7FFFFFFFF	1	1	1	1	0	00	00	11
55	16	1FFFC0	7F9FF99F	1FFFFFC3	7FFFFFFFF	1	1	1	1	0	10	00	00
56	17	1FFF80	7F3FF33E	1FFFFF87	7FFFFFFFE	1	0	1	1	1	00	10	00
57	18	1FFF00	7E7FE67C	1FFFFF0F	7FFFFFFFC	1	0	1	1	1	00	00	10
58	19	1FFE01	7CFFCCF8	1FFFFE1E	7FFFFFFF8	1	1	1	1	0	10	00	00
59	20	1FFC03	79FF99F0	1FFFFC3C	7FFFFFFF0	1	1	1	1	0	01	10	00
60	21	1FFF807	73FF33E0	1FFFF878	7FFFFFFE1	1	1	1	1	1	10	01	10
61	22	1FFF00F	67FE67C0	1FFFF0F0	7FFFFFFC3	1	1	1	1	0	10	10	01
62	23	1FFE01E	4FFCCF80	1FFFE1E1	7FFFFFF87	1	1	1	1	0	00	10	10
63	24	1FFC03C	1FF99F00	1FFFC3C3	7FFFFFF0F	1	1	1	1	0	11	00	10
64	25	1FF8078	3FF33E01	1FFF8787	7FFFFFE1E	1	1	1	1	1	00	11	00
65	26	1FF00F0	7FE67C02	1FFF0F0F	7FFFFFC3C	1	1	1	1	0	00	00	11
66	27	1FE01E1	7FCCF805	1FFE1E1E	7FFFFF878	1	1	1	1	0	10	00	00
67	28	1FC03C3	7F99F00A	1FFC3C3C	7FFFFF0F0	1	1	1	1	0	01	10	00
68	29	1F80787	7F33E015	1FFF87878	7FFFFE1E1	1	0	1	1	0	10	01	10
69	30	1F00F0F	7E67C02A	1FFF0F0F0	7FFFFC3C3	1	0	1	1	1	11	10	01
70	31	1E01E1E	7CCF8054	1FFE1E1E1	7FFFF8787	1	1	1	1	1	01	11	10
71	32	1C03C3C	799F00A9	1FFC3C3C3	7FFFF0F0F	1	1	1	1	1	01	01	11
72	33	1807878	733E0152	1FF878787	7FFFE1E1E	1	0	1	1	0	00	01	01
73	34	100F0F0	667C02A5	1FF0F0F0F	7FFFC3C3C	0	0	1	1	0	10	00	01
74	35	001E1E0	4CF8054B	1FE1E1E1F	7FFFF87878	0	1	1	1	1	00	10	00
75	36	003C3C1	19F00A96	1FC3C3C3F	7FFFF0F0F0	0	1	1	1	1	00	00	10
76	37	0078783	33E0152C	1F878787F	7FFFE1E1E1	0	1	1	1	1	01	00	00
77	38	00F0F07	67C02A59	1F0F0F0FF	7FFFC3C3C3	0	1	1	1	0	11	01	00
78	39	01E1E0E	4F8054B3	1E1E1E1FF	7FFF878787	0	1	1	1	0	11	11	01
79	40	03C3C1C	1F00A966	1C3C3C3FF	7FFF0F0F0F	0	0	1	1	1	11	11	11
80	41	0787838	3E0152CC	1878787FF	7FFE1E1E1E	0	0	1	1	1	11	11	11
81	42	0F0F070	7C02A598	10F0F0FFF	7FFC3C3C3C	1	0	0	1	1	11	11	11
82	43	1E1E0E0	78054B30	01E1E1FFF	7FF8787878	1	0	0	1	1	11	11	11
83	44	1C3C1C0	700A9660	03C3C3FFE	7FF0F0F0F0	1	0	0	1	1	11	11	11
84	45	1878380	60152CC0	078787FFC	7FE1E1E1E0	1	0	0	1	1	11	11	11
85	46	10F0700	402A5980	0F0F0FFF8	7FC3C3C3C0	0	0	1	1	1	11	11	11
86	47	01E0E00	0054B300	1E1E1FFF0	7F87878780	0	0	1	1	1	11	11	11
87	48	03C1C00	00A96601	1C3C3FFE0	7F0F0F0F00	0	1	1	0	1	11	11	11
88	49	0783800	0152CC03	18787FFC0	7E1E1E1E01	0	0	1	0	0	11	11	11

89	50	0F07000	02A59806	10F0FFF80	7C3C3C3C03	1	1	0	0	1	11	11	11
90	51	1E0E000	054B300D	01E1FFF00	7878787807	1	0	0	0	0	11	11	11
91	52	1C1C001	0A96601A	03C3FFE01	70F0F0F00F	1	1	0	1	0	10	11	11
92	53	1838003	152CC035	0787FFC03	61E1E1E01E	1	0	0	1	0	10	10	11
93	54	1070007	2A59806B	0F0FFF807	43C3C3C03C	0	0	1	1	0	01	10	10
94	55	00E000F	54B300D7	1E1FFF00F	0787878078	0	1	1	1	0	10	01	10
95	56	01C001F	296601AE	1C3FFE01F	0F0F0F00F1	0	0	1	0	1	00	10	01
96	57	038003F	52CC035C	187FFC03F	1E1E1E01E2	0	1	1	0	0	00	00	10
97	58	070007F	259806B8	10FFF807F	3C3C3C03C4	0	1	0	0	1	00	00	00
98	59	0E000FE	4B300D71	01FFF00FE	7878780788	1	0	0	0	1	00	00	00
99	60	1C001FD	16601AE2	03FFE01FD	70F0F00F10	1	0	0	1	0	01	00	00
100	61	18003FA	2CC035C5	07FFC03FB	61E1E01E21	1	1	0	1	0	11	01	00
101	62	10007F4	59806B8B	0FFF807F7	43C3C03C43	0	1	1	1	0	11	11	01
102	63	0000FE8	3300D717	1FFF00FEE	0787807887	0	0	1	1	1	11	11	11
103	64	0001FD0	6601AE2F	1FFE01FDC	0F0F00F10E	0	0	1	0	0	11	11	11
104	65	0003FA0	4C035C5F	1FFC03FB8	1E1E01E21D	0	0	1	0	0	11	11	11
105	66	0007F40	1806B8BE	1FF807F70	3C3C03C43B	0	0	1	0	0	11	11	11
106	67	000FE81	300D717C	1FF00FEE1	7878078877	0	0	1	0	0	11	11	11
107	68	001FD02	601AE2F8	1FE01FDC2	70F00F10EF	0	0	1	1	1	11	11	11
108	69	003FA05	4035C5F0	1FC03FB84	61E01E21DE	0	0	1	1	1	11	11	11
109	70	007F40B	006B8BE0	1F807F708	43C03C43BC	0	0	1	1	1	11	11	11
110	71	00FE816	00D717C0	1F00FEE11	0780788778	0	1	1	1	0	10	11	11
111	72	01FD02C	01AE2F81	1E01FDC23	0F00F10EF1	0	1	1	0	0	10	10	11
112	73	03FA059	035C5F02	1C03FB847	1E01E21DE3	0	0	1	0	1	10	10	10
113	74	07F40B3	06B8BE05	1807F708F	3C03C43BC7	0	1	1	0	0	01	10	10
114	75	0FE8166	0D717C0B	100FEE11E	780788778F	1	0	0	0	0	01	01	10
115	76	1FD02CD	1AE2F817	001FDC23D	700F10EF1F	1	1	0	0	1	11	01	01
116	77	1FA059B	35C5F02F	003FB847A	601E21DE3F	1	1	0	0	1	10	11	01
117	78	1F40B37	6B8BE05E	007F708F4	403C43BC7F	1	1	0	0	0	10	10	11
118	79	1E8166E	5717C0BD	00FEE11E9	00788778FF	1	0	0	0	1	10	10	10
119	80	1D02CDC	2E2F817A	01FDC23D3	00F10EF1FE	1	0	0	1	0	01	10	10
120	81	1A059B9	5C5F02F5	03FB847A6	01E21DE3FD	1	0	0	1	1	01	01	10
121	82	140B373	38BE05EB	07F708F4C	03C43BC7FB	0	1	0	1	1	11	01	01
122	83	08166E7	717C0BD7	0FEE11E98	0788778FF7	1	0	1	1	0	11	11	01
123	84	102CDCF	62F817AE	1FDC23D31	0F10EF1FEF	0	1	1	0	1	11	11	11
124	85	0059B9F	45F02F5C	1FB847A63	1E21DE3FDE	0	1	1	0	1	11	11	11
125	86	00B373E	0BE05EB9	1F708F4C7	3C43BC7FBC	0	1	1	0	1	11	11	11
126	87	0166E7D	17C0BD72	1EE11E98F	788778FF78	0	1	1	1	0	10	11	11
127	88	02CDCFB	2F817AE5	1DC23D31F	710EF1FEF1	0	1	1	0	0	10	10	11
128	89	059B9F7	5F02F5CA	1B847A63F	621DE3FDE2	0	0	1	0	1	10	10	10
129	90	0B373EF	3E05EB94	1708F4C7F	443BC7FBC4	1	0	0	0	1	10	10	10
130	91	166E7DF	7C0BD728	0E11E98FF	08778FF788	0	0	1	0	1	10	10	10
131	92	0CDCFBE	7817AE50	1C23D31FF	10EF1FEF10	1	0	1	1	1	01	10	10
132	93	19B9F7D	702F5CA1	1847A63FE	21DE3FDE21	1	0	1	1	0	10	01	10
133	94	1373EFB	605EB942	108F4C7FC	43BC7FBC43	0	0	0	1	1	00	10	01
134	95	06E7DF7	40BD7285	011E98FF8	0778FF7886	0	1	0	0	1	01	00	10
135	96	0DCFBEF	017AE50A	023D31FF0	0EF1FEF10D	1	0	0	1	1	00	01	00
136	97	1B9F7DF	02F5CA15	047A63FE1	1DE3FDE21A	1	1	0	1	1	10	00	01
137	98	173EFBF	05EB942B	08F4C7FC3	3BC7FBC434	0	1	1	1	1	00	10	00
138	99	0E7DF7F	0BD72856	11E98FF87	778FF78869	1	1	0	1	1	00	00	10
139	100	1CFBEFF	17AE50AC	03D31FF0F	6F1FEF10D3	1	1	0	0	0	01	00	00
140	101	19F7DFE	2F5CA159	07A63FE1E	5E3FDE21A7	1	0	0	0	0	00	01	00
141	102	13EFBFC	5EB942B3	0F4C7FC3C	3C7FBC434F	0	1	1	0	0	10	00	01
142	103	07DF7F8	3D728566	1E98FF878	78FF78869F	0	0	1	1	0	00	10	00
143	104	0FBEFF0	7AE50ACD	1D31FF0F0	71FEF10D3E	1	1	1	1	0	11	00	10
144	105	1F7DFE1	75CA159B	1A63FE1E1	63FDE21A7D	1	1	1	1	1	00	11	00
145	106	1EFBFC3	6B942B36	14C7FC3C3	47FBC434FB	1	1	0	1	1	11	00	11

146	107	1DF7F86	5728566D	098FF8786	0FF78869F7	1	0	1	1	0	00	11	00
147	108	1BEFF0C	2E50ACDB	131FF0F0C	1FEF10D3EF	1	0	0	1	0	11	00	11
148	109	17DFE19	5CA159B6	063FE1E19	3FDE21A7DF	0	1	0	1	1	01	11	00
149	110	0FBFC33	3942B36D	0C7FC3C32	7FBC434FBF	1	0	1	1	0	01	01	11
150	111	1F7F866	728566DB	18FF87865	7F78869F7E	1	1	1	0	0	00	01	01
151	112	1EFF0CC	650ACDB6	11FF0F0CB	7EF10D3EFC	1	0	0	1	0	10	00	01
152	113	1DFE199	4A159B6D	03FE1E196	7DE21A7DF9	1	0	0	1	0	00	10	00
153	114	1BFC333	142B36DB	07FC3C32C	7BC434FBF3	1	0	0	1	0	00	00	10
154	115	17F8666	28566DB6	0FF878659	778869F7E6	0	0	1	1	0	01	00	00
155	116	0FF0CCC	50ACDB6D	1FF0F0CB3	6F10D3EFCC	1	1	1	0	0	11	01	00
156	117	1FE1999	2159B6DA	1FE1E1966	5E21A7DF99	1	0	1	0	1	10	11	01
157	118	1FC3332	42B36DB5	1FC3C32CC	3C434FBF33	1	1	1	0	1	10	10	11
158	119	1F86664	0566DB6B	1F8786599	78869F7E67	1	0	1	1	1	01	10	10
159	120	1F0CCC8	0ACDB6D6	1F0F0CB33	710D3EFCCE	1	1	1	0	0	10	01	10
160	121	1E19991	159B6DAC	1E1E19666	621A7DF99D	1	1	1	0	1	11	10	01
161	122	1C33323	2B36DB58	1C3C32CCC	4434FBF33B	1	0	1	0	1	00	11	10
162	123	1866647	566DB6B0	187865999	0869F7E676	1	0	1	0	0	11	00	11
163	124	10CCC8F	2CDB6D60	10F0CB333	10D3EFCCEC	0	1	0	1	1	01	11	00
164	125	019991E	59B6DAC0	01E196666	21A7DF99D9	0	1	0	1	1	10	01	11
165	126	033323C	336DB580	03C32CCCD	434FBF33B3	0	0	0	0	0	00	10	01
166	127	0666478	66DB6B01	07865999A	069F7E6766	0	1	0	1	0	00	00	10
167	128	0CCC8F0	4DB6D603	0F0CB3334	0D3EFCCECD	1	1	1	0	1	01	00	00
168	129	19991E1	1B6DAC07	1E1966669	1A7DF99D9B	1	0	1	0	1	00	01	00
169	130	13323C3	36DB580E	1C32CCCD3	34FBF33B37	0	1	1	1	1	10	00	01
170	131	0664786	6DB6B01C	1865999A7	69F7E6766F	0	1	1	1	1	00	10	00
171	132	0CC8F0D	5B6D6039	10CB3334F	53EFCCECDF	1	0	0	1	0	00	00	10
172	133	1991E1A	36DAC073	01966669E	27DF99D9BF	1	1	0	1	1	01	00	00
173	134	1323C35	6DB580E6	032CCCD3C	4FBF33B37E	0	1	0	1	1	00	01	00
174	135	064786A	5B6B01CD	065999A78	1F7E6766FC	0	0	0	0	0	11	00	01
175	136	0C8F0D5	36D6039B	0CB3334F0	3EFCCECDF9	1	1	1	1	1	00	11	00
176	137	191E1AA	6DAC0737	1966669E1	7DF99D9BF3	1	1	1	1	0	00	00	11
177	138	123C354	5B580E6E	12CCCD3C3	7BF33B37E7	0	0	0	1	1	00	00	00
178	139	04786A9	36B01CDC	05999A787	77E6766FCE	0	1	0	1	0	01	00	00
179	140	08F0D53	6D6039B8	0B3334F0E	6FCCECDF9C	1	0	1	1	0	11	01	00
180	141	11E1AA6	5AC07370	166669E1D	5F99D9BF38	0	1	0	1	1	10	11	01
181	142	03C354C	3580E6E0	0CCCD3C3A	3F33B37E70	0	1	1	0	0	10	10	11
182	143	0786A99	6B01CDC0	1999A7875	7E6766FCE1	0	0	1	0	1	10	10	10
183	144	0F0D533	56039B81	13334F0EB	7CCECDF9C2	1	0	0	1	0	01	10	10
184	145	1E1AA66	2C073703	06669E1D6	799D9BF385	1	0	0	1	1	01	01	10
185	146	1C354CC	580E6E06	0CCD3C3AC	733B37E70B	1	0	1	0	1	11	01	01
186	147	186A998	301CDC0C	199A78759	66766FCE17	1	0	1	0	1	10	11	01
187	148	10D5331	6039B818	1334F0EB2	4CECDF9C2F	0	0	0	1	1	01	10	11
188	149	01AA662	40737031	0669E1D65	19D9BF385E	0	0	0	1	0	01	01	10
189	150	0354CC5	00E6E063	0CD3C3ACB	33B37E70BD	0	1	1	1	0	00	01	01
190	151	06A998A	01CDC0C6	19A787596	6766FCE17B	0	1	1	0	0	10	00	01
191	152	0D53315	039B818C	134F0EB2C	4ECDF9C2F6	1	1	0	1	1	00	10	00
192	153	1AA662A	07370318	069E1D659	1D9BF385ED	1	0	0	1	0	00	00	10
193	154	154CC54	0E6E0630	0D3C3ACB3	3B37E70BDB	0	0	1	0	1	00	00	00
194	155	0A998A8	1CDC0C60	1A7875967	766FCE17B6	1	1	1	0	1	01	00	00
195	156	1533151	39B818C0	14F0EB2CE	6CDF9C2F6C	0	1	0	1	1	00	01	00
196	157	0A662A3	73703180	09E1D659D	59BF385ED8	1	0	1	1	1	10	00	01
197	158	14CC547	66E06301	13C3ACB3A	337E70BDB0	0	1	0	0	1	11	10	00
198	159	0998A8E	4DC0C602	078759675	66FCE17B61	1	1	0	1	0	01	11	10
199	160	133151D	1B818C05	0F0EB2CEB	4DF9C2F6C2	0	1	1	1	0	01	01	11
200	161	0662A3B	3703180B	1E1D659D6	1BF385ED85	0	0	1	1	1	11	01	01
201	162	0CC5477	6E063017	1C3ACB3AC	37E70BDB0B	1	0	1	1	0	11	11	01
202	163	198A8EF	5C0C602F	187596759	6FCE17B617	1	0	1	1	0	10	11	11

Appendix IV - Sample Data

203	164	13151DE	3818C05F	10EB2CEB2	5F9C2F6C2F	0	0	0	1	1	01	10	11
204	165	062A3BC	703180BF	01D659D65	3F385ED85E	0	0	0	0	1	00	01	10
205	166	0C54779	6063017E	03ACB3ACB	7E70BDB0BD	1	0	0	0	1	11	00	01
206	167	18A8EF2	40C602FD	075967597	7CE17B617B	1	1	0	1	0	00	11	00
207	168	1151DE4	018C05FA	0EB2CEB2F	79C2F6C2F7	0	1	1	1	1	11	00	11
208	169	02A3BC9	03180BF5	1D659D65E	7385ED85EE	0	0	1	1	1	01	11	00
209	170	0547793	063017EB	1ACB3ACBC	670BDB0BDC	0	0	1	0	0	10	01	11
210	171	0A8EF27	0C602FD6	159675978	4E17B617B9	1	0	0	0	1	00	10	01
211	172	151DE4E	18C05FAD	0B2CEB2F1	1C2F6C2F73	0	1	1	0	0	00	00	10
212	173	0A3BC9C	3180BF5A	1659D65E3	385ED85EE6	1	1	0	0	0	01	00	00
213	174	1477938	63017EB5	0CB3ACBC6	70BDB0BDCC	0	0	1	1	1	00	01	00
214	175	08EF270	4602FD6A	19675978D	617B617B99	1	0	1	0	0	10	00	01
215	176	11DE4E1	0C05FAD5	12CEB2F1A	42F6C2F733	0	0	0	1	1	11	10	00
216	177	03BC9C3	180BF5AA	059D65E34	05ED85EE67	0	0	0	1	0	00	11	10
217	178	0779387	3017EB55	0B3ACBC68	0BDB0BDCCF	0	0	1	1	0	11	00	11
218	179	0EF270F	602FD6AA	1675978D0	17B617B99F	1	0	0	1	1	01	11	00
219	180	1DE4E1F	405FAD54	0CEB2F1A1	2F6C2F733F	1	0	1	0	1	10	01	11
220	181	1BC9C3F	00BF5AA9	19D65E342	5ED85EE67F	1	1	1	1	0	10	10	01
221	182	179387F	017EB552	13ACBC684	3DB0BDCCFE	0	0	0	1	1	10	10	10
222	183	0F270FF	02FD6AA5	075978D09	7B617B99FC	1	1	0	0	0	01	10	10
223	184	1E4E1FF	05FAD54A	0EB2F1A12	76C2F733F9	1	1	1	1	1	10	01	10
224	185	1C9C3FE	0BF5AA94	1D65E3425	6D85EE67F2	1	1	1	1	0	10	10	01
225	186	19387FD	17EB5529	1ACBC684B	5B0BDCCFE4	1	1	1	0	1	01	10	10
226	187	1270FFA	2FD6AA53	15978D096	3617B99FC9	0	1	0	0	0	01	01	10
227	188	04E1FF5	5FAD54A7	0B2F1A12C	6C2F733F93	0	1	1	0	1	11	01	01
228	189	09C3FEB	3F5AA94E	165E34258	585EE67F27	1	0	0	0	0	10	11	01
229	190	1387FD7	7EB5529C	0CBC684B1	30BDCCFE4F	0	1	1	1	1	10	10	11
230	191	070FFAE	7D6AA538	1978D0962	617B99FC9E	0	0	1	0	1	10	10	10
231	192	0E1FF5C	7AD54A70	12F1A12C4	42F733F93D	1	1	0	1	1	01	10	10
232	193	1C3FEB9	75AA94E1	05E342588	05EE67F27A	1	1	0	1	0	10	01	10
233	194	187FD73	6B5529C3	0BC684B10	0BDCCFE4F4	1	0	1	1	1	11	10	01
234	195	10FFAE6	56AA5386	178D09621	17B99FC9E8	0	1	0	1	1	00	11	10
235	196	01FF5CC	2D54A70C	0F1A12C43	2F733F93D0	0	0	1	0	1	10	00	11
236	197	03FEB98	5AA94E19	1E3425887	5EE67F27A1	0	1	1	1	1	00	10	00
237	198	07FD731	35529C33	1C684B10F	3DCCFE4F42	0	0	1	1	0	00	00	10
238	199	0FFAE63	6AA53866	18D09621F	7B99FC9E84	1	1	1	1	0	10	00	00
239	200	1FF5CC6	554A70CD	11A12C43F	7733F93D09	1	0	0	0	1	11	10	00

```
Z[0]  = 59
Z[1]  = 3B
Z[2]  = EF
Z[3]  = 07
Z[4]  = 13
Z[5]  = 70
Z[6]  = 9B
Z[7]  = B7
Z[8]  = 52
Z[9]  = 8F
Z[10] = 3E
Z[11] = B9
Z[12] = A5
Z[13] = AC
Z[14] = EA
Z[15] = 9E
```

```
================================================================================================
         Reload this pattern into the LFSRs
         Hold content of Summation Combiner regs and calculate new C[t+1] and Z values
================================================================================================
LFSR1  <= 1521359
LFSR2  <= 528F703B
LFSR3  <= 0AC3E9BEF
LFSR4  <= 4FEAB9B707
C[t+1] <= 00

================================================================================================
         Generating 125 key symbols (encryption/decryption sequence)
================================================================================================
```

240	1	1521359	528F703B	0AC3E9BEF	4FEAB9B707	0	1	1	1	1	00	10	00
241	2	0A426B3	251EE076	1587D37DE	1FD5736E0F	1	0	0	1	0	00	00	10
242	3	1484D67	4A3DC0ED	0B0FA6FBD	3FAAE6DC1E	0	0	1	1	0	01	00	00
243	4	0909ACF	147B81DA	161F4DF7A	7F55CDB83D	1	0	0	0	0	00	01	00
244	5	121359E	28F703B5	0C3E9BEF5	7EAB9B707B	0	1	1	1	1	10	00	01
245	6	0426B3C	51EE076B	187D37DEB	7D5736E0F6	0	1	1	0	0	00	10	00
246	7	084D679	23DC0ED6	10FA6FBD7	7AAE6DC1EC	1	1	0	1	1	00	00	10
247	8	109ACF2	47B81DAC	01F4DF7AF	755CDB83D8	0	1	0	0	1	00	00	00
248	9	01359E4	0F703B59	03E9BEF5E	6AB9B707B1	0	0	0	1	1	00	00	00
249	10	026B3C8	1EE076B3	07D37DEBD	55736E0F63	0	1	0	0	1	00	00	00
250	11	04D6791	3DC0ED67	0FA6FBD7A	2AE6DC1EC7	0	1	1	1	1	01	00	00
251	12	09ACF22	7B81DACF	1F4DF7AF4	55CDB83D8F	1	1	1	1	1	11	01	00
252	13	1359E44	7703B59E	1E9BEF5E8	2B9B707B1F	0	0	1	1	1	10	11	01
253	14	06B3C88	6E076B3C	1D37DEBD0	5736E0F63F	0	0	1	0	1	01	10	11
254	15	0D67911	5C0ED678	1A6FBD7A1	2E6DC1EC7E	1	0	1	0	1	01	01	10
255	16	1ACF223	381DACF0	14DF7AF42	5CDB83D8FD	1	0	0	1	1	11	01	01
256	17	159E446	703B59E0	09BEF5E85	39B707B1FA	0	0	1	1	1	10	11	01
257	18	0B3C88C	6076B3C0	137DEBD0A	736E0F63F4	1	0	0	0	1	01	10	11
258	19	1679118	40ED6780	06FBD7A15	66DC1EC7E8	0	1	0	1	1	01	01	10
259	20	0CF2231	01DACF00	0DF7AF42A	4DB83D8FD1	1	1	1	1	1	00	01	01
260	21	19E4463	03B59E01	1BEF5E854	1B707B1FA3	1	1	1	0	1	10	00	01
261	22	13C88C6	076B3C03	17DEBD0A9	36E0F63F47	0	0	0	1	1	11	10	00
262	23	079118C	0ED67807	0FBD7A152	6DC1EC7E8E	0	1	1	1	0	01	11	10
263	24	0F22318	1DACF00E	1F7AF42A4	5B83D8FD1D	1	1	1	1	1	01	01	11
264	25	1E44630	3B59E01C	1EF5E8548	3707B1FA3B	1	0	1	0	1	11	01	01
265	26	1C88C61	76B3C039	1DEBD0A91	6E0F63F477	1	1	1	0	0	11	11	01
266	27	19118C3	6D678073	1BD7A1523	5C1EC7E8EF	1	0	1	0	1	11	11	11
267	28	1223187	5ACF00E6	17AF42A46	383D8FD1DE	0	1	0	0	0	11	11	11
268	29	044630E	359E01CC	0F5E8548D	707B1FA3BD	0	1	1	0	1	11	11	11
269	30	088C61C	6B3C0399	1EBD0A91A	60F63F477B	1	0	1	1	0	10	11	11
270	31	1118C39	56780733	1D7A15234	41EC7E8EF6	0	0	1	1	0	10	10	11
271	32	0231872	2CF00E67	1AF42A468	03D8FD1DEC	0	1	1	1	1	01	10	10
272	33	04630E5	59E01CCE	15E8548D1	07B1FA3BD8	0	1	0	1	1	01	01	10
273	34	08C61CB	33C0399D	0BD0A91A3	0F63F477B1	1	1	1	0	0	00	01	01
274	35	118C396	6780733A	17A152347	1EC7E8EF63	0	1	0	1	0	10	00	01
275	36	031872D	4F00E674	0F42A468E	3D8FD1DEC7	0	0	1	1	0	00	10	00
276	37	0630E5A	1E01CCE8	1E8548D1D	7B1FA3BD8E	0	0	1	0	1	01	00	10
277	38	0C61CB5	3C0399D0	1D0A91A3B	763F477B1C	1	0	1	0	1	00	01	00
278	39	18C396A	780733A0	1A1523477	6C7E8EF639	1	0	1	0	0	10	00	01
279	40	11872D5	700E6741	142A468EF	58FD1DEC72	0	0	0	1	1	11	10	00
280	41	030E5AB	601CCE83	08548D1DF	31FA3BD8E5	0	0	1	1	1	00	11	10
281	42	061CB57	40399D07	10A91A3BF	63F477B1CB	0	0	0	1	1	10	00	11
282	43	0C396AF	00733A0F	01523477E	47E8EF6396	1	0	0	1	0	00	10	00
283	44	1872D5F	00E6741F	02A468EFD	0FD1DEC72C	1	1	0	1	1	00	00	10

284	45	10E5ABE	01CCE83F	0548D1DFA	1FA3BD8E58	0	1	0	1	0	01	00	00
285	46	01CB57C	0399D07F	0A91A3BF4	3F477B1CB0	0	1	1	0	1	00	01	00
286	47	0396AF9	0733A0FE	1523477E9	7E8EF63961	0	0	0	1	1	11	00	01
287	48	072D5F3	0E6741FD	0A468EFD2	7D1DEC72C3	0	0	1	0	0	01	11	00
288	49	0E5ABE7	1CCE83FA	148D1DFA4	7A3BD8E587	1	1	0	0	1	10	01	11
289	50	1CB57CE	399D07F4	091A3BF49	7477B1CB0F	1	1	1	0	1	11	10	01
290	51	196AF9D	733A0FE9	123477E92	68EF63961E	1	0	0	1	1	00	11	10
291	52	12D5F3B	66741FD2	0468EFD25	51DEC72C3C	0	0	0	1	1	10	00	11
292	53	05ABE77	4CE83FA4	08D1DFA4B	23BD8E5879	0	1	1	1	1	00	10	00
293	54	0B57CEE	19D07F49	11A3BF496	477B1CB0F2	1	1	0	0	0	00	00	10
294	55	16AF9DC	33A0FE92	03477E92C	0EF63961E4	0	1	0	1	0	01	00	00
295	56	0D5F3B8	6741FD25	068EFD259	1DEC72C3C9	1	0	0	1	1	00	01	00
296	57	1ABE771	4E83FA4B	0D1DFA4B3	3BD8E58793	1	1	1	1	0	01	00	01
297	58	157CEE2	1D07F496	1A3BF4967	77B1CB0F26	0	0	1	1	1	00	01	00
298	59	0AF9DC5	3A0FE92D	1477E92CE	6F63961E4D	1	0	0	0	1	11	00	01
299	60	15F3B8B	741FD25A	08EFD259C	5EC72C3C9B	0	0	1	1	1	01	11	00
300	61	0BE7716	683FA4B4	11DFA4B39	3D8E587937	1	0	0	1	1	10	01	11
301	62	17CEE2D	507F4968	03BF49672	7B1CB0F26E	0	0	0	0	0	00	10	01
302	63	0F9DC5B	20FE92D0	077E92CE4	763961E4DC	1	1	0	0	0	00	00	10
303	64	1F3B8B6	41FD25A0	0EFD259C9	6C72C3C9B9	1	1	1	0	1	01	00	00
304	65	1E7716D	03FA4B40	1DFA4B393	58E5879373	1	1	1	1	1	11	01	00
305	66	1CEE2DB	07F49680	1BF496727	31CB0F26E6	1	1	1	1	1	11	11	01
306	67	19DC5B7	0FE92D00	17E92CE4E	63961E4DCD	1	1	0	1	0	10	11	11
307	68	13B8B6F	1FD25A00	0FD259C9C	472C3C9B9A	0	1	1	0	0	10	10	11
308	69	07716DF	3FA4B400	1FA4B3938	0E58793735	0	1	1	0	0	01	10	10
309	70	0EE2DBF	7F496800	1F4967271	1CB0F26E6A	1	0	1	1	0	10	01	10
310	71	1DC5B7F	7E92D000	1E92CE4E2	3961E4DCD4	1	1	1	0	1	11	10	01
311	72	1B8B6FF	7D25A001	1D259C9C4	72C3C9B9A9	1	0	1	1	0	01	11	10
312	73	1716DFF	7A4B4002	1A4B39389	6587937352	0	0	1	1	1	10	01	11
313	74	0E2DBFF	74968005	149672713	4B0F26E6A5	1	1	0	0	0	11	10	01
314	75	1C5B7FE	692D000B	092CE4E26	161E4DCD4B	1	0	1	0	1	00	11	10
315	76	18B6FFC	525A0017	1259C9C4D	2C3C9B9A96	1	0	0	0	1	10	00	11
316	77	116DFF8	24B4002F	04B39389B	587937352C	0	1	0	0	1	11	10	00
317	78	02DBFF1	4968005F	096727136	30F26E6A58	0	0	1	1	1	00	11	10
318	79	05B7FE3	12D000BF	12CE4E26C	61E4DCD4B1	0	1	0	1	0	11	00	11
319	80	0B6FFC7	25A0017F	059C9C4D8	43C9B9A963	1	1	0	1	0	00	11	00
320	81	16DFF8E	4B4002FF	0B39389B1	07937352C6	0	0	1	1	0	11	00	11
321	82	0DBFF1C	168005FF	167271363	0F26E6A58C	1	1	0	0	1	01	11	00
322	83	1B7FE38	2D000BFF	0CE4E26C7	1E4DCD4B18	1	0	1	0	1	10	01	11
323	84	16FFC70	5A0017FF	19C9C4D8F	3C9B9A9631	0	0	1	1	0	11	10	01
324	85	0DFF8E1	34002FFF	139389B1E	7937352C62	1	0	0	0	0	00	11	10
325	86	1BFF1C3	68005FFF	07271363D	726E6A58C4	1	0	0	0	1	10	00	11
326	87	17FE387	5000BFFE	0E4E26C7B	64DCD4B188	0	0	1	1	0	00	10	00
327	88	0FFC70F	20017FFD	1C9C4D8F6	49B9A96311	1	0	1	1	1	00	00	10
328	89	1FF8E1F	4002FFFB	19389B1ED	137352C623	1	0	1	0	0	01	00	00
329	90	1FF1C3F	0005FFF7	1271363DB	26E6A58C46	1	0	0	1	1	00	01	00
330	91	1FE387F	000BFFEE	04E26C7B6	4DCD4B188C	1	0	0	1	0	10	00	01
331	92	1FC70FF	0017FFDC	09C4D8F6D	1B9A963118	1	0	1	1	1	00	10	00
332	93	1F8E1FF	002FFFB8	1389B1EDA	37352C6231	1	0	0	0	1	01	00	10
333	94	1F1C3FF	005FFF70	071363DB4	6E6A58C462	1	0	0	0	0	00	01	00
334	95	1E387FE	00BFFEE0	0E26C7B68	5CD4B188C5	1	1	1	1	0	01	00	01
335	96	1C70FFC	017FFDC1	1C4D8F6D1	39A963118A	1	0	1	1	0	11	01	00
336	97	18E1FF9	02FFFB82	189B1EDA2	7352C62315	1	1	1	0	0	11	11	01
337	98	11C3FF2	05FFF705	11363DB45	66A58C462B	0	1	0	1	1	11	11	11
338	99	0387FE4	0BFFEE0A	026C7B68B	4D4B188C56	0	1	0	0	0	11	11	11
339	100	070FFC9	17FFDC15	04D8F6D16	1A963118AD	0	1	0	1	1	11	11	11
340	101	0E1FF92	2FFFB82B	09B1EDA2C	352C62315A	1	1	1	0	0	10	11	11

341	102	1C3FF24	5FFF7057	1363DB458	6A58C462B4	1	1	0	0	0	10	10	11
342	103	187FE48	3FFEE0AE	06C7B68B0	54B188C569	1	1	0	1	1	01	10	10
343	104	10FFC90	7FFDC15C	0D8F6D161	2963118AD2	0	1	1	0	1	01	01	10
344	105	01FF920	7FFB82B9	1B1EDA2C2	52C62315A5	0	1	1	1	0	00	01	01
345	106	03FF240	7FF70573	163DB4584	258C462B4B	0	1	0	1	0	10	00	01
346	107	07FE481	7FEE0AE6	0C7B68B08	4B188C5696	0	1	1	0	0	00	10	00
347	108	0FFC902	7FDC15CD	18F6D1610	163118AD2D	1	1	1	0	1	00	00	10
348	109	1FF9204	7FB82B9A	11EDA2C20	2C62315A5B	1	1	0	0	0	01	00	00
349	110	1FF2408	7F705735	03DB45841	58C462B4B6	1	0	0	1	1	00	01	00
350	111	1FE4810	7EE0AE6B	07B68B082	3188C5696C	1	1	0	1	1	10	00	01
351	112	1FC9021	7DC15CD6	0F6D16105	63118AD2D8	1	1	1	0	1	00	10	00
352	113	1F92042	7B82B9AD	1EDA2C20B	462315A5B0	1	1	1	0	1	00	00	10
353	114	1F24084	7705735A	1DB458416	0C462B4B61	1	0	1	0	0	01	00	00
354	115	1E48108	6E0AE6B5	1B68B082C	188C5696C3	1	0	1	1	0	11	01	00
355	116	1C90211	5C15CD6A	16D161059	3118AD2D86	1	0	0	0	0	10	11	01
356	117	1920422	382B9AD5	0DA2C20B3	62315A5B0D	1	0	1	0	0	10	10	11
357	118	1240845	705735AA	1B4584167	4462B4B61A	0	0	1	0	1	10	10	10
358	119	048108A	60AE6B55	168B082CF	08C5696C34	0	1	0	1	0	01	10	10
359	120	0902114	415CD6AB	0D161059E	118AD2D869	1	0	1	1	0	10	01	10
360	121	1204228	02B9AD56	1A2C20B3D	2315A5B0D2	0	1	1	0	0	11	10	01
361	122	0408451	05735AAD	14584167B	462B4B61A4	0	0	0	0	1	11	11	10
362	123	08108A2	0AE6B55B	08B082CF7	0C5696C348	1	1	1	0	0	10	11	11
363	124	1021144	15CD6AB6	1161059EF	18AD2D8690	0	1	0	1	0	10	10	11
364	125	0042289	2B9AD56C	02C20B3DE	315A5B0D20	0	1	0	0	1	10	10	10

1.5 FOURTH SET OF SAMPLES

Initial values for the key, pan address and clock

K'c4[0] = 21 K'c4[1] = 87 K'c4[2] = F0 K'c4[3] = 4A
K'c4[4] = BA K'c4[5] = 90 K'c4[6] = 31 K'c4[7] = D0
K'c4[8] = 78 K'c4[9] = 0D K'c4[10] = 4C K'c4[11] = 53
K'c4[12] = E0 K'c4[13] = 15 K'c4[14] = 3A K'c4[15] = 63

Addr4[0] = 2C Addr4[1] = 7F Addr4[2] = 94
Addr4[3] = 56 Addr4[4] = 0F Addr4[5] = 1B

CL4[0] = 5F CL4[1] = 1A CL4[2] = 00 CL4[3] = 02

Fill LFSRs with initial data

t	clk#	LFSR1	LFSR2	LFSR3	LFSR4	X1	X2	X3	X4	Z	C[t+1]	C[t]	C[t-1]
0	0	0000000*	00000000*	000000000*	0000000000*	0	0	0	0	0	00	00	00
1	1	0000000*	00000001*	000000001*	0000000001*	0	0	0	0	0	00	00	00
2	2	0000001*	00000002*	000000002*	0000000003*	0	0	0	0	0	00	00	00
3	3	0000002*	00000004*	000000004*	0000000007*	0	0	0	0	0	00	00	00
4	4	0000004*	00000009*	000000008*	000000000F*	0	0	0	0	0	00	00	00
5	5	0000008*	00000013*	000000010*	000000001E*	0	0	0	0	0	00	00	00
6	6	0000010*	00000027*	000000021*	000000003D*	0	0	0	0	0	00	00	00
7	7	0000021*	0000004F*	000000043*	000000007A*	0	0	0	0	0	00	00	00
8	8	0000042*	0000009F*	000000087*	00000000F4*	0	0	0	0	0	00	00	00
9	9	0000084*	0000013F*	00000010F*	00000001E9*	0	0	0	0	0	00	00	00
10	10	0000108*	0000027F*	00000021F*	00000003D2*	0	0	0	0	0	00	00	00
11	11	0000211*	000004FE*	00000043E*	00000007A5*	0	0	0	0	0	00	00	00
12	12	0000422*	000009FC*	00000087C*	0000000F4A*	0	0	0	0	0	00	00	00
13	13	0000845*	000013F8*	0000010F8*	0000001E94*	0	0	0	0	0	00	00	00
14	14	000108B*	000027F0*	0000021F1*	0000003D29*	0	0	0	0	0	00	00	00
15	15	0002117*	00004FE1*	0000043E3*	0000007A52*	0	0	0	0	0	00	00	00
16	16	000422E*	00009FC2*	0000087C6*	000000F4A4*	0	0	0	0	0	00	00	00
17	17	000845D*	00013F84*	000010F8C*	000001E948*	0	0	0	0	0	00	00	00
18	18	00108BA*	00027F08*	000021F18*	000003D290*	0	0	0	0	0	00	00	00
19	19	0021174*	0004FE10*	000043E30*	000007A520*	0	0	0	0	0	00	00	00
20	20	00422E8*	0009FC21*	000087C61*	00000F4A41*	0	0	0	0	0	00	00	00
21	21	00845D1*	0013F842*	00010F8C3*	00001E9482*	0	0	0	0	0	00	00	00
22	22	0108BA3*	0027F084*	00021F186*	00003D2905*	0	0	0	0	0	00	00	00
23	23	0211747*	004FE109*	00043E30C*	00007A520B*	0	0	0	0	0	00	00	00
24	24	0422E8F*	009FC213*	00087C619*	0000F4A417*	0	1	0	0	1	00	00	00
25	25	0845D1E*	013F8426*	0010F8C32*	0001E9482F*	1	0	0	0	1	00	00	00
26	26	108BA3D	027F084D*	0021F1864*	0003D2905E*	0	0	0	0	0	00	00	00
27	27	011747B	04FE109B*	0043E30C9*	0007A520BC*	0	1	0	0	1	00	00	00
28	28	022E8F6	09FC2136*	0087C6192*	000F4A4179*	0	1	0	0	1	00	00	00
29	29	045D1EC	13F8426C*	010F8C325*	001E9482F2*	0	1	0	0	1	00	00	00
30	30	08BA3D9	27F084D8*	021F1864B*	003D2905E5*	1	1	0	0	0	01	00	00
31	31	11747B3	4FE109B0*	043E30C97*	007A520BCA*	0	1	0	0	1	00	00	00
32	32	02E8F67	1FC21360	087C6192E*	00F4A41795*	0	1	1	1	1	01	00	00
33	33	05D1ECF	3F8426C1	10F8C325C*	01E9482F2B*	0	1	0	1	0	01	00	00
34	34	0BA3D9F	7F084D82	01F1864B8	03D2905E56*	1	0	0	1	0	01	00	00

35	35	1747B3E	7E109B04	03E30C970	07A520BCAC*	0	0	0	1	1	00	00	00
36	36	0E8F67C	7C213608	07C6192E1	0F4A417958*	1	0	0	0	1	00	00	00
37	37	1D1ECF8	78426C11	0F8C325C3	1E9482F2B1*	1	0	1	1	1	01	00	00
38	38	1A3D9F0	7084D822	1F1864B86	3D2905E563*	1	1	1	0	1	01	00	00
39	39	147B3E1	6109B044	1E30C970C	7A520BCAC6*	0	0	1	0	1	00	00	00

Start clocking Summation Combiner

40	1	08F67C2	42136088	1C6192E18	74A417958D	1	0	1	1	1	01	00	00
41	2	11ECF84	0426C111	18C325C30	69482F2B1B	0	0	1	0	0	00	01	00
42	3	03D9F08	084D8222	11864B861	52905E5637	0	0	0	1	1	11	00	01
43	4	07B3E10	109B0444	030C970C3	2520BCAC6E	0	1	0	0	0	01	11	00
44	5	0F67C21	21360889	06192E186	4A417958DC	1	0	0	0	0	10	01	11
45	6	1ECF843	426C1112	0C325C30C	1482F2B1B8	1	0	1	1	1	11	10	01
46	7	1D9F086	04D82225	1864B8619	2905E56370	1	1	1	0	0	01	11	10
47	8	1B3E10D	09B0444B	10C970C32	520BCAC6E1	1	1	0	0	1	10	01	11
48	9	167C21B	13608897	0192E1865	2417958DC3	0	0	0	0	0	00	10	01
49	10	0CF8436	26C1112F	0325C30CB	482F2B1B87	1	1	0	0	0	00	00	10
50	11	19F086D	4D82225E	064B86197	105E56370F	1	1	0	0	0	01	00	00
51	12	13E10DB	1B0444BC	0C970C32F	20BCAC6E1F	0	0	1	1	1	00	01	00
52	13	07C21B7	36088979	192E1865E	417958DC3F	0	0	1	0	1	11	00	01
53	14	0F8436E	6C1112F2	125C30CBD	02F2B1B87F	1	0	0	1	1	01	11	00
54	15	1F086DD	582225E4	04B86197B	05E56370FF	1	0	0	1	1	10	01	11
55	16	1E10DBA	30444BC9	0970C32F7	0BCAC6E1FF	1	0	1	1	1	11	10	01
56	17	1C21B75	60889793	12E1865EE	17958DC3FF	1	1	0	1	0	01	11	10
57	18	18436EA	41112F27	05C30CBDD	2F2B1B87FF	1	0	0	0	0	10	01	11
58	19	1086DD4	02225E4E	0B86197BA	5E56370FFF	0	0	1	0	1	00	10	01
59	20	010DBA8	0444BC9D	170C32F74	3CAC6E1FFF	0	0	0	1	1	01	00	10
60	21	021B750	0889793A	0E1865EE8	7958DC3FFF	0	1	1	0	1	00	01	00
61	22	0436EA0	1112F274	1C30CBDD0	72B1B87FFE	0	0	1	1	0	10	00	01
62	23	086DD40	2225E4E9	186197BA1	656370FFFC	1	0	1	0	0	00	10	00
63	24	10DBA81	444BC9D3	10C32F743	4AC6E1FFF8	0	0	0	1	1	01	00	10
64	25	01B7502	089793A7	01865EE86	158DC3FFF1	0	1	0	1	1	00	01	00
65	26	036EA05	112F274E	030CBDD0D	2B1B87FFE3	0	0	0	0	0	11	00	01
66	27	06DD40B	225E4E9C	06197BA1A	56370FFFC6	0	0	0	0	1	10	11	00
67	28	0DBA817	44BC9D39	0C32F7434	2C6E1FFF8D	1	1	1	0	1	10	10	11
68	29	1B7502E	09793A72	1865EE868	58DC3FFF1B	1	0	1	1	1	01	10	10
69	30	16EA05D	12F274E5	10CBDD0D0	31B87FFE36	0	1	0	1	1	01	01	10
70	31	0DD40BA	25E4E9CB	0197BA1A1	6370FFFC6D	1	1	0	0	1	11	01	01
71	32	1BA8174	4BC9D397	032F74343	46E1FFF8DA	1	1	0	1	0	11	11	01
72	33	17502E8	1793A72F	065EE8687	0DC3FFF1B4	0	1	0	1	1	11	11	11
73	34	0EA05D0	2F274E5E	0CBDD0D0F	1B87FFE369	1	0	1	1	0	10	11	11
74	35	1D40BA0	5E4E9CBD	197BA1A1F	370FFFC6D2	1	0	1	0	0	10	10	11
75	36	1A81741	3C9D397B	12F74343F	6E1FFF8DA5	1	1	0	0	0	01	10	10
76	37	1502E82	793A72F6	05EE8687F	5C3FFF1B4B	0	0	0	0	1	00	01	10
77	38	0A05D05	7274E5ED	0BDD0D0FF	387FFE3696	1	0	1	0	0	10	00	01
78	39	140BA0B	64E9CBDA	17BA1A1FF	70FFFC6D2C	0	1	0	1	0	00	10	00
79	40	0817416	49D397B4	0F74343FE	61FFF8DA59	1	1	1	1	0	11	00	10
80	41	102E82C	13A72F69	1EE8687FD	43FFF1B4B3	0	1	1	1	0	00	11	00
81	42	005D058	274E5ED2	1DD0D0FFA	07FFE36966	0	0	1	1	0	11	00	11
82	43	00BA0B0	4E9CBDA5	1BA1A1FF5	0FFFC6D2CD	0	1	1	1	0	00	11	00
83	44	0174160	1D397B4A	174343FEA	1FFF8DA59B	0	0	0	1	1	10	00	11
84	45	02E82C0	3A72F695	0E8687FD4	3FFF1B4B37	0	0	1	1	0	00	10	00
85	46	05D0580	74E5ED2B	1D0D0FFA9	7FFE36966E	0	1	1	1	1	00	00	10
86	47	0BA0B00	69CBDA56	1A1A1FF53	7FFC6D2CDC	1	1	1	1	0	10	00	00
87	48	1741600	5397B4AC	14343FEA6	7FF8DA59B8	0	1	0	1	0	00	10	00
88	49	0E82C01	272F6959	08687FD4D	7FF1B4B370	1	0	1	1	1	00	00	10

89	50	1D05802	4E5ED2B3	10D0FFA9A	7FE36966E0	1	0	0	1	0	01	00	00
90	51	1A0B004	1CBDA566	01A1FF535	7FC6D2CDC0	1	1	0	1	0	11	01	00
91	52	1416009	397B4ACC	0343FEA6B	7F8DA59B80	0	0	0	1	0	10	11	01
92	53	082C013	72F69599	0687FD4D7	7F1B4B3701	1	1	0	0	0	10	10	11
93	54	1058026	65ED2B33	0D0FFA9AF	7E36966E03	0	1	1	0	0	01	10	10
94	55	00B004D	4BDA5667	1A1FF535E	7C6D2CDC06	0	1	1	0	1	01	01	10
95	56	016009B	17B4ACCE	143FEA6BD	78DA59B80D	0	1	0	1	1	11	01	01
96	57	02C0137	2F69599D	087FD4D7B	71B4B3701A	0	0	1	1	1	10	11	01
97	58	058026F	5ED2B33B	10FFA9AF6	636966E034	0	1	0	0	1	01	10	11
98	59	0B004DF	3DA56677	01FF535ED	46D2CDC068	1	1	0	1	0	10	01	10
99	60	16009BF	7B4ACCEF	03FEA6BDB	0DA59B80D0	0	0	0	1	1	00	10	01
100	61	0C0137F	769599DF	07FD4D7B7	1B4B3701A1	1	1	0	0	0	00	00	10
101	62	18026FE	6D2B33BE	0FFA9AF6E	36966E0342	1	0	1	1	1	01	00	00
102	63	1004DFC	5A56677D	1FF535EDD	6D2CDC0684	0	0	1	0	0	00	01	00
103	64	0009BF9	34ACCEFB	1FEA6BDBB	5A59B80D09	0	1	1	0	0	10	00	01
104	65	00137F2	69599DF7	1FD4D7B76	34B3701A12	0	0	1	1	0	00	10	00
105	66	0026FE5	52B33BEF	1FA9AF6EC	6966E03424	0	1	1	0	0	00	00	10
106	67	004DFCA	256677DF	1F535EDD8	52CDC06848	0	0	1	1	0	01	00	00
107	68	009BF94	4ACCEFBE	1EA6BDBB0	259B80D091	0	1	1	1	0	11	01	00
108	69	0137F29	1599DF7C	1D4D7B760	4B3701A123	0	1	1	0	1	10	11	01
109	70	026FE53	2B33BEF9	1A9AF6EC0	166E034246	0	0	1	0	1	01	10	11
110	71	04DFCA7	56677DF2	1535EDD81	2CDC06848D	0	0	0	1	0	01	01	10
111	72	09BF94F	2CCEFBE4	0A6BDBB03	59B80D091B	1	1	1	1	1	00	01	01
112	73	137F29E	599DF7C9	14D7B7607	33701A1236	0	1	0	0	1	11	00	01
113	74	06FE53C	333BEF93	09AF6EC0E	66E034246C	0	0	1	1	1	01	11	00
114	75	0DFCA79	6677DF26	135EDD81D	4DC06848D8	1	0	0	1	1	10	01	11
115	76	1BF94F2	4CEFBE4D	06BDBB03B	1B80D091B1	1	1	0	1	1	11	10	01
116	77	17F29E5	19DF7C9A	0D7B76077	3701A12363	0	1	1	0	1	00	11	10
117	78	0FE53CA	33BEF934	1AF6EC0EF	6E034246C6	1	1	1	0	1	11	00	11
118	79	1FCA794	677DF269	15EDD81DF	5C06848D8C	1	0	0	0	0	01	11	00
119	80	1F94F29	4EFBE4D2	0BDBB03BE	380D091B19	1	1	1	0	0	01	01	11
120	81	1F29E53	1DF7C9A5	17B76077D	701A123633	1	1	0	0	1	11	01	01
121	82	1E53CA6	3BEF934B	0F6EC0EFB	6034246C66	1	1	1	0	0	11	11	01
122	83	1CA794D	77DF2696	1EDD81DF6	406848D8CD	1	1	1	0	0	10	11	11
123	84	194F29B	6FBE4D2C	1DBB03BED	00D091B19B	1	1	1	1	0	11	10	11
124	85	129E536	5F7C9A59	1B76077DA	01A1236337	0	0	1	1	1	00	11	10
125	86	053CA6C	3EF934B3	16EC0EFB4	034246C66E	0	1	0	0	1	10	00	11
126	87	0A794D9	7DF26967	0DD81DF69	06848D8CDD	1	1	1	1	0	01	10	00
127	88	14F29B3	7BE4D2CF	1BB03BED3	0D091B19BB	0	1	1	0	1	01	01	10
128	89	09E5366	77C9A59F	176077DA6	1A12363377	1	1	0	0	1	11	01	01
129	90	13CA6CD	6F934B3F	0EC0EFB4D	34246C66EF	0	1	1	0	1	10	11	01
130	91	0794D9B	5F26967F	1D81DF69A	6848D8CDDF	0	0	1	0	1	01	10	11
131	92	0F29B37	3E4D2CFE	1B03BED35	5091B19BBE	1	0	1	1	0	10	01	10
132	93	1E5366F	7C9A59FD	16077DA6B	212363377C	1	1	0	0	0	11	10	01
133	94	1CA6CDF	7934B3FB	0C0EFB4D6	4246C66EF9	1	0	1	0	1	00	11	10
134	95	194D9BE	726967F6	181DF69AD	048D8CDDF2	1	0	1	1	1	11	00	11
135	96	129B37D	64D2CFED	103BED35B	091B19BBE5	0	1	0	0	0	01	11	00
136	97	05366FA	49A59FDA	0077DA6B7	12363377CA	0	1	0	0	0	10	01	11
137	98	0A6CDF5	134B3FB4	00EFB4D6E	246C66EF95	1	0	0	0	1	00	10	01
138	99	14D9BEA	26967F69	01DF69ADD	48D8CDDF2B	0	1	0	1	0	00	00	10
139	100	09B37D4	4D2CFED2	03BED35BB	11B19BBE56	1	0	0	1	0	01	00	00
140	101	1366FA8	1A59FDA5	077DA6B77	2363377CAC	0	0	0	0	1	01	01	00
141	102	06CDF51	34B3FB4A	0EFB4D6EF	46C66EF959	0	1	1	1	0	00	01	01
142	103	0D9BEA2	6967F695	1DF69ADDF	0D8CDDF2B2	1	0	1	1	1	10	00	01
143	104	1B37D45	52CFED2A	1BED35BBF	1B19BBE564	1	1	1	0	1	00	10	00
144	105	166FA8A	259FDA54	17DA6B77E	363377CAC8	0	1	0	0	1	01	00	10
145	106	0CDF515	4B3FB4A9	0FB4D6EFC	6C66EF9591	1	0	1	0	1	00	01	00

146	107	19BEA2B	167F6952	1F69ADDF8	58CDDF2B22	1	0	1	1	1	10	00	01
147	108	137D457	2CFED2A5	1ED35BBF1	319BBE5645	0	1	1	1	1	00	10	00
148	109	06FA8AF	59FDA54A	1DA6B77E2	63377CAC8B	0	1	1	0	0	00	00	10
149	110	0DF515F	33FB4A95	1B4D6EFC4	466EF95916	1	1	1	0	1	01	00	00
150	111	1BEA2BF	67F6952A	169ADDF88	0CDDF2B22C	1	1	0	1	0	11	01	00
151	112	17D457F	4FED2A55	0D35BBF10	19BBE56459	0	1	1	1	0	11	11	01
152	113	0FA8AFE	1FDA54AB	1A6B77E20	3377CAC8B3	1	1	1	0	0	10	11	11
153	114	1F515FD	3FB4A957	14D6EFC40	66EF959166	1	1	0	1	1	10	10	11
154	115	1EA2BFA	7F6952AF	09ADDF880	4DDF2B22CC	1	0	1	1	1	01	10	10
155	116	1D457F4	7ED2A55F	135BBF100	1BBE564598	1	1	0	1	0	10	01	10
156	117	1A8AFE8	7DA54ABF	06B77E200	377CAC8B31	1	1	0	0	0	11	10	01
157	118	1515FD0	7B4A957F	0D6EFC401	6EF9591663	0	0	1	1	1	00	11	10
158	119	0A2BFA1	76952AFE	1ADDF8803	5DF2B22CC7	1	1	1	1	0	00	00	11
159	120	1457F42	6D2A55FD	15BBF1007	3BE564598E	0	0	0	1	1	00	00	00
160	121	08AFE84	5A54ABFB	0B77E200F	77CAC8B31C	1	0	1	1	1	01	00	00
161	122	115FD09	34A957F7	16EFC401F	6F95916639	0	1	0	1	1	00	01	00
162	123	02BFA12	6952AFEF	0DDF8803E	5F2B22CC73	0	0	1	0	1	11	00	01
163	124	057F424	52A55FDF	1BBF1007D	3E564598E7	0	1	1	0	1	01	11	00
164	125	0AFE848	254ABFBF	177E200FA	7CAC8B31CF	1	0	0	1	1	10	01	11
165	126	15FD090	4A957F7E	0EFC401F5	795916639E	0	1	1	0	0	11	10	01
166	127	0BFA121	152AFEFD	1DF8803EA	72B22CC73C	1	0	1	1	0	01	11	10
167	128	17F4243	2A55FDFA	1BF1007D4	6564598E78	0	0	1	0	0	10	01	11
168	129	0FE8486	54ABFBF4	17E200FA8	4AC8B31CF0	1	1	0	1	1	11	10	01
169	130	1FD090C	2957F7E8	0FC401F51	15916639E1	1	0	1	1	0	01	11	10
170	131	1FA1219	52AFEFD1	1F8803EA3	2B22CC73C2	1	1	1	0	0	01	01	11
171	132	1F42432	255FDFA2	1F1007D47	564598E785	1	0	1	0	1	11	01	01
172	133	1E84865	4ABFBF44	1E200FA8F	2C8B31CF0B	1	1	1	1	1	11	11	01
173	134	1D090CB	157F7E88	1C401F51E	5916639E17	1	0	1	0	1	11	11	11
174	135	1A12196	2AFEFD11	18803EA3C	322CC73C2E	1	1	1	0	0	10	11	11
175	136	142432C	55FDFA23	11007D479	64598E785C	0	1	0	0	1	01	10	11
176	137	0848659	2BFBF446	0200FA8F2	48B31CF0B9	1	1	0	1	0	10	01	10
177	138	1090CB2	57F7E88C	0401F51E4	116639E173	0	1	0	0	1	00	10	01
178	139	0121964	2FEFD118	0803EA3C8	22CC73C2E6	0	1	1	1	1	00	00	10
179	140	02432C9	5FDFA230	1007D4791	4598E785CD	0	1	0	1	0	01	00	00
180	141	0486593	3FBF4461	000FA8F23	0B31CF0B9B	0	1	0	0	0	00	01	00
181	142	090CB26	7F7E88C3	001F51E47	16639E1736	1	0	0	0	1	11	00	01
182	143	121964D	7EFD1187	003EA3C8F	2CC73C2E6C	0	1	0	1	1	01	11	00
183	144	0432C9B	7DFA230E	007D4791E	598E785CD8	0	1	0	1	1	10	01	11
184	145	0865936	7BF4461C	00FA8F23C	331CF0B9B0	1	1	0	0	0	11	10	01
185	146	10CB26D	77E88C38	01F51E479	6639E17361	0	1	0	0	0	00	11	10
186	147	01964DA	6FD11870	03EA3C8F2	4C73C2E6C2	0	1	0	0	1	10	00	11
187	148	032C9B4	5FA230E1	07D4791E4	18E785CD84	0	1	0	1	0	00	10	00
188	149	0659368	3F4461C2	0FA8F23C9	31CF0B9B09	0	0	1	1	0	00	00	10
189	150	0CB26D0	7E88C384	1F51E4793	639E173612	1	1	1	1	0	10	00	00
190	151	1964DA0	7D118709	1EA3C8F27	473C2E6C24	1	0	1	0	0	00	10	00
191	152	12C9B41	7A230E12	1D4791E4E	0E785CD848	0	0	1	0	1	01	00	10
192	153	0593683	74461C24	1A8F23C9C	1CF0B9B091	0	0	1	1	1	00	01	00
193	154	0B26D06	688C3848	151E47938	39E1736123	1	1	0	1	1	10	00	01
194	155	164DA0D	51187091	0A3C8F271	73C2E6C247	0	0	1	1	0	00	10	00
195	156	0C9B41A	2230E123	14791E4E3	6785CD848F	1	0	0	1	0	00	00	10
196	157	1936835	4461C247	08F23C9C6	4F0B9B091E	1	0	1	0	0	01	00	00
197	158	126D06A	08C3848E	11E47938D	1E1736123C	0	1	0	0	0	00	01	00
198	159	04DA0D5	1187091C	03C8F271B	3C2E6C2478	0	1	0	0	1	11	00	01
199	160	09B41AA	230E1238	0791E4E37	785CD848F1	1	0	0	0	0	01	11	00
200	161	1368354	461C2470	0F23C9C6F	70B9B091E3	0	0	1	1	1	10	01	11
201	162	06D06A9	0C3848E1	1E47938DF	61736123C6	0	0	1	0	1	00	10	01
202	163	0DA0D52	187091C3	1C8F271BE	42E6C2478D	1	0	1	1	1	00	00	10

203	164	1B41AA4	30E12387	191E4E37C	05CD848F1A	1	1	1	1	0	10	00	00
204	165	1683549	61C2470F	123C9C6F9	0B9B091E34	0	1	0	1	0	00	10	00
205	166	0D06A92	43848E1E	047938DF3	1736123C68	1	1	0	0	0	00	00	10
206	167	1A0D524	07091C3C	08F271BE7	2E6C2478D1	1	0	1	0	0	01	00	00
207	168	141AA49	0E123879	11E4E37CF	5CD848F1A2	0	0	0	1	0	00	01	00
208	169	0835492	1C2470F3	03C9C6F9F	39B091E345	1	0	0	1	0	10	00	01
209	170	106A925	3848E1E6	07938DF3F	736123C68B	0	0	0	0	0	11	10	00
210	171	00D524A	7091C3CD	0F271BE7E	66C2478D16	0	1	1	1	0	01	11	10
211	172	01AA495	6123879B	1E4E37CFD	4D848F1A2D	0	0	1	1	1	10	01	11
212	173	035492A	42470F36	1C9C6F9FB	1B091E345B	0	0	1	0	1	00	10	01
213	174	06A9255	048E1E6C	1938DF3F6	36123C68B7	0	1	1	0	0	00	00	10
214	175	0D524AB	091C3CD8	1271BE7EC	6C2478D16E	1	0	0	0	1	00	00	00
215	176	1AA4957	123879B1	04E37CFD8	5848F1A2DD	1	0	0	0	1	00	00	00
216	177	15492AF	2470F363	09C6F9FB0	3091E345BA	0	0	1	1	0	01	00	00
217	178	0A9255E	48E1E6C7	138DF3F61	6123C68B75	1	1	0	0	1	00	01	00
218	179	1524ABD	11C3CD8F	071BE7EC3	42478D16EB	0	1	0	0	1	11	00	01
219	180	0A4957B	23879B1F	0E37CFD87	048F1A2DD6	1	1	1	1	1	00	11	00
220	181	1492AF6	470F363F	1C6F9FB0E	091E345BAD	0	0	1	0	1	10	00	11
221	182	09255EC	0E1E6C7F	18DF3F61D	123C68B75B	1	0	1	0	0	00	10	00
222	183	124ABD9	1C3CD8FF	11BE7EC3A	2478D16EB6	0	0	0	0	0	01	00	10
223	184	04957B3	3879B1FE	037CFD874	48F1A2DD6D	0	0	0	1	0	00	01	00
224	185	092AF66	70F363FD	06F9FB0E9	11E345BADB	1	1	0	1	1	10	00	01
225	186	1255ECD	61E6C7FA	0DF3F61D3	23C68B75B7	0	1	1	1	1	00	10	00
226	187	04ABD9B	43CD8FF5	1BE7EC3A7	478D16EB6E	0	1	1	1	1	00	00	10
227	188	0957B37	079B1FEA	17CFD874E	0F1A2DD6DD	1	1	0	0	0	01	00	00
228	189	12AF66F	0F363FD4	0F9FB0E9C	1E345BADBB	0	0	1	0	0	00	01	00
229	190	055ECDE	1E6C7FA9	1F3F61D39	3C68B75B76	0	0	1	0	1	11	00	01
230	191	0ABD9BC	3CD8FF53	1E7EC3A73	78D16EB6EC	1	1	1	1	1	00	11	00
231	192	157B379	79B1FEA7	1CFD874E6	71A2DD6DD9	0	1	1	1	1	11	00	11
232	193	0AF66F3	7363FD4E	19FB0E9CD	6345BADBB2	1	0	1	0	1	01	11	00
233	194	15ECDE6	66C7FA9D	13F61D39A	468B75B765	0	1	0	1	1	10	01	11
234	195	0BD9BCC	4D8FF53A	07EC3A735	0D16EB6ECA	1	1	0	0	0	11	10	01
235	196	17B3799	1B1FEA75	0FD874E6A	1A2DD6DD94	0	0	1	0	0	00	11	10
236	197	0F66F33	363FD4EA	1FB0E9CD5	345BADBB28	1	0	1	0	0	11	00	11
237	198	1ECDE67	6C7FA9D5	1F61D39AA	68B75B7650	1	0	1	1	0	00	11	00
238	199	1D9BCCF	58FF53AB	1EC3A7354	516EB6ECA0	1	1	1	0	1	11	00	11
239	200	1B3799E	31FEA756	1D874E6A8	22DD6DD940	1	1	1	1	1	00	11	00

```
Z[0]  = 3F
Z[1]  = B1
Z[2]  = 67
Z[3]  = D2
Z[4]  = 2F
Z[5]  = A6
Z[6]  = 1F
Z[7]  = B9
Z[8]  = E6
Z[9]  = 84
Z[10] = 43
Z[11] = 07
Z[12] = D8
Z[13] = 1E
Z[14] = E7
Z[15] = C3
```

```
================================================================================================
         Reload this pattern into the LFSRs
         Hold content of Summation Combiner regs and calculate new C[t+1] and Z values
================================================================================================
LFSR1  <= 0E62F3F
LFSR2  <= 6C84A6B1
LFSR3  <= 11E431F67
LFSR4  <= 61E707B9D2
C[t+1] <= 00

================================================================================================
         Generating 125 key symbols (encryption/decryption sequence)
================================================================================================
240     1    0E62F3F  6C84A6B1  11E431F67  61E707B9D2      1  1  0  1    0  00     11   00
241     2    1CC5E7F  59094D63  03C863ECE  43CE0F73A5      1  0  0  1    0  11     00   11
242     3    198BCFF  32129AC6  0790C7D9D  079C1EE74A      1  0  0  1    1  01     11   00
243     4    13179FE  6425358C  0F218FB3A  0F383DCE94      0  0  1  0    0  10     01   11
244     5    062F3FD  484A6B19  1E431F675  1E707B9D28      0  0  1  0    1  00     10   01
245     6    0C5E7FB  1094D632  1C863ECEB  3CE0F73A50      1  1  1  1    0  11     00   10
246     7    18BCFF7  2129AC64  190C7D9D7  79C1EE74A1      1  0  1  1    0  00     11   00
247     8    1179FEE  425358C8  1218FB3AE  7383DCE942      0  0  0  1    1  10     00   11
248     9    02F3FDD  04A6B190  0431F675D  6707B9D285      0  1  0  0    1  11     10   00
249    10    05E7FBB  094D6320  0863ECEBB  4E0F73A50B      0  0  1  0    0  00     11   10
250    11    0BCFF77  129AC640  10C7D9D77  1C1EE74A16      1  1  0  0    0  11     00   11
251    12    179FEEE  25358C80  018FB3AEE  383DCE942C      0  0  0  0    1  10     11   00
252    13    0F3FDDC  4A6B1900  031F675DD  707B9D2859      1  0  0  0    1  01     10   11
253    14    1E7FBB8  14D63200  063ECEBBA  60F73A50B3      1  1  0  1    0  10     01   10
254    15    1CFF771  29AC6401  0C7D9D774  41EE74A167      1  1  1  1    0  10     10   01
255    16    19FEEE2  5358C803  18FB3AEE9  03DCE942CE      1  0  1  1    1  01     10   10
256    17    13FDDC4  26B19007  11F675DD2  07B9D2859C      0  1  0  1    1  01     01   10
257    18    07FBB88  4D63200E  03ECEBBA4  0F73A50B38      0  0  0  0    1  10     01   01
258    19    0FF7711  1AC6401D  07D9D7748  1EE74A1670      1  1  0  1    1  11     10   01
259    20    1FEEE23  358C803B  0FB3AEE91  3DCE942CE1      1  1  1  1    1  01     11   10
260    21    1FDDC47  6B190076  1F675DD23  7B9D2859C2      1  0  1  1    0  01     01   11
261    22    1FBB88F  563200ED  1ECEBBA47  773A50B385      1  0  1  0    1  11     01   01
262    23    1F7711E  2C6401DB  1D9D7748F  6E74A1670A      1  0  1  0    1  10     11   01
263    24    1EEE23D  58C803B6  1B3AEE91E  5CE942CE15      1  1  1  1    0  11     10   11
264    25    1DDC47A  3190076C  1675DD23D  39D2859C2B      1  1  0  1    0  01     11   10
265    26    1BB88F4  63200ED9  0CEBBA47A  73A50B3856      1  0  1  1    0  01     01   11
266    27    17711E8  46401DB2  19D7748F5  674A1670AD      0  0  1  0    0  11     01   01
267    28    0EE23D0  0C803B64  13AEE91EA  4E942CE15B      1  1  0  1    0  11     11   01
268    29    1DC47A0  190076C8  075DD23D4  1D2859C2B7      1  0  0  0    0  11     11   11
269    30    1B88F41  3200ED90  0EBBA47A9  3A50B3856E      1  0  1  0    1  11     11   11
270    31    1711E83  6401DB20  1D7748F53  74A1670ADC      0  0  1  1    1  11     11   11
271    32    0E23D07  4803B641  1AEE91EA7  6942CE15B8      1  0  1  0    1  11     11   11
272    33    1C47A0F  10076C82  15DD23D4F  52859C2B71      1  0  0  1    1  11     11   11
273    34    188F41E  200ED905  0BBA47A9E  250B3856E3      1  0  1  0    1  11     11   11
274    35    111E83C  401DB20A  17748F53D  4A1670ADC7      0  0  0  0    1  00     11   11
275    36    023D078  003B6414  0EE91EA7A  142CE15B8E      0  0  1  0    1  10     00   11
276    37    047A0F0  0076C828  1DD23D4F5  2859C2B71C      0  0  1  0    1  11     10   00
277    38    08F41E1  00ED9050  1BA47A9EA  50B3856E39      1  1  1  1    1  01     11   10
278    39    11E83C2  01DB20A0  1748F53D5  21670ADC72      0  1  0  0    0  10     01   11
279    40    03D0785  03B64141  0E91EA7AA  42CE15B8E4      0  1  1  1    1  11     10   01
280    41    07A0F0A  076C8283  1D23D4F54  059C2B71C8      0  0  1  1    1  00     11   10
281    42    0F41E14  0ED90507  1A47A9EA9  0B3856E390      1  1  1  0    1  11     00   11
282    43    1E83C29  1DB20A0F  148F53D52  1670ADC720      1  1  0  0    1  01     11   00
283    44    1D07853  3B64141E  091EA7AA5  2CE15B8E40      1  0  1  1    0  01     01   11
```

284	45	1A0F0A6	76C8283C	123D4F54B	59C2B71C81	1	1	0	1	0	00	01	01
285	46	141E14C	6D905079	047A9EA97	33856E3902	0	1	0	1	0	10	00	01
286	47	083C299	5B20A0F2	08F53D52F	670ADC7204	1	0	1	0	0	00	10	00
287	48	1078533	364141E4	11EA7AA5E	4E15B8E408	0	0	0	0	0	01	00	10
288	49	00F0A67	6C8283C8	03D4F54BC	1C2B71C811	0	1	0	0	0	00	01	00
289	50	01E14CE	59050791	07A9EA978	3856E39022	0	0	0	0	0	11	00	01
290	51	03C299C	320A0F23	0F53D52F1	70ADC72045	0	0	1	1	1	01	11	00
291	52	0785339	64141E47	1EA7AA5E2	615B8E408A	0	0	1	0	0	10	01	11
292	53	0F0A673	48283C8E	1D4F54BC4	42B71C8115	1	0	1	1	1	11	10	01
293	54	1E14CE6	1050791C	1A9EA9788	056E39022B	1	0	1	0	1	00	11	10
294	55	1C299CD	20A0F239	153D52F10	0ADC720456	1	1	0	1	1	11	00	11
295	56	185339B	4141E472	0A7AA5E20	15B8E408AC	1	0	1	1	0	00	11	00
296	57	10A6736	0283C8E4	14F54BC41	2B71C81158	0	1	0	0	1	10	00	11
297	58	014CE6C	050791C9	09EA97882	56E39022B0	0	0	1	1	0	00	10	00
298	59	0299CD9	0A0F2393	13D52F104	2DC7204561	0	0	0	1	1	01	00	10
299	60	05339B3	141E4726	07AA5E208	5B8E408AC3	0	0	0	1	0	00	01	00
300	61	0A67366	283C8E4C	0F54BC411	371C811587	1	0	1	0	0	10	00	01
301	62	14CE6CC	50791C98	1EA978822	6E39022B0F	0	0	1	0	1	11	10	00
302	63	099CD99	20F23930	1D52F1045	5C7204561E	1	1	1	0	0	01	11	10
303	64	1339B33	41E47260	1AA5E208B	38E408AC3D	0	1	1	1	0	01	01	11
304	65	0673666	03C8E4C0	154BC4117	71C811587A	0	1	0	1	1	11	01	01
305	66	0CE6CCC	0791C980	0A978822E	639022B0F5	1	1	1	1	1	11	11	01
306	67	19CD999	0F239301	152F1045C	47204561EB	1	0	0	0	0	11	11	11
307	68	139B332	1E472603	0A5E208B9	0E408AC3D6	0	0	1	0	0	11	11	11
308	69	0736664	3C8E4C06	14BC41172	1C811587AD	0	1	0	1	1	11	11	11
309	70	0E6CCC8	791C980C	0978822E5	39022B0F5A	1	0	1	0	1	11	11	11
310	71	1CD9990	72393019	12F1045CB	7204561EB4	1	0	0	0	0	11	11	11
311	72	19B3320	64726033	05E208B97	6408AC3D69	1	0	0	0	0	11	11	11
312	73	1366640	48E4C067	0BC41172F	4811587AD3	0	1	1	0	1	11	11	11
313	74	06CCC81	11C980CF	178822E5E	1022B0F5A6	0	1	0	0	0	11	11	11
314	75	0D99903	2393019E	0F1045CBC	204561EB4C	1	1	1	0	0	10	11	11
315	76	1B33206	4726033D	1E208B979	408AC3D699	1	0	1	1	1	10	10	11
316	77	166640D	0E4C067B	1C41172F2	011587AD33	0	0	1	0	1	10	10	10
317	78	0CCC81B	1C980CF6	18822E5E5	022B0F5A66	1	1	1	0	1	01	10	10
318	79	1999036	393019EC	11045CBCA	04561EB4CD	1	0	0	0	0	01	01	10
319	80	133206C	726033D9	0208B9794	08AC3D699B	0	0	0	1	0	11	01	01
320	81	06640D9	64C067B3	041172F29	11587AD337	0	1	0	0	0	10	11	01
321	82	0CC81B3	4980CF66	0822E5E53	22B0F5A66F	1	1	1	1	0	11	10	11
322	83	1990366	13019ECC	1045CBCA6	4561EB4CDF	1	0	0	0	0	00	11	10
323	84	13206CC	26033D98	008B9794D	0AC3D699BE	0	0	0	1	1	10	00	11
324	85	0640D98	4C067B31	01172F29B	1587AD337C	0	0	0	1	1	11	10	00
325	86	0C81B30	180CF662	022E5E537	2B0F5A66F9	1	0	0	0	0	00	11	10
326	87	1903660	3019ECC5	045CBCA6F	561EB4CDF3	1	0	0	0	1	10	00	11
327	88	1206CC1	6033D98A	08B9794DE	2C3D699BE6	0	0	1	0	1	11	10	00
328	89	040D983	4067B315	1172F29BD	587AD337CC	0	0	0	0	1	11	11	10
329	90	081B306	00CF662A	02E5E537A	30F5A66F98	1	1	0	1	0	10	11	11
330	91	103660C	019ECC55	05CBCA6F4	61EB4CDF31	0	1	0	1	0	10	10	11
331	92	006CC19	033D98AB	0B9794DE8	43D699BE62	0	0	1	1	0	01	10	10
332	93	00D9833	067B3156	172F29BD0	07AD337CC5	0	0	0	1	0	01	01	10
333	94	01B3066	0CF662AC	0E5E537A0	0F5A66F98B	0	1	1	0	1	11	01	01
334	95	03660CD	19ECC559	1CBCA6F41	1EB4CDF317	0	1	1	1	0	11	11	01
335	96	06CC19B	33D98AB2	19794DE83	3D699BE62F	0	1	1	0	1	11	11	11
336	97	0D98336	67B31565	12F29BD06	7AD337CC5F	1	1	0	1	0	10	11	11
337	98	1B3066D	4F662ACA	05E537A0C	75A66F98BF	1	0	0	1	0	10	10	11
338	99	1660CDB	1ECC5594	0BCA6F418	6B4CDF317E	0	1	1	0	0	01	10	10
339	100	0CC19B7	3D98AB29	1794DE831	5699BE62FC	1	1	0	1	0	10	01	10
340	101	198336F	7B315653	0F29BD062	2D337CC5F9	1	0	1	0	0	11	10	01

341	102	13066DE	7662ACA7	1E537A0C5	5A66F98BF2	0	0	1	0	0	00	11	10
342	103	060CDBC	6CC5594F	1CA6F418B	34CDF317E4	0	1	1	1	1	11	00	11
343	104	0C19B78	598AB29F	194DE8317	699BE62FC9	1	1	1	1	1	00	11	00
344	105	18336F1	3315653F	129BD062E	5337CC5F92	1	0	0	0	1	10	00	11
345	106	1066DE2	662ACA7E	0537A0C5C	266F98BF25	0	0	0	0	0	11	10	00
346	107	00CDBC5	4C5594FD	0A6F418B9	4CDF317E4B	0	0	1	1	1	00	11	10
347	108	019B78B	18AB29FA	14DE83172	19BE62FC96	0	1	0	1	0	11	00	11
348	109	0336F16	315653F4	09BD062E5	337CC5F92C	0	0	1	0	0	01	11	00
349	110	066DE2D	62ACA7E8	137A0C5CA	66F98BF258	0	1	0	1	1	10	01	11
350	111	0CDBC5B	45594FD1	06F418B95	4DF317E4B1	1	0	0	1	0	11	10	01
351	112	19B78B6	0AB29FA2	0DE83172B	1BE62FC962	1	1	1	1	1	01	11	10
352	113	136F16C	15653F45	1BD062E57	37CC5F92C5	0	0	1	1	1	10	01	11
353	114	06DE2D9	2ACA7E8B	17A0C5CAE	6F98BF258B	0	1	0	1	0	11	10	01
354	115	0DBC5B2	5594FD16	0F418B95D	5F317E4B16	1	1	1	0	0	01	11	10
355	116	1B78B64	2B29FA2C	1E83172BB	3E62FC962C	1	0	1	0	1	10	01	11
356	117	16F16C8	5653F458	1D062E577	7CC5F92C58	0	0	1	1	0	11	10	01
357	118	0DE2D91	2CA7E8B0	1A0C5CAEF	798BF258B1	1	1	1	1	1	01	11	10
358	119	1BC5B23	594FD161	1418B95DF	7317E4B163	1	0	0	0	0	10	01	11
359	120	178B647	329FA2C2	083172BBF	662FC962C7	0	1	1	0	0	11	10	01
360	121	0F16C8E	653F4584	1062E577F	4C5F92C58E	1	0	0	0	0	00	11	10
361	122	1E2D91C	4A7E8B09	00C5CAEFE	18BF258B1C	1	0	0	1	0	11	00	11
362	123	1C5B238	14FD1613	018B95DFC	317E4B1639	1	1	0	0	1	01	11	00
363	124	18B6471	29FA2C27	03172BBF9	62FC962C72	1	1	0	1	0	01	01	11
364	125	116C8E2	53F4584E	062E577F3	45F92C58E4	0	1	0	1	1	11	01	01

2 FREQUENCY HOPPING SAMPLE DATA—MANDATORY SCHEME

These sets of sample data show the mandatory frequency hopping scheme for three different combinations of addresses and initial clock values. The first part is for the 79-hop system, the second is for the 23-hop system.

2.1 THE 79-HOP SYSTEM SAMPLE DATA

2.1.1 First set

```
================================================================================
                                    79 HOP SYSTEM
================================================================================

Hop sequence {k} for PAGE SCAN/INQUIRY SCAN:
CLKN start:     0x0000000
ULAP:           0x00000000
#ticks:         0000 | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 |
                ---------------------------------------------------------
0x0000000:         0 |    2 |    4 |    6 |    8 |   10 |   12 |   14 |
0x0008000:        16 |   18 |   20 |   22 |   24 |   26 |   28 |   30 |
0x0010000:        32 |   34 |   36 |   38 |   40 |   42 |   44 |   46 |
0x0018000:        48 |   50 |   52 |   54 |   56 |   58 |   60 |   62 |
0x0020000:         0 |    2 |    4 |    6 |    8 |   10 |   12 |   14 |
0x0028000:        16 |   18 |   20 |   22 |   24 |   26 |   28 |   30 |
0x0030000:        32 |   34 |   36 |   38 |   40 |   42 |   44 |   46 |
0x0038000:        48 |   50 |   52 |   54 |   56 |   58 |   60 |   62 |

Hop sequence {k} for PAGE STATE/INQUIRY STATE:
CLKE start:     0x0000000
ULAP:           0x00000000
#ticks:         00 01 02 03 | 04 05 06 07 | 08 09 0a 0b | 0c 0d 0e 0f |
                -----------------------------------------------------------
0x0000000:      48 50 09 13 | 52 54 41 45 | 56 58 11 15 | 60 62 43 47 |
0x0000010:      00 02 64 68 | 04 06 17 21 | 08 10 66 70 | 12 14 19 23 |
0x0000020:      48 50 09 13 | 52 54 41 45 | 56 58 11 15 | 60 62 43 47 |
0x0000030:      00 02 64 68 | 04 06 17 21 | 08 10 66 70 | 12 14 19 23 |
...
0x0001000:      48 18 09 05 | 20 22 33 37 | 24 26 03 07 | 28 30 35 39 |
0x0001010:      32 34 72 76 | 36 38 25 29 | 40 42 74 78 | 44 46 27 31 |
0x0001020:      48 18 09 05 | 20 22 33 37 | 24 26 03 07 | 28 30 35 39 |
0x0001030:      32 34 72 76 | 36 38 25 29 | 40 42 74 78 | 44 46 27 31 |
...
0x0002000:      16 18 01 05 | 52 54 41 45 | 56 58 11 15 | 60 62 43 47 |
0x0002010:      00 02 64 68 | 04 06 17 21 | 08 10 66 70 | 12 14 19 23 |
0x0002020:      16 18 01 05 | 52 54 41 45 | 56 58 11 15 | 60 62 43 47 |
0x0002030:      00 02 64 68 | 04 06 17 21 | 08 10 66 70 | 12 14 19 23 |
...
0x0003000:      48 50 09 13 | 52 22 41 37 | 24 26 03 07 | 28 30 35 39 |
0x0003010:      32 34 72 76 | 36 38 25 29 | 40 42 74 78 | 44 46 27 31 |
0x0003020:      48 50 09 13 | 52 22 41 37 | 24 26 03 07 | 28 30 35 39 |
0x0003030:      32 34 72 76 | 36 38 25 29 | 40 42 74 78 | 44 46 27 31 |

Hop sequence {k} for SLAVE PAGE RESPONSE STATE:
CLKN* =         0x0000010
ULAP:           0x00000000
#ticks:         00 | 02 04 | 06 08 | 0a 0c | 0e 10 | 12 14 | 16 18 | 1a 1c | 1e
                -----------------------------------------------------------------
0x0000012:      64 | 02 68 | 04 17 | 06 21 | 08 66 | 10 70 | 12 19 | 14 23 | 16
0x0000032:      01 | 18 05 | 20 33 | 22 37 | 24 03 | 26 07 | 28 35 | 30 39 | 32
0x0000052:      72 | 34 76 | 36 25 | 38 29 | 40 74 | 42 78 | 44 27 | 46 31 | 48
0x0000072:      09 | 50 13 | 52 41 | 54 45 | 56 11 | 58 15 | 60 43 | 62 47 | 00

Hop sequence {k} for MASTER PAGE RESPONSE STATE:
Offset value:   24
CLKE* =         0x0000012
ULAP:           0x00000000
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                -----------------------------------------------------------------
0x0000014:      02 68 | 04 17 | 06 21 | 08 66 | 10 70 | 12 19 | 14 23 | 16 01 |
0x0000034:      18 05 | 20 33 | 22 37 | 24 03 | 26 07 | 28 35 | 30 39 | 32 72 |
0x0000054:      34 76 | 36 25 | 38 29 | 40 74 | 42 78 | 44 27 | 46 31 | 48 09 |
0x0000074:      50 13 | 52 41 | 54 45 | 56 11 | 58 15 | 60 43 | 62 47 | 00 64 |
```

Hop sequence {k} for CONNECTION STATE:
CLK start: 0x0000010
ULAP: 0x00000000

#ticks:	00 02	04 06	08 0a	0c 0e	10 12	14 16	18 1a	1c 1e
0x0000010:	08 66	10 70	12 19	14 23	16 01	18 05	20 33	22 37
0x0000030:	24 03	26 07	28 35	30 39	32 72	34 76	36 25	38 29
0x0000050:	40 74	42 78	44 27	46 31	48 09	50 13	52 41	54 45
0x0000070:	56 11	58 15	60 43	62 47	32 17	36 19	34 49	38 51
0x0000090:	40 21	44 23	42 53	46 55	48 33	52 35	50 65	54 67
0x00000b0:	56 37	60 39	58 69	62 71	64 25	68 27	66 57	70 59
0x00000d0:	72 29	76 31	74 61	78 63	01 41	05 43	03 73	07 75
0x00000f0:	09 45	13 47	11 77	15 00	64 49	66 53	68 02	70 06
0x0000110:	01 51	03 55	05 04	07 08	72 57	74 61	76 10	78 14
0x0000130:	09 59	11 63	13 12	15 16	17 65	19 69	21 18	23 22
0x0000150:	33 67	35 71	37 20	39 24	25 73	27 77	29 26	31 30
0x0000170:	41 75	43 00	45 28	47 32	17 02	21 04	19 34	23 36
0x0000190:	33 06	37 08	35 38	39 40	25 10	29 12	27 42	31 44
0x00001b0:	41 14	45 16	43 46	47 48	49 18	53 20	51 50	55 52
0x00001d0:	65 22	69 24	67 54	71 56	57 26	61 28	59 58	63 60
0x00001f0:	73 30	77 32	75 62	00 64	49 34	51 42	57 66	59 74
0x0000210:	53 36	55 44	61 68	63 76	65 50	67 58	73 03	75 11
0x0000230:	69 52	71 60	77 05	00 13	02 38	04 46	10 70	12 78
0x0000250:	06 40	08 48	14 72	16 01	18 54	20 62	26 07	28 15
0x0000270:	22 56	24 64	30 09	32 17	02 66	06 74	10 19	14 27
0x0000290:	04 70	08 78	12 23	16 31	18 03	22 11	26 35	30 43
0x00002b0:	20 07	24 15	28 39	32 47	34 68	38 76	42 21	46 29
0x00002d0:	36 72	40 01	44 25	48 33	50 05	54 13	58 37	62 45
0x00002f0:	52 09	56 17	60 41	64 49	34 19	36 35	50 51	52 67
0x0000310:	38 21	40 37	54 53	56 69	42 27	44 43	58 59	60 75
0x0000330:	46 29	48 45	62 61	64 77	66 23	68 39	03 55	05 71
0x0000350:	70 25	72 41	07 57	09 73	74 31	76 47	11 63	13 00
0x0000370:	78 33	01 49	15 65	17 02	66 51	70 67	03 04	07 20
0x0000390:	68 55	72 71	05 08	09 24	74 59	78 75	11 12	15 28
0x00003b0:	76 63	01 00	13 16	17 32	19 53	23 69	35 06	39 22
0x00003d0:	21 57	25 73	37 10	41 26	27 61	31 77	43 14	47 30
0x00003f0:	29 65	33 02	45 18	49 34	19 04	21 08	23 20	25 24

2.1.2 Second set

```
                                                    _  _
Set mode:
Set clock:
Set ULAP:

================================================================================
                                    79 HOP SYSTEM
================================================================================


Hop sequence {k} for PAGE SCAN/INQUIRY SCAN:
CLKN start:     0x0000000
ULAP:           0x2a96ef25
#ticks:         0000 | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 |
                -------------------------------------------------------
0x0000000:        49 |   13 |   17 |   51 |   55 |   19 |   23 |   53 |
0x0008000:        57 |   21 |   25 |   27 |   31 |   74 |   78 |   29 |
0x0010000:        33 |   76 |    1 |   35 |   39 |    3 |    7 |   37 |
0x0018000:        41 |    5 |    9 |   43 |   47 |   11 |   15 |   45 |
0x0020000:        49 |   13 |   17 |   51 |   55 |   19 |   23 |   53 |
0x0028000:        57 |   21 |   25 |   27 |   31 |   74 |   78 |   29 |
0x0030000:        33 |   76 |    1 |   35 |   39 |    3 |    7 |   37 |
0x0038000:        41 |    5 |    9 |   43 |   47 |   11 |   15 |   45 |

Hop sequence {k} for PAGE STATE/INQUIRY STATE:
CLKE start:     0x0000000
ULAP:           0x2a96ef25
#ticks:         00 01 02 03 | 04 05 06 07 | 08 09 0a 0b | 0c 0d 0e 0f |
                -----------------------------------------------------
0x0000000:      41 05 10 04 | 09 43 06 16 | 47 11 18 12 | 15 45 14 32 |
0x0000010:      49 13 34 28 | 17 51 30 24 | 55 19 26 20 | 23 53 22 40 |
0x0000020:      41 05 10 04 | 09 43 06 16 | 47 11 18 12 | 15 45 14 32 |
0x0000030:      49 13 34 28 | 17 51 30 24 | 55 19 26 20 | 23 53 22 40 |
...
0x0001000:      41 21 10 36 | 25 27 38 63 | 31 74 65 59 | 78 29 61 00 |
0x0001010:      33 76 02 75 | 01 35 77 71 | 39 03 73 67 | 07 37 69 08 |
0x0001020:      41 21 10 36 | 25 27 38 63 | 31 74 65 59 | 78 29 61 00 |
0x0001030:      33 76 02 75 | 01 35 77 71 | 39 03 73 67 | 07 37 69 08 |
...
0x0002000:      57 21 42 36 | 09 43 06 16 | 47 11 18 12 | 15 45 14 32 |
0x0002010:      49 13 34 28 | 17 51 30 24 | 55 19 26 20 | 23 53 22 40 |
0x0002020:      57 21 42 36 | 09 43 06 16 | 47 11 18 12 | 15 45 14 32 |
0x0002030:      49 13 34 28 | 17 51 30 24 | 55 19 26 20 | 23 53 22 40 |
...
0x0003000:      41 05 10 04 | 09 27 06 63 | 31 74 65 59 | 78 29 61 00 |
0x0003010:      33 76 02 75 | 01 35 77 71 | 39 03 73 67 | 07 37 69 08 |
0x0003020:      41 05 10 04 | 09 27 06 63 | 31 74 65 59 | 78 29 61 00 |
0x0003030:      33 76 02 75 | 01 35 77 71 | 39 03 73 67 | 07 37 69 08 |

Hop sequence {k} for SLAVE PAGE RESPONSE STATE:
CLKN* =         0x0000010
ULAP:           0x2a96ef25
#ticks:         00 | 02 04 | 06 08 | 0a 0c | 0e 10 | 12 14 | 16 18 | 1a 1c | 1e
                -----------------------------------------------------------------
0x0000012:      34 | 13 28 | 17 30 | 51 24 | 55 26 | 19 20 | 23 22 | 53 40 | 57
0x0000032:      42 | 21 36 | 25 38 | 27 63 | 31 65 | 74 59 | 78 61 | 29 00 | 33
0x0000052:      02 | 76 75 | 01 77 | 35 71 | 39 73 | 03 67 | 07 69 | 37 08 | 41
0x0000072:      10 | 05 04 | 09 06 | 43 16 | 47 18 | 11 12 | 15 14 | 45 32 | 49

Hop sequence {k} for MASTER PAGE RESPONSE STATE:
Offset value:   24
CLKE* =         0x0000012
ULAP:           0x2a96ef25
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                -----------------------------------------------------------------
0x0000014:      13 28 | 17 30 | 51 24 | 55 26 | 19 20 | 23 22 | 53 40 | 57 42 |
0x0000034:      21 36 | 25 38 | 27 63 | 31 65 | 74 59 | 78 61 | 29 00 | 33 02 |
0x0000054:      76 75 | 01 77 | 35 71 | 39 73 | 03 67 | 07 69 | 37 08 | 41 10 |
0x0000074:      05 04 | 09 06 | 43 16 | 47 18 | 11 12 | 15 14 | 45 32 | 49 34 |
```

```
Hop sequence {k} for CONNECTION STATE:
CLK start:      0x0000010
ULAP:           0x2a96ef25
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                -----------------------------------------------------------------
0x0000010:      55 26 | 19 20 | 23 22 | 53 40 | 57 42 | 21 36 | 25 38 | 27 63 |
0x0000030:      31 65 | 74 59 | 78 61 | 29 00 | 33 02 | 76 75 | 01 77 | 35 71 |
0x0000050:      39 73 | 03 67 | 07 69 | 37 08 | 41 10 | 05 04 | 09 06 | 43 16 |
0x0000070:      47 18 | 11 12 | 15 14 | 45 32 | 02 66 | 47 60 | 49 64 | 04 54 |
0x0000090:      06 58 | 51 52 | 53 56 | 08 70 | 10 74 | 55 68 | 57 72 | 59 14 |
0x00000b0:      61 18 | 27 12 | 29 16 | 63 30 | 65 34 | 31 28 | 33 32 | 67 22 |
0x00000d0:      69 26 | 35 20 | 37 24 | 71 38 | 73 42 | 39 36 | 41 40 | 75 46 |
0x00000f0:      77 50 | 43 44 | 45 48 | 00 62 | 26 11 | 69 05 | 73 07 | 36 17 |
0x0000110:      40 19 | 04 13 | 08 15 | 38 25 | 42 27 | 06 21 | 10 23 | 12 48 |
0x0000130:      16 50 | 59 44 | 63 46 | 14 56 | 18 58 | 61 52 | 65 54 | 28 64 |
0x0000150:      32 66 | 75 60 | 00 62 | 30 72 | 34 74 | 77 68 | 02 70 | 20 01 |
0x0000170:      24 03 | 67 76 | 71 78 | 22 09 | 58 43 | 24 37 | 26 41 | 68 47 |
0x0000190:      70 51 | 36 45 | 38 49 | 72 55 | 74 59 | 40 53 | 42 57 | 44 78 |
0x00001b0:      46 03 | 12 76 | 14 01 | 48 07 | 50 11 | 16 05 | 18 09 | 60 15 |
0x00001d0:      62 19 | 28 13 | 30 17 | 64 23 | 66 27 | 32 21 | 34 25 | 52 31 |
0x00001f0:      54 35 | 20 29 | 22 33 | 56 39 | 19 04 | 62 63 | 66 00 | 07 73 |
0x0000210:      11 10 | 54 69 | 58 06 | 23 75 | 27 12 | 70 71 | 74 08 | 76 33 |
0x0000230:      01 49 | 44 29 | 48 45 | 13 35 | 17 51 | 60 31 | 64 47 | 05 41 |
0x0000250:      09 57 | 52 37 | 56 53 | 21 43 | 25 59 | 68 39 | 72 55 | 78 65 |
0x0000270:      03 02 | 46 61 | 50 77 | 15 67 | 51 36 | 17 18 | 19 34 | 41 24 |
0x0000290:      43 40 | 09 22 | 11 38 | 57 28 | 59 44 | 25 26 | 27 42 | 29 63 |
0x00002b0:      31 00 | 76 61 | 78 77 | 45 67 | 47 04 | 13 65 | 15 02 | 37 71 |
0x00002d0:      39 08 | 05 69 | 07 06 | 53 75 | 55 12 | 21 73 | 23 10 | 33 16 |
0x00002f0:      35 32 | 01 14 | 03 30 | 49 20 | 75 60 | 39 48 | 43 56 | 00 66 |
0x0000310:      04 74 | 47 62 | 51 70 | 08 68 | 12 76 | 55 64 | 59 72 | 61 18 |
0x0000330:      65 26 | 29 14 | 33 22 | 69 20 | 73 28 | 37 16 | 41 24 | 77 34 |
0x0000350:      02 42 | 45 30 | 49 38 | 06 36 | 10 44 | 53 32 | 57 40 | 63 50 |
0x0000370:      67 58 | 31 46 | 35 54 | 71 52 | 28 13 | 73 03 | 75 11 | 34 17 |
0x0000390:      36 25 | 02 15 | 04 23 | 42 21 | 44 29 | 10 19 | 12 27 | 14 48 |
0x00003b0:      16 56 | 61 46 | 63 54 | 22 52 | 24 60 | 69 50 | 71 58 | 30 64 |
0x00003d0:      32 72 | 77 62 | 00 70 | 38 68 | 40 76 | 06 66 | 08 74 | 18 01 |
0x00003f0:      20 09 | 65 78 | 67 07 | 26 05 | 44 29 | 32 23 | 36 25 | 70 43 |
```

2.1.3 Third set

```
Set mode:
Set clock:
Set ULAP:

================================================================================
                                    79 HOP SYSTEM
================================================================================


Hop sequence {k} for PAGE SCAN/INQUIRY SCAN:
CLKN start:     0x0000000
ULAP:           0x6587cba9
#ticks:         0000 | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 |
                -----------------------------------------------------
0x0000000:        16 |   65 |   67 |   18 |   20 |   53 |   55 |    6 |
0x0008000:         8 |   57 |   59 |   10 |   12 |   69 |   71 |   22 |
0x0010000:        24 |   73 |   75 |   26 |   28 |   45 |   47 |   77 |
0x0018000:         0 |   49 |   51 |    2 |    4 |   61 |   63 |   14 |
0x0020000:        16 |   65 |   67 |   18 |   20 |   53 |   55 |    6 |
0x0028000:         8 |   57 |   59 |   10 |   12 |   69 |   71 |   22 |
0x0030000:        24 |   73 |   75 |   26 |   28 |   45 |   47 |   77 |
0x0038000:         0 |   49 |   51 |    2 |    4 |   61 |   63 |   14 |

Hop sequence {k} for PAGE STATE/INQUIRY STATE:
CLKE start:     0x0000000
ULAP:           0x6587cba9
#ticks:         00 01 02 03 | 04 05 06 07 | 08 09 0a 0b | 0c 0d 0e 0f |
                -------------------------------------------------------
0x0000000:      00 49 36 38 | 51 02 42 40 | 04 61 44 46 | 63 14 50 48 |
0x0000010:      16 65 52 54 | 67 18 58 56 | 20 53 60 62 | 55 06 66 64 |
0x0000020:      00 49 36 38 | 51 02 42 40 | 04 61 44 46 | 63 14 50 48 |
0x0000030:      16 65 52 54 | 67 18 58 56 | 20 53 60 62 | 55 06 66 64 |
...
0x0001000:      00 57 36 70 | 59 10 74 72 | 12 69 76 78 | 71 22 03 01 |
0x0001010:      24 73 05 07 | 75 26 11 09 | 28 45 13 30 | 47 77 34 32 |
0x0001020:      00 57 36 70 | 59 10 74 72 | 12 69 76 78 | 71 22 03 01 |
0x0001030:      24 73 05 07 | 75 26 11 09 | 28 45 13 30 | 47 77 34 32 |
...
0x0002000:      08 57 68 70 | 51 02 42 40 | 04 61 44 46 | 63 14 50 48 |
0x0002010:      16 65 52 54 | 67 18 58 56 | 20 53 60 62 | 55 06 66 64 |
0x0002020:      08 57 68 70 | 51 02 42 40 | 04 61 44 46 | 63 14 50 48 |
0x0002030:      16 65 52 54 | 67 18 58 56 | 20 53 60 62 | 55 06 66 64 |
...
0x0003000:      00 49 36 38 | 51 10 42 72 | 12 69 76 78 | 71 22 03 01 |
0x0003010:      24 73 05 07 | 75 26 11 09 | 28 45 13 30 | 47 77 34 32 |
0x0003020:      00 49 36 38 | 51 10 42 72 | 12 69 76 78 | 71 22 03 01 |
0x0003030:      24 73 05 07 | 75 26 11 09 | 28 45 13 30 | 47 77 34 32 |

Hop sequence {k} for SLAVE PAGE RESPONSE STATE:
CLKN* =         0x0000010
ULAP:           0x6587cba9
#ticks:         00 | 02 04 | 06 08 | 0a 0c | 0e 10 | 12 14 | 16 18 | 1a 1c | 1e
                -------------------------------------------------------------
0x0000012:      52 | 65 54 | 67 58 | 18 56 | 20 60 | 53 62 | 55 66 | 06 64 | 08
0x0000032:      68 | 57 70 | 59 74 | 10 72 | 12 76 | 69 78 | 71 03 | 22 01 | 24
0x0000052:      05 | 73 07 | 75 11 | 26 09 | 28 13 | 45 30 | 47 34 | 77 32 | 00
0x0000072:      36 | 49 38 | 51 42 | 02 40 | 04 44 | 61 46 | 63 50 | 14 48 | 16

Hop sequence {k} for MASTER PAGE RESPONSE STATE:
Offset value:   24
CLKE* =         0x0000012
ULAP:           0x6587cba9
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                -------------------------------------------------------------
0x0000014:      65 54 | 67 58 | 18 56 | 20 60 | 53 62 | 55 66 | 06 64 | 08 68 |
0x0000034:      57 70 | 59 74 | 10 72 | 12 76 | 69 78 | 71 03 | 22 01 | 24 05 |
0x0000054:      73 07 | 75 11 | 26 09 | 28 13 | 45 30 | 47 34 | 77 32 | 00 36 |
0x0000074:      49 38 | 51 42 | 02 40 | 04 44 | 61 46 | 63 50 | 14 48 | 16 52 |
```

```
Hop sequence {k} for CONNECTION STATE:
CLK start:      0x0000010
ULAP:           0x6587cba9
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                -----------------------------------------------------------------
0x0000010:      20 60 | 53 62 | 55 66 | 06 64 | 08 68 | 57 70 | 59 74 | 10 72 |
0x0000030:      12 76 | 69 78 | 71 03 | 22 01 | 24 05 | 73 07 | 75 11 | 26 09 |
0x0000050:      28 13 | 45 30 | 47 34 | 77 32 | 00 36 | 49 38 | 51 42 | 02 40 |
0x0000070:      04 44 | 61 46 | 63 50 | 14 48 | 50 05 | 16 07 | 20 09 | 48 11 |
0x0000090:      52 13 | 06 15 | 10 17 | 38 19 | 42 21 | 08 23 | 12 25 | 40 27 |
0x00000b0:      44 29 | 22 31 | 26 33 | 54 35 | 58 37 | 24 39 | 28 41 | 56 43 |
0x00000d0:      60 45 | 77 62 | 02 64 | 30 66 | 34 68 | 00 70 | 04 72 | 32 74 |
0x00000f0:      36 76 | 14 78 | 18 01 | 46 03 | 72 29 | 42 39 | 44 43 | 74 41 |
0x0000110:      76 45 | 46 47 | 48 51 | 78 49 | 01 53 | 50 63 | 52 67 | 03 65 |
0x0000130:      05 69 | 54 55 | 56 59 | 07 57 | 09 61 | 58 71 | 60 75 | 11 73 |
0x0000150:      13 77 | 30 15 | 32 19 | 62 17 | 64 21 | 34 31 | 36 35 | 66 33 |
0x0000170:      68 37 | 38 23 | 40 27 | 70 25 | 27 61 | 72 71 | 76 73 | 25 75 |
0x0000190:      29 77 | 78 00 | 03 02 | 31 04 | 35 06 | 01 16 | 05 18 | 33 20 |
0x00001b0:      37 22 | 07 08 | 11 10 | 39 12 | 43 14 | 09 24 | 13 26 | 41 28 |
0x00001d0:      45 30 | 62 47 | 66 49 | 15 51 | 19 53 | 64 63 | 68 65 | 17 67 |
0x00001f0:      21 69 | 70 55 | 74 57 | 23 59 | 53 22 | 35 12 | 37 28 | 67 14 |
0x0000210:      69 30 | 23 32 | 25 48 | 55 34 | 57 50 | 39 40 | 41 56 | 71 42 |
0x0000230:      73 58 | 27 36 | 29 52 | 59 38 | 61 54 | 43 44 | 45 60 | 75 46 |
0x0000250:      77 62 | 15 00 | 17 16 | 47 02 | 49 18 | 31 08 | 33 24 | 63 10 |
0x0000270:      65 26 | 19 04 | 21 20 | 51 06 | 06 54 | 65 42 | 69 58 | 18 46 |
0x0000290:      22 62 | 55 64 | 59 01 | 08 68 | 12 05 | 71 72 | 75 09 | 24 76 |
0x00002b0:      28 13 | 57 66 | 61 03 | 10 70 | 14 07 | 73 74 | 77 11 | 26 78 |
0x00002d0:      30 15 | 47 32 | 51 48 | 00 36 | 04 52 | 63 40 | 67 56 | 16 44 |
0x00002f0:      20 60 | 49 34 | 53 50 | 02 38 | 38 78 | 12 05 | 14 13 | 44 07 |
0x0000310:      46 15 | 16 17 | 18 25 | 48 19 | 50 27 | 24 33 | 26 41 | 56 35 |
0x0000330:      58 43 | 20 21 | 22 29 | 52 23 | 54 31 | 28 37 | 30 45 | 60 39 |
0x0000350:      62 47 | 00 64 | 02 72 | 32 66 | 34 74 | 08 01 | 10 09 | 40 03 |
0x0000370:      42 11 | 04 68 | 06 76 | 36 70 | 70 31 | 42 35 | 46 43 | 74 39 |
0x0000390:      78 47 | 48 49 | 52 57 | 01 53 | 05 61 | 56 65 | 60 73 | 09 69 |
0x00003b0:      13 77 | 50 51 | 54 59 | 03 55 | 07 63 | 58 67 | 62 75 | 11 71 |
0x00003d0:      15 00 | 32 17 | 36 25 | 64 21 | 68 29 | 40 33 | 44 41 | 72 37 |
0x00003f0:      76 45 | 34 19 | 38 27 | 66 23 | 11 71 | 05 18 | 07 22 | 13 20 |
```

2.2 THE 23-HOP SYSTEM SAMPLE DATA

2.2.1 First set

```
                                                  _  _
Set mode:
Set clock:
Set ULAP:

================================================================================
=
                                          23 HOP SYSTEM
================================================================================
=


Hop sequence {k} for PAGE SCAN/INQUIRY SCAN:
CLKN start:     0x0000000
ULAP:           0x00000000
#ticks:         0000 | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 |
                ---------------------------------------------------------
0x0000000:         0 |    2 |    4 |    6 |    8 |   10 |   12 |   14 |
0x0008000:        16 |   18 |   20 |   22 |    1 |    3 |    5 |    7 |
0x0010000:         0 |    2 |    4 |    6 |    8 |   10 |   12 |   14 |
0x0018000:        16 |   18 |   20 |   22 |    1 |    3 |    5 |    7 |
0x0020000:         0 |    2 |    4 |    6 |    8 |   10 |   12 |   14 |
0x0028000:        16 |   18 |   20 |   22 |    1 |    3 |    5 |    7 |
0x0030000:         0 |    2 |    4 |    6 |    8 |   10 |   12 |   14 |
0x0038000:        16 |   18 |   20 |   22 |    1 |    3 |    5 |    7 |

Hop sequence {k} for PAGE STATE/INQUIRY STATE:
CLKE start:     0x0000000
ULAP:           0x00000000
#ticks:         00 01 02 03 | 04 05 06 07 | 08 09 0a 0b | 0c 0d 0e 0f |
                ---------------------------------------------------------
0x0000000:      16 18 02 10 | 20 22 06 14 | 01 03 04 12 | 05 07 08 16 |
0x0000010:      00 02 09 17 | 04 06 13 21 | 08 10 11 19 | 12 14 15 00 |
0x0000020:      16 18 02 10 | 20 22 06 14 | 01 03 04 12 | 05 07 08 16 |
0x0000030:      00 02 09 17 | 04 06 13 21 | 08 10 11 19 | 12 14 15 00 |
...
0x0001000:      18 20 10 06 | 22 01 14 04 | 03 05 12 08 | 07 00 16 09 |
0x0001010:      02 04 17 13 | 06 08 21 11 | 10 12 19 15 | 14 16 00 02 |
0x0001020:      18 20 10 06 | 22 01 14 04 | 03 05 12 08 | 07 00 16 09 |
0x0001030:      02 04 17 13 | 06 08 21 11 | 10 12 19 15 | 14 16 00 02 |
...
0x0002000:      20 22 06 14 | 01 03 04 12 | 05 07 08 16 | 00 02 09 17 |
0x0002010:      04 06 13 21 | 08 10 11 19 | 12 14 15 00 | 16 18 02 10 |
0x0002020:      20 22 06 14 | 01 03 04 12 | 05 07 08 16 | 00 02 09 17 |
0x0002030:      04 06 13 21 | 08 10 11 19 | 12 14 15 00 | 16 18 02 10 |
...
0x0003000:      22 01 14 04 | 03 05 12 08 | 07 00 16 09 | 02 04 17 13 |
0x0003010:      06 08 21 11 | 10 12 19 15 | 14 16 00 02 | 18 20 10 06 |
0x0003020:      22 01 14 04 | 03 05 12 08 | 07 00 16 09 | 02 04 17 13 |
0x0003030:      06 08 21 11 | 10 12 19 15 | 14 16 00 02 | 18 20 10 06 |

Hop sequence {k} for SLAVE PAGE RESPONSE STATE:
CLKN* =         0x0000010
ULAP:           0x00000000
#ticks:         00 | 02 04 | 06 08 | 0a 0c | 0e 10 | 12 14 | 16 18 | 1a 1c | 1e
                ---------------------------------------------------------------
0x0000012:      09 | 02 17 | 04 13 | 06 21 | 08 11 | 10 19 | 12 15 | 14 00 | 16
0x0000032:      02 | 18 10 | 20 06 | 22 14 | 01 04 | 03 12 | 05 08 | 07 16 | 00
0x0000052:      09 | 02 17 | 04 13 | 06 21 | 08 11 | 10 19 | 12 15 | 14 00 | 16
0x0000072:      02 | 18 10 | 20 06 | 22 14 | 01 04 | 03 12 | 05 08 | 07 16 | 00

Hop sequence {k} for MASTER PAGE RESPONSE STATE:
Offset value:   24
CLKE* =         0x0000012
ULAP:           0x00000000
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                ---------------------------------------------------------------
0x0000014:      02 17 | 04 13 | 06 21 | 08 11 | 10 19 | 12 15 | 14 00 | 16 02 |
0x0000034:      18 10 | 20 06 | 22 14 | 01 04 | 03 12 | 05 08 | 07 16 | 00 09 |
```

```
0x0000054:      02 17 | 04 13 | 06 21 | 08 11 | 10 19 | 12 15 | 14 00 | 16 02 |
0x0000074:      18 10 | 20 06 | 22 14 | 01 04 | 03 12 | 05 08 | 07 16 | 00 09 |

Hop sequence {k} for CONNECTION STATE:
CLK start:      0x0000010
ULAP:           0x00000000
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                ------------------------------------------------------------------
0x0000010:      08 11 | 10 19 | 12 15 | 14 00 | 16 02 | 18 10 | 20 06 | 22 14 |
0x0000030:      01 04 | 03 12 | 05 08 | 07 16 | 16 02 | 18 10 | 20 06 | 22 14 |
0x0000050:      01 04 | 03 12 | 05 08 | 07 16 | 09 18 | 11 03 | 13 22 | 15 07 |
0x0000070:      17 20 | 19 05 | 21 01 | 00 09 | 09 18 | 13 03 | 11 20 | 15 05 |
0x0000090:      17 22 | 21 07 | 19 01 | 00 09 | 02 11 | 06 19 | 04 13 | 08 21 |
0x00000b0:      10 15 | 14 00 | 12 17 | 16 02 | 02 11 | 06 19 | 04 13 | 08 21 |
0x00000d0:      10 15 | 14 00 | 12 17 | 16 02 | 18 04 | 22 12 | 20 06 | 01 14 |
0x00000f0:      03 08 | 07 16 | 05 10 | 09 18 | 18 04 | 20 20 | 22 08 | 01 01 |
0x0000110:      11 06 | 13 22 | 15 10 | 17 03 | 03 12 | 05 05 | 07 16 | 09 09 |
0x0000130:      19 14 | 21 07 | 00 18 | 02 11 | 11 20 | 13 13 | 15 01 | 17 17 |
0x0000150:      04 22 | 06 15 | 08 03 | 10 19 | 19 05 | 21 21 | 00 09 | 02 02 |
0x0000170:      12 07 | 14 00 | 16 11 | 18 04 | 04 13 | 08 06 | 06 15 | 10 08 |
0x0000190:      20 17 | 01 10 | 22 19 | 03 12 | 12 21 | 16 14 | 14 00 | 18 16 |
0x00001b0:      05 02 | 09 18 | 07 04 | 11 20 | 20 06 | 01 22 | 22 08 | 03 01 |
0x00001d0:      13 10 | 17 03 | 15 12 | 19 05 | 05 14 | 09 07 | 07 16 | 11 09 |
0x00001f0:      21 18 | 02 11 | 00 20 | 04 13 | 13 22 | 06 07 | 17 03 | 10 11 |
0x0000210:      21 15 | 14 00 | 02 19 | 18 04 | 15 01 | 08 09 | 19 05 | 12 13 |
0x0000230:      00 17 | 16 02 | 04 21 | 20 06 | 06 15 | 22 00 | 10 19 | 03 04 |
0x0000250:      14 08 | 07 16 | 18 12 | 11 20 | 08 17 | 01 02 | 12 21 | 05 06 |
0x0000270:      16 10 | 09 18 | 20 14 | 13 22 | 22 08 | 15 16 | 01 10 | 17 18 |
0x0000290:      07 01 | 00 09 | 09 03 | 02 11 | 03 12 | 19 20 | 05 14 | 21 22 |
0x00002b0:      11 05 | 04 13 | 13 07 | 06 15 | 15 01 | 08 09 | 17 03 | 10 11 |
0x00002d0:      00 17 | 16 02 | 02 19 | 18 04 | 19 05 | 12 13 | 21 07 | 14 15 |
0x00002f0:      04 21 | 20 06 | 06 00 | 22 08 | 08 17 | 16 10 | 12 21 | 20 14 |
0x0000310:      01 02 | 09 18 | 05 06 | 13 22 | 10 19 | 18 12 | 14 00 | 22 16 |
0x0000330:      03 04 | 11 20 | 07 08 | 15 01 | 01 10 | 09 03 | 05 14 | 13 07 |
0x0000350:      17 18 | 02 11 | 21 22 | 06 15 | 03 12 | 11 05 | 07 16 | 15 09 |
0x0000370:      19 20 | 04 13 | 00 01 | 08 17 | 17 03 | 02 19 | 19 05 | 04 21 |
0x0000390:      10 11 | 18 04 | 12 13 | 20 06 | 21 07 | 06 00 | 00 09 | 08 02 |
0x00003b0:      14 15 | 22 08 | 16 17 | 01 10 | 10 19 | 18 12 | 12 21 | 20 14 |
0x00003d0:      03 04 | 11 20 | 05 06 | 13 22 | 14 00 | 22 16 | 16 02 | 01 18 |
0x00003f0:      07 08 | 15 01 | 09 10 | 17 03 | 03 12 | 05 16 | 11 20 | 13 01 |
```

2.2.2 Second set

```
Set mode:
Set clock:
Set ULAP:

================================================================================

                                    23 HOP SYSTEM
================================================================================


Hop sequence {k} for PAGE SCAN/INQUIRY SCAN:
CLKN start:     0x0000000
ULAP:           0x2a96ef25
#ticks:         0000 | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 |
                -------------------------------------------------------
0x0000000:         7 |   18 |    3 |    8 |   16 |    4 |   12 |    1 |
0x0008000:         9 |   20 |    5 |    6 |   14 |    2 |   10 |   22 |
0x0010000:         7 |   18 |    3 |    8 |   16 |    4 |   12 |    1 |
0x0018000:         9 |   20 |    5 |    6 |   14 |    2 |   10 |   22 |
0x0020000:         7 |   18 |    3 |    8 |   16 |    4 |   12 |    1 |
0x0028000:         9 |   20 |    5 |    6 |   14 |    2 |   10 |   22 |
0x0030000:         7 |   18 |    3 |    8 |   16 |    4 |   12 |    1 |
0x0038000:         9 |   20 |    5 |    6 |   14 |    2 |   10 |   22 |

Hop sequence {k} for PAGE STATE/INQUIRY STATE:
CLKE start:     0x0000000
ULAP:           0x2a96ef25
#ticks:         00 01 02 03 | 04 05 06 07 | 08 09 0a 0b | 0c 0d 0e 0f |
                -----------------------------------------------------
0x0000000:      09 20 18 00 | 05 06 02 04 | 14 02 06 11 | 10 22 13 12 |
0x0000010:      07 18 14 19 | 03 08 21 08 | 16 04 10 15 | 12 01 17 16 |
0x0000020:      09 20 18 00 | 05 06 02 04 | 14 02 06 11 | 10 22 13 12 |
0x0000030:      07 18 14 19 | 03 08 21 08 | 16 04 10 15 | 12 01 17 16 |
...
0x0001000:      20 05 00 02 | 06 14 04 06 | 02 10 11 13 | 22 07 12 14 |
0x0001010:      18 03 19 21 | 08 16 08 10 | 04 12 15 17 | 01 09 16 18 |
0x0001020:      20 05 00 02 | 06 14 04 06 | 02 10 11 13 | 22 07 12 14 |
0x0001030:      18 03 19 21 | 08 16 08 10 | 04 12 15 17 | 01 09 16 18 |
...
0x0002000:      05 06 02 04 | 14 02 06 11 | 10 22 13 12 | 07 18 14 19 |
0x0002010:      03 08 21 08 | 16 04 10 15 | 12 01 17 16 | 09 20 18 00 |
0x0002020:      05 06 02 04 | 14 02 06 11 | 10 22 13 12 | 07 18 14 19 |
0x0002030:      03 08 21 08 | 16 04 10 15 | 12 01 17 16 | 09 20 18 00 |
...
0x0003000:      06 14 04 06 | 02 10 11 13 | 22 07 12 14 | 18 03 19 21 |
0x0003010:      08 16 08 10 | 04 12 15 17 | 01 09 16 18 | 20 05 00 02 |
0x0003020:      06 14 04 06 | 02 10 11 13 | 22 07 12 14 | 18 03 19 21 |
0x0003030:      08 16 08 10 | 04 12 15 17 | 01 09 16 18 | 20 05 00 02 |

Hop sequence {k} for SLAVE PAGE RESPONSE STATE:
CLKN* =         0x0000010
ULAP:           0x2a96ef25
#ticks:         00 | 02 04 | 06 08 | 0a 0c | 0e 10 | 12 14 | 16 18 | 1a 1c | 1e
                ---------------------------------------------------------------
0x0000012:      14 | 18 19 | 03 21 | 08 08 | 16 10 | 04 15 | 12 17 | 01 16 | 09
0x0000032:      18 | 20 00 | 05 02 | 06 04 | 14 06 | 02 11 | 10 13 | 22 12 | 07
0x0000052:      14 | 18 19 | 03 21 | 08 08 | 16 10 | 04 15 | 12 17 | 01 16 | 09
0x0000072:      18 | 20 00 | 05 02 | 06 04 | 14 06 | 02 11 | 10 13 | 22 12 | 07

Hop sequence {k} for MASTER PAGE RESPONSE STATE:
Offset value:   24
CLKE* =         0x0000012
ULAP:           0x2a96ef25
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                -----------------------------------------------------------
0x0000014:      18 19 | 03 21 | 08 08 | 16 10 | 04 15 | 12 17 | 01 16 | 09 18 |
0x0000034:      20 00 | 05 02 | 06 04 | 14 06 | 02 11 | 10 13 | 22 12 | 07 14 |
```

```
                                  fh_23_B.txt
0x0000054:      18 19 | 03 21 | 08 08 | 16 10 | 04 15 | 12 17 | 01 16 | 09 18 |
0x0000074:      20 00 | 05 02 | 06 04 | 14 06 | 02 11 | 10 13 | 22 12 | 07 14 |

Hop sequence {k} for CONNECTION STATE:
CLK start:      0x0000010
ULAP:           0x2a96ef25
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                ---------------------------------------------------------------
0x0000010:      16 10 | 04 15 | 12 17 | 01 16 | 09 18 | 20 00 | 05 02 | 06 04 |
0x0000030:      14 06 | 02 11 | 10 13 | 22 12 | 00 07 | 11 12 | 19 14 | 01 01 |
0x0000050:      09 03 | 20 08 | 05 10 | 17 09 | 02 11 | 13 16 | 21 18 | 22 20 |
0x0000070:      07 22 | 18 04 | 03 06 | 15 05 | 14 02 | 04 05 | 12 09 | 17 15 |
0x0000090:      02 19 | 15 22 | 00 03 | 10 00 | 18 04 | 08 07 | 16 11 | 13 13 |
0x00000b0:      21 17 | 11 20 | 19 01 | 06 21 | 07 18 | 20 21 | 05 02 | 10 08 |
0x00000d0:      18 12 | 08 15 | 16 19 | 03 16 | 11 20 | 01 00 | 09 04 | 06 06 |
0x00000f0:      14 10 | 04 13 | 12 17 | 22 14 | 02 09 | 05 22 | 21 01 | 03 18 |
0x0000110:      19 20 | 22 10 | 15 12 | 11 11 | 04 13 | 07 03 | 00 05 | 01 14 |
0x0000130:      17 16 | 20 06 | 13 08 | 09 07 | 18 02 | 21 15 | 14 17 | 19 11 |
0x0000150:      12 13 | 15 03 | 08 05 | 04 04 | 20 06 | 00 19 | 16 21 | 17 07 |
0x0000170:      10 09 | 13 22 | 06 01 | 02 00 | 09 20 | 14 08 | 07 12 | 12 02 |
0x0000190:      05 06 | 10 17 | 03 21 | 20 18 | 13 22 | 18 10 | 11 14 | 08 00 |
0x00001b0:      01 04 | 06 15 | 22 19 | 16 16 | 02 13 | 07 01 | 00 05 | 05 18 |
0x00001d0:      21 22 | 03 10 | 19 14 | 13 11 | 06 15 | 11 03 | 04 07 | 01 16 |
0x00001f0:      17 20 | 22 08 | 15 12 | 09 09 | 20 00 | 08 05 | 12 07 | 02 02 |
0x0000210:      06 04 | 17 09 | 21 11 | 18 06 | 22 08 | 10 13 | 14 15 | 00 17 |
0x0000230:      04 19 | 15 01 | 19 03 | 16 21 | 13 16 | 01 21 | 05 00 | 18 18 |
0x0000250:      22 20 | 10 02 | 14 04 | 11 22 | 15 01 | 03 06 | 07 08 | 16 10 |
0x0000270:      20 12 | 08 17 | 12 19 | 09 14 | 04 09 | 17 12 | 19 16 | 13 11 |
0x0000290:      15 15 | 05 18 | 07 22 | 06 13 | 08 17 | 21 20 | 00 01 | 09 03 |
0x00002b0:      11 07 | 01 10 | 03 14 | 02 05 | 20 02 | 10 05 | 12 09 | 06 04 |
0x00002d0:      08 08 | 21 11 | 00 15 | 22 06 | 01 10 | 14 13 | 16 17 | 02 19 |
0x00002f0:      04 00 | 17 03 | 19 07 | 18 21 | 15 10 | 18 00 | 22 02 | 05 20 |
0x0000310:      09 22 | 12 12 | 16 14 | 13 01 | 17 03 | 20 16 | 01 18 | 03 04 |
0x0000330:      07 06 | 10 19 | 14 21 | 11 08 | 08 03 | 11 16 | 15 18 | 21 13 |
0x0000350:      02 15 | 05 05 | 09 07 | 06 17 | 10 19 | 13 09 | 17 11 | 19 20 |
0x0000370:      00 22 | 03 12 | 07 14 | 04 01 | 22 19 | 04 07 | 06 11 | 16 06 |
0x0000390:      18 10 | 00 21 | 02 02 | 01 08 | 03 12 | 08 00 | 10 04 | 12 13 |
0x00003b0:      14 17 | 19 05 | 21 09 | 20 15 | 15 12 | 20 00 | 22 04 | 09 22 |
0x00003d0:      11 03 | 16 14 | 18 18 | 17 01 | 19 05 | 01 16 | 03 20 | 05 06 |
0x00003f0:      07 10 | 12 21 | 14 02 | 13 08 | 19 17 | 07 22 | 15 15 | 02 20 |
```

2.2.3 Third set

```
                                                    _  _
Set mode:
Set clock:
Set ULAP:

================================================================================

                                        23 HOP SYSTEM
================================================================================


Hop sequence {k} for PAGE SCAN/INQUIRY SCAN:
CLKN start:     0x0000000
ULAP:           0x6587cba9
#ticks:         0000 | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 | 7000 |
                -------------------------------------------------------
0x0000000:         0 |   10 |   14 |   12 |   16 |    9 |   13 |   11 |
0x0008000:        15 |    2 |    6 |    4 |    8 |   17 |   21 |   19 |
0x0010000:         0 |   10 |   14 |   12 |   16 |    9 |   13 |   11 |
0x0018000:        15 |    2 |    6 |    4 |    8 |   17 |   21 |   19 |
0x0020000:         0 |   10 |   14 |   12 |   16 |    9 |   13 |   11 |
0x0028000:        15 |    2 |    6 |    4 |    8 |   17 |   21 |   19 |
0x0030000:         0 |   10 |   14 |   12 |   16 |    9 |   13 |   11 |
0x0038000:        15 |    2 |    6 |    4 |    8 |   17 |   21 |   19 |

Hop sequence {k} for PAGE STATE/INQUIRY STATE:
CLKE start:     0x0000000
ULAP:           0x6587cba9
#ticks:         00 01 02 03 | 04 05 06 07 | 08 09 0a 0b | 0c 0d 0e 0f |
                -----------------------------------------------------
0x0000000:      15 02 05 11 | 06 04 19 13 | 08 17 21 22 | 21 19 07 01 |
0x0000010:      00 10 09 15 | 14 12 00 17 | 16 09 02 18 | 13 11 03 20 |
0x0000020:      15 02 05 11 | 06 04 19 13 | 08 17 21 22 | 21 19 07 01 |
0x0000030:      00 10 09 15 | 14 12 00 17 | 16 09 02 18 | 13 11 03 20 |
...
0x0001000:      02 06 11 19 | 04 08 13 21 | 17 21 22 07 | 19 00 01 09 |
0x0001010:      10 14 15 00 | 12 16 17 02 | 09 13 18 03 | 11 15 20 05 |
0x0001020:      02 06 11 19 | 04 08 13 21 | 17 21 22 07 | 19 00 01 09 |
0x0001030:      10 14 15 00 | 12 16 17 02 | 09 13 18 03 | 11 15 20 05 |
...
0x0002000:      06 04 19 13 | 08 17 21 22 | 21 19 07 01 | 00 10 09 15 |
0x0002010:      14 12 00 17 | 16 09 02 18 | 13 11 03 20 | 15 02 05 11 |
0x0002020:      06 04 19 13 | 08 17 21 22 | 21 19 07 01 | 00 10 09 15 |
0x0002030:      14 12 00 17 | 16 09 02 18 | 13 11 03 20 | 15 02 05 11 |
...
0x0003000:      04 08 13 21 | 17 21 22 07 | 19 00 01 09 | 10 14 15 00 |
0x0003010:      12 16 17 02 | 09 13 18 03 | 11 15 20 05 | 02 06 11 19 |
0x0003020:      04 08 13 21 | 17 21 22 07 | 19 00 01 09 | 10 14 15 00 |
0x0003030:      12 16 17 02 | 09 13 18 03 | 11 15 20 05 | 02 06 11 19 |

Hop sequence {k} for SLAVE PAGE RESPONSE STATE:
CLKN* =         0x0000010
ULAP:           0x6587cba9
#ticks:         00 | 02 04 | 06 08 | 0a 0c | 0e 10 | 12 14 | 16 18 | 1a 1c | 1e
                ---------------------------------------------------------------
0x0000012:      09 | 10 15 | 14 00 | 12 17 | 16 02 | 09 18 | 13 03 | 11 20 | 15
0x0000032:      05 | 02 11 | 06 19 | 04 13 | 08 21 | 17 22 | 21 07 | 19 01 | 00
0x0000052:      09 | 10 15 | 14 00 | 12 17 | 16 02 | 09 18 | 13 03 | 11 20 | 15
0x0000072:      05 | 02 11 | 06 19 | 04 13 | 08 21 | 17 22 | 21 07 | 19 01 | 00

Hop sequence {k} for MASTER PAGE RESPONSE STATE:
Offset value:   24
CLKE* =         0x0000012
ULAP:           0x6587cba9
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                -----------------------------------------------------------
0x0000014:      10 15 | 14 00 | 12 17 | 16 02 | 09 18 | 13 03 | 11 20 | 15 05 |
0x0000034:      02 11 | 06 19 | 04 13 | 08 21 | 17 22 | 21 07 | 19 01 | 00 09 |
```

```
                                        fh_23_0.txt
0x0000054:      10 15 | 14 00 | 12 17 | 16 02 | 09 18 | 13 03 | 11 20 | 15 05 |
0x0000074:      02 11 | 06 19 | 04 13 | 08 21 | 17 22 | 21 07 | 19 01 | 00 09 |

Hop sequence {k} for CONNECTION STATE:
CLK start:      0x0000010
ULAP:           0x6587cba9
#ticks:         00 02 | 04 06 | 08 0a | 0c 0e | 10 12 | 14 16 | 18 1a | 1c 1e |
                ---------------------------------------------------------------
0x0000010:      16 02 | 09 18 | 13 03 | 11 20 | 15 05 | 02 11 | 06 19 | 04 13 |
0x0000030:      08 21 | 17 22 | 21 07 | 19 01 | 16 02 | 03 08 | 07 16 | 05 10 |
0x0000050:      09 18 | 02 11 | 06 19 | 04 13 | 08 21 | 18 04 | 22 12 | 20 06 |
0x0000070:      01 14 | 10 15 | 14 00 | 12 17 | 09 18 | 19 22 | 21 07 | 00 03 |
0x0000090:      02 11 | 18 04 | 20 12 | 22 08 | 01 16 | 11 20 | 13 05 | 15 01 |
0x00000b0:      17 09 | 03 06 | 05 14 | 07 10 | 02 11 | 12 15 | 14 00 | 16 19 |
0x00000d0:      18 04 | 11 20 | 13 05 | 15 01 | 17 09 | 04 13 | 06 21 | 08 17 |
0x00000f0:      10 02 | 19 22 | 21 07 | 00 03 | 03 12 | 05 02 | 09 18 | 07 04 |
0x0000110:      11 20 | 04 13 | 08 06 | 06 15 | 10 08 | 12 21 | 16 14 | 14 00 |
0x0000130:      18 16 | 20 17 | 01 10 | 22 19 | 19 05 | 21 18 | 02 11 | 00 20 |
0x0000150:      04 13 | 20 06 | 01 22 | 22 08 | 03 01 | 05 14 | 09 07 | 07 16 |
0x0000170:      11 09 | 13 10 | 17 03 | 15 12 | 12 21 | 14 09 | 16 02 | 18 13 |
0x0000190:      20 06 | 13 22 | 15 15 | 17 03 | 19 19 | 21 07 | 00 00 | 02 11 |
0x00001b0:      04 04 | 06 01 | 08 17 | 10 05 | 05 14 | 07 02 | 09 18 | 11 06 |
0x00001d0:      13 22 | 06 15 | 08 08 | 10 19 | 12 12 | 14 00 | 16 16 | 18 04 |
0x00001f0:      20 20 | 22 17 | 01 10 | 03 21 | 13 22 | 17 05 | 21 07 | 02 13 |
0x0000210:      06 15 | 22 08 | 03 10 | 07 16 | 11 18 | 15 01 | 19 03 | 00 09 |
0x0000230:      04 11 | 01 12 | 05 14 | 09 20 | 06 15 | 10 21 | 14 00 | 18 06 |
0x0000250:      22 08 | 15 01 | 19 03 | 00 09 | 04 11 | 08 17 | 12 19 | 16 02 |
0x0000270:      20 04 | 17 05 | 21 07 | 02 13 | 22 08 | 05 12 | 07 16 | 13 20 |
0x0000290:      15 01 | 08 17 | 10 21 | 16 02 | 18 06 | 01 10 | 03 14 | 09 18 |
0x00002b0:      11 22 | 12 19 | 14 00 | 20 04 | 15 01 | 21 05 | 00 09 | 06 13 |
0x00002d0:      08 17 | 01 10 | 03 14 | 09 18 | 11 22 | 17 03 | 19 07 | 02 11 |
0x00002f0:      04 15 | 05 12 | 07 16 | 13 20 | 16 02 | 04 15 | 08 17 | 20 08 |
0x0000310:      01 10 | 17 03 | 21 05 | 10 19 | 14 21 | 02 11 | 06 13 | 18 04 |
0x0000330:      22 06 | 19 07 | 00 09 | 12 00 | 09 18 | 20 08 | 01 10 | 13 01 |
0x0000350:      17 03 | 10 19 | 14 21 | 03 12 | 07 14 | 18 04 | 22 06 | 11 20 |
0x0000370:      15 22 | 12 00 | 16 02 | 05 16 | 02 11 | 15 22 | 17 03 | 08 15 |
0x0000390:      10 19 | 03 12 | 05 16 | 19 05 | 21 09 | 11 20 | 13 01 | 04 13 |
0x00003b0:      06 17 | 07 14 | 09 18 | 00 07 | 18 04 | 08 15 | 10 19 | 01 08 |
0x00003d0:      03 12 | 19 05 | 21 09 | 12 21 | 14 02 | 04 13 | 06 17 | 20 06 |
0x00003f0:      22 10 | 00 07 | 02 11 | 16 00 | 15 01 | 01 18 | 17 03 | 03 20 |
```

3 ACCESS CODE SAMPLE DATA

Different access codes (GIAC, DIACs, others...)

LAP with LSB as rightmost bit.

	Bit transmit order on air		
LAP:	Preamble:	Sync word:	Trailer:
000000	5	7e7041e3 4000000d	5
ffffff	a	e758b522 7ffffff2	a
9e8b33	5	475c58cc 73345e72	a
9e8b34	5	28ed3c34 cb345e72	a
9e8b36	5	62337b64 1b345e72	a
9e8b39	a	c05747b9 e7345e72	a
9e8b3d	5	7084eab0 2f345e72	a
9e8b42	5	64c86d2b 90b45e72	a
9e8b48	a	e3c3725e 04b45e72	a
9e8b4f	a	8c7216a6 bcb45e72	a
9e8b57	a	b2f16c30 fab45e72	a
9e8b60	5	57bd3b22 c1b45e72	a
9e8b6a	a	d0b62457 55b45e72	a
9e8b75	a	81843a39 abb45e72	a
9e8b81	5	0ca96681 e0745e72	a
9e8b8e	a	aecd5a5c 1c745e72	a
9e8b9c	5	17453fbf ce745e72	a
9e8bab	a	f20968ad f5745e72	a
9e8bbb	5	015f4a1e f7745e72	a
9e8bcc	a	d8c695a0 0cf45e72	a
9e8bde	5	614ef043 def45e72	a
9e8bf1	a	ba81ddc7 a3f45e72	a
9e8c05	5	64a7dc4f 680c5e72	a
9e8c1a	5	3595c221 960c5e72	a
9e8c30	a	cb35cc0d 830c5e72	a
9e8c47	5	12ac13b3 788c5e72	a
9e8c5f	5	2c2f6925 3e8c5e72	a
9e8c78	5	3a351c84 078c5e72	a
9e8c92	5	7396d0f3 124c5e72	a
9e8cad	5	5b0fdfc4 6d4c5e72	a
9e8cc9	a	aea2eb38 e4cc5e72	a
9e8ce6	5	756dc6bc 99cc5e72	a
9e8d04	5	214cf934 882c5e72	a
9e8d23	5	37568c95 b12c5e72	a
9e8d43	5	72281560 f0ac5e72	a
9e8d64	5	643260c1 c9ac5e72	a
9e8d86	a	e044f493 986c5e72	a
9e8da9	5	3b8bd917 e56c5e72	a
9e8dcd	a	ce26edeb 6cec5e72	a
9e8df2	a	e6bfe2dc 13ec5e72	a
9e8e18	a	82dcde3d c61c5e72	a
9e8e3f	a	94c6ab9c ff1c5e72	a

9e8e67	a	969059a6 799c5e72	a
9e8e90	a	c4dfccef 425c5e72	a
9e8eba	5	3a7fc2c3 575c5e72	a
9e8ee5	5	57985401 69dc5e72	a
9e8f11	5	0ae2a363 623c5e72	a
9e8f3e	a	d12d8ee7 1f3c5e72	a
9e8f6c	5	547063a8 0dbc5e72	a
9e8f9b	5	063ff6e1 367c5e72	a
9e8fcb	a	c9bc5cfe f4fc5e72	a
9e8ffc	5	2cf00bec cffc5e72	a
9e902e	a	8ec5052f 5d025e72	a
9e9061	5	1074b15e 61825e72	a
9e9095	a	9d59ede6 2a425e72	a
9e90ca	a	f0be7b24 14c25e72	a
9e9100	5	10e10dd0 c0225e72	a
9e9137	a	f5ad5ac2 fb225e72	a
9e916f	a	f7fba8f8 7da25e72	a
9e91a8	5	2f490e5b c5625e72	a
9e91e2	a	94979982 91e25e72	a
9e921d	5	26cda478 2e125e72	a
9e9259	a	aacb81dd 26925e72	a
9e9296	a	bfac7f5b da525e72	a
9e92d4	a	c9a7b0a7 cad25e72	a
9e9313	a	c142bdde 32325e72	a
616cec	5	586a491f 0dcda18d	5
616ceb	5	37db2de7 b5cda18d	5
616ce9	5	7d056ab7 65cda18d	5
616ce6	a	df61566a 99cda18d	5
616ce2	5	6fb2fb63 51cda18d	5
616cdd	5	472bf454 2ecda18d	5
616cd7	a	c020eb21 bacda18d	5
616cd0	a	af918fd9 02cda18d	5
616cc8	a	9112f54f 44cda18d	5
616cbf	5	488b2af1 bf4da18d	5
616cb5	a	cf803584 2b4da18d	5
616caa	a	9eb22bea d54da18d	5
616c9e	a	a49cb509 9e4da18d	5
616c91	5	06f889d4 624da18d	5
616c83	a	bf70ec37 b04da18d	5
616c74	a	ed3f797e 8b8da18d	5
616c64	5	1e695bcd 898da18d	5
616c53	a	fb250cdf b28da18d	5
616c41	5	42ad693c 608da18d	5
616c2e	a	a5b7cc14 dd0da18d	5
616c1a	a	9f9952f7 960da18d	5
616c05	a	ceab4c99 680da18d	5
616bef	a	d403ddde fdf5a18d	5
616bd8	5	314f8acc c6f5a18d	5
616bc0	5	0fccf05a 80f5a18d	5
616ba7	5	25030d57 7975a18d	5
616b8d	a	dba3037b 6c75a18d	5
616b72	5	4439ce17 13b5a18d	5

616b56	a	8d417247 5ab5a18d	5
616b39	5	6a5bd76f e735a18d	5
616b1b	5	592e8166 b635a18d	5
616afc	5	28609d46 cfd5a18d	5
616adc	5	51cb8c1f 4ed5a18d	5
616abb	5	7b047112 b755a18d	5
616a99	5	4871271b e655a18d	5
616a76	5	24bdc8c4 9b95a18d	5
616a52	a	edc57494 d295a18d	5
616a2d	a	f989f30f 6d15a18d	5
616a07	5	0729fd23 7815a18d	5
6169e0	a	8bf0ba4f 81e5a18d	5
6169b8	a	89a64875 0765a18d	5
61698f	5	6cea1f67 3c65a18d	5
616965	5	2549d310 29a5a18d	5
61693a	5	48ae45d2 1725a18d	5
61690e	5	7280db31 5c25a18d	5
6168e1	a	ce1b9f34 61c5a18d	5
6168b3	5	4b46727b 7345a18d	5
616884	a	ae0a2569 4845a18d	5
616854	a	ea5fc581 4a85a18d	5
616823	5	33c61a3f b105a18d	5
6167f1	a	c49fb8c5 63f9a18d	5
6167be	5	5a2e0cb4 5f79a18d	5
61678a	5	60009257 1479a18d	5
616755	a	86314e62 eab9a18d	5
61671f	5	3defd9bb be39a18d	5
6166e8	a	bff7e728 c5d9a18d	5
6166b0	a	bda11512 4359a18d	5
616677	5	6513b3b1 fb99a18d	5
61663d	a	decd2468 af19a18d	5
616602	a	f6542b5f d019a18d	5
6165c6	a	dc44b49b d8e9a18d	5
616589	5	42f500ea e469a18d	5
61654b	a	bf2885e1 34a9a18d	5
61650c	a	ec4c69b5 4c29a18d	5

4 HEC AND PACKET HEADER SAMPLE DATA

Test vectors for HEC and header. Note that UAP, Data, and HEC are in hexadecimal notation, while the header is in octal notation. The header is transmitted from left to right over air.

UAP	Data	HEC	Header (octal not.)
00	123	e1	770007 007070 000777
47	123	06	770007 007007 700000
00	124	32	007007 007007 007700
47	124	d5	007007 007070 707077
00	125	5a	707007 007007 077070
47	125	bd	707007 007070 777707
00	126	e2	077007 007007 000777
47	126	05	077007 007070 700000
00	127	8a	777007 007007 070007
47	127	6d	777007 007070 770770
00	11b	9e	770770 007007 777007
47	11b	79	770770 007070 077770
00	11c	4d	007770 007070 770070
47	11c	aa	007770 007007 070707
00	11d	25	707770 007070 700700
47	11d	c2	707770 007007 000077
00	11e	9d	077770 007070 777007
47	11e	7a	077770 007007 077770
00	11f	f5	777770 007070 707777
47	11f	12	777770 007007 007000

5 CRC SAMPLE DATA

```
Sample CRC generation


Data:
-----

data[0] = 0x4e
data[1] = 0x01
data[2] = 0x02
data[3] = 0x03
data[4] = 0x04
data[5] = 0x05
data[6] = 0x06
data[7] = 0x07
data[8] = 0x08
data[9] = 0x09

UAP = 0x47


==> CRC = 6d d2

Codeword (hexadecimal notation):
--------------------------------

4e 01 02 03 04 05 06 07 08 09 6d d2


NB: Over air each byte in the codeword
    is sent with the LSB first.
```

6 COMPLETE SAMPLE PACKETS

6.1 EXAMPLE OF DH1 PACKET

```
Packet header: (MSB...LSB)
--------------------------
AM_ADDR = 011
TYPE = 0100 (DH1)
FLOW = 0
ARQN = 1
SEQN = 0

Payload: (MSB...LSB)
--------------------
payload length: 5 bytes
logical channel = 10 (UA/I, Start L2CAP message)
flow = 1
data byte 1 = 00000001
data byte 2 = 00000010
data byte 3 = 00000011
data byte 4 = 00000100
data byte 5 = 00000101

HEC and CRC initialization: (MSB...LSB)
---------------------------------------
uap = 01000111

NO WHITENING USED




AIR DATA (LSB...MSB)

Packet header (incl HEC):
-------------------------
111111000
000000111000
000111000
000111111000000000000000

Payload (incl payload header and CRC):
--------------------------------------------------------------
01110100
10000000
01000000
11000000
00100000
10100000
1110110000110110
```

6.2 EXAMPLE OF DM1 PACKET

```
Packet header: (MSB...LSB)
--------------------------
AM_ADDR = 011
TYPE = 0011 (DM1)
FLOW = 0
ARQN = 1
SEQN = 0

Payload: (MSB...LSB)
--------------------
payload length: 5 bytes
logical channel = 10 (UA/I, Start L2CAP message)
flow = 1
data byte 1 = 00000001
data byte 2 = 00000010
data byte 3 = 00000011
data byte 4 = 00000100
data byte 5 = 00000101

HEC and CRC initialization: (MSB...LSB)
---------------------------------------
uap = 01000111

NO WHITENING USED




AIR DATA (LSB...MSB)

Packet header (incl HEC):
-------------------------
111111000
111111000000
000111000
111000000111111111111000

Payload (incl payload header, FEC23, CRC and 6 padded zeros):
-------------------------------------------------------------
0111010010 11001
0000000100 01011
0000110000 11110
0000100000 00111
1010000011 01100
1011000011 00010
0110000000 10001
```

7 WHITENING SEQUENCE SAMPLE DATA

Whitening sequence generator.

Whitening Sequence (=D7)	Whitening LFSR D7.....D0
1	1111111
1	1101111
1	1001111
0	0001111
0	0011110
0	0111100
1	1111000
1	1100001
1	1010011
0	0110111
1	1101110
1	1001101
0	0001011
0	0010110
0	0101100
1	1011000
0	0100001
1	1000010
0	0010101
0	0101010
1	1010100
0	0111001
1	1110010
1	1110101
1	1111011
1	1100111
1	1011111
0	0101111
1	1011110
0	0101101
1	1011010
0	0100101
1	1001010
0	0000101
0	0001010
0	0010100
0	0101000
1	1010000
0	0110001
1	1100010
1	1010101
0	0111011
1	1110110

```
1       1111101
1       1101011
1       1000111
0       0011111
0       0111110
1       1111100
1       1101001
1       1000011
0       0010111
0       0101110
1       1011100
0       0101001
1       1010010
0       0110101
1       1101010
1       1000101
0       0011011
0       0110110
1       1101100
1       1001001
0       0000011
0       0000110
0       0001100
0       0011000
0       0110000
1       1100000
1       1010001
0       0110011
1       1100110
1       1011101
0       0101011
1       1010110
0       0111101
1       1111010
1       1100101
1       1011011
0       0100111
1       1001110
0       0001101
0       0011010
0       0110100
1       1101000
1       1000001
0       0010011
0       0100110
1       1001100
0       0001001
0       0010010
0       0100100
1       1001000
0       0000001
0       0000010
```

0	0000100
0	0001000
0	0010000
0	0100000
1	1000000
0	0010001
0	0100010
1	1000100
0	0011001
0	0110010
1	1100100
1	1011001
0	0100011
1	1000110
0	0011101
0	0111010
1	1110100
1	1111001
1	1100011
1	1010111
0	0111111
1	1111110
1	1101101
1	1001011
0	0000111
0	0001110
0	0011100
0	0111000
1	1110000
1	1110001
1	1110011
1	1110111
1	1111111

8 FEC SAMPLE DATA

```
================================================
 Rate 2/3 FEC -- (15,10) Shortened Hamming Code
================================================
```

Data is in hexadecimal notation, the codewords are in binary notation. The codeword bits are sent from left to right over the air interface. The space in the codeword indicates the start of parity bits.

Data:	Codeword:
0x001	1000000000 11010
0x002	0100000000 01101
0x004	0010000000 11100
0x008	0001000000 01110
0x010	0000100000 00111
0x020	0000010000 11001
0x040	0000001000 10110
0x080	0000000100 01011
0x100	0000000010 11111
0x200	0000000001 10101

9 ENCRYPTION KEY SAMPLE DATA

Explanation:

For the sections 9.1 - 9.5, the hexadecimal sample data is written with the least significant byte at the leftmost position and the most significant byte at the rightmost position. Within each byte, the *least significant bit* (LSB) is at the rightmost position and the *most significant bit* (MSB) is at the leftmost position Thus, a line reading:

aco: 48afcdd4bd40fef76693b113

means aco[0]=0x48, ac[1]=0xaf, ..., aco[11]=0x13. The LSB of aco[11] is ‘1’ and the MSB of aco[11] is ‘0’.

Key [i]: denotes the ith sub-key in Ar or A'r;
round r: denotes the input to the rth round;
added ->: denotes the input to round 3 in
A'r after adding original input (of round 1).

9.1 FOUR TESTS OF E1

```
rand    :00000000000000000000000000000000
address :000000000000
key     :00000000000000000000000000000000
round  1:00000000000000000000000000000000
Key [ 1]:00000000000000000000000000000000
Key [ 2]:4697b1baa3b7100ac537b3c95a28ac64
round  2:78d19f9307d2476a523ec7a8a026042a
Key [ 3]:ecabaac66795580df89af66e66dc053d
Key [ 4]:8ac3d8896ae9364943bfebd4969b68a0
round  3:600265247668dda0e81c07bbb30ed503
Key [ 5]:5d57921fd5715cbb22c1be7bbc996394
Key [ 6]:2a61b8343219fdfb1740e6511d41448f
round  4:d7552ef7cc9dbde568d80c2215bc4277
Key [ 7]:dd0480dee731d67f01a2f739da6f23ca
Key [ 8]:3ad01cd1303e12a1cd0fe0a8af82592c
round  5:fb06bef32b52ab8f2a4f2b6ef7f6d0cd
Key [ 9]:7dadb2efc287ce75061302904f2e7233
Key [10]:c08dcfa981e2c4272f6c7a9f52e11538
round  6:b46b711ebb3cf69e847a75f0ab884bdd
Key [11]:fc2042c708e409555e8c147660ffdfd7
Key [12]:fa0b21001af9a6b9e89e624cd99150d2
round  7:c585f308ff19404294f06b292e978994
Key [13]:18b40784ea5ba4c80ecb48694b4e9c35
Key [14]:454d54e5253c0c4a8b3fcca7db6baef4
round  8:2665fadb13acf952bf74b4ab12264b9f
Key [15]:2df37c6d9db52674f29353b0f011ed83
Key [16]:b60316733b1e8e70bd861b477e2456f1
Key [17]:884697b1baa3b7100ac537b3c95a28ac
```

```
round  1:158ffe43352085e8a5ec7a88e1ff2ba8
Key [ 1]:e9e5dfc1b3a79583e9e5dfc1b3a79583
Key [ 2]:7595bf57e0632c59f435c16697d4c864
round  2:0b5cc75febcdf7827ca29ec0901b6b5b
Key [ 3]:e31b96afcc75d286ef0ae257cbbc05b7
Key [ 4]:0d2a27b471bc0108c6263aff9d9b3b6b
round  3:e4278526c8429211f7f2f0016220aef4
added ->:f1b68365fd6217f952de6a89831fd95c
Key [ 5]:98d1eb5773cf59d75d3b17b3bc37c191
Key [ 6]:fd2b79282408ddd4ea0aa7511133336f
round  4:d0304ad18337f86040145d27aa5c8153
Key [ 7]:331227756638a41d57b0f7e071ee2a98
Key [ 8]:aa0dd8cc68b406533d0f1d64aabacf20
round  5:84db909d213bb0172b8b6aaf71bf1472
Key [ 9]:669291b0752e63f806fce76f10e119c8
Key [10]:ef8bdd46be8ee0277e9b78adef1ec154
round  6:f835f52921e903dfa762f1df5abd7f95
Key [11]:f3902eb06dc409cfd78384624964bf51
Key [12]:7d72702b21f97984a721c99b0498239d
round  7:ae6c0b4bb09f25c6a5d9788a31b605d1
Key [13]:532e60bceaf902c52a06c2c283ecfa32
Key [14]:181715e5192efb2a64129668cf5d9dd4
round  8:744a6235b86cc0b853cc9f74f6b65311
Key [15]:83017c1434342d4290e961578790f451
Key [16]:2603532f365604646ff65803795ccce5
Key [17]:882f7c907b565ea58dae1c928a0dcf41
sres    :056c0fe6
aco     :48afcdd4bd40fef76693b113
----------------------------------------
rand    :bc3f30689647c8d7c5a03ca80a91eceb
address :7ca89b233c2d
key     :159dd9f43fc3d328efba0cd8a861fa57
round  1:bc3f30689647c8d7c5a03ca80a91eceb
Key [ 1]:159dd9f43fc3d328efba0cd8a861fa57
Key [ 2]:326558b3c15551899a97790e65ff669e
round  2:3e950edf197615638cc19c09f8fedc9b
Key [ 3]:62e879b65b9f53bbfbd020c624b1d682
Key [ 4]:73415f30bac8ab61f410adc9442992db
round  3:6a7640791cb536678936c5ecd4ae5a73
Key [ 5]:5093cfa1d31c1c48acd76df030ea3c31
Key [ 6]:0b4acc2b8f1f694fc7bd91f4a70f3009
round  4:fca2c022a577e2ffb2aa007589693ec7
Key [ 7]:2ca43fc817947804ecff148d50d6f6c6
Key [ 8]:3fcd73524b533e00b7f7825bea2040a4
round  5:e97f8ea4ed1a6f4a36ffc179dc6bb563
Key [ 9]:6c67bec76ae8c8cc4d289f69436d3506
Key [10]:95ed95ee8cb97e61d75848464bffb379
round  6:38b07261d7340d028749de1773a415c7
Key [11]:ff566c1fc6b9da9ac502514550f3e9d2
Key [12]:ab5ce3f5c887d0f49b87e0d380e12f47
round  7:58241f1aed7c1c3e047d724331a0b774
Key [13]:a2cab6f95eac7d655dbe84a6cd4c47f5
```

```
Key [14]:f5caff88af0af8c42a20b5bbd2c8b460
round  8:3d1aaeff53c0910de63b9788b13c490f
Key [15]:185099c1131cf97001e2f36fda415025
Key [16]:a0ebb82676bc75e8378b189eff3f6b1d
Key [17]:cf5b348aaee27ae332b4f1bfa10289a6
round  1:2e4b417b9a2a9cfd7d8417d9a6a556eb
Key [ 1]:fe78b835f26468ab069fd3991b086fda
Key [ 2]:095c5a51c6fa6d3ac1d57fa19aa382bd
round  2:b8bca81d6bb45af9d92beadd9300f5ed
Key [ 3]:1af866df817fd9f4ec00bc704192cffc
Key [ 4]:f4a8a059c1f575f076f5fbb24bf16590
round  3:351aa16dec2c3a4787080249ed323eae
added ->:1b65e2167656d6bafa8c19904bd79445
Key [ 5]:8c9d18d9356a9954d341b4286e88ea1f
Key [ 6]:5c958d370102c9881bf753e69c7da029
round  4:2ce8fef47dda6a5bee74372e33e478a2
Key [ 7]:7eb2985c3697429fbe0da334bb51f795
Key [ 8]:af900f4b63a1138e2874bfb7c628b7b8
round  5:572787f563e1643c1c862b7555637fb4
Key [ 9]:834c8588dd8f3d4f31117a488420d69b
Key [10]:bc2b9b81c15d9a80262f3f48e9045895
round  6:16b4968c5d02853c3a43aa4cdb5f26ac
Key [11]:f08608c9e39ad3147cba61327919c958
Key [12]:2d4131decf4fa3a959084714a9e85c11
round  7:10e4120c7cccef9dd4ba4e6da8571b01
Key [13]:c934fd319c4a2b5361fa8eef05ae9572
Key [14]:4904c17aa47868e40471007cde3a97c0
round  8:f9081772498fed41b6ffd72b71fcf6c6
Key [15]:ea5e28687e97fa3f833401c86e6053ef
Key [16]:1168f58252c4ecfccafbdb3af857b9f2
Key [17]:b3440f69ef951b78b5cbd6866275301b
sres    :8d5205c5
aco     :3ed75df4abd9af638d144e94
----------------------------------------
rand    :0891caee063f5da1809577ff94ccdcfb
address :c62f19f6ce98
key     :45298d06e46bac21421ddfbed94c032b
round  1:0891caee063f5da1809577ff94ccdcfb
Key [ 1]:45298d06e46bac21421ddfbed94c032b
Key [ 2]:8f03e1e1fe1c191cad35a897bc400597
round  2:1c6ca013480a685c1b28e0317f7167e1
Key [ 3]:4f2ce3a092dde854ef496c8126a69e8e
Key [ 4]:968caee2ac6d7008c07283daec67f2f2
round  3:06b4915f5fcc1fc551a52048f0af8a26
Key [ 5]:ab0d5c31f94259a6bf85ee2d22edf56c
Key [ 6]:dfb74855c0085ce73dc17b84bfd50a92
round  4:077a92b040acc86e6e0a877db197a167
Key [ 7]:8f888952662b3db00d4e904e7ea53b5d
Key [ 8]:5e18bfcc07799b0132db88cd6042f599
round  5:7204881fb300914825fdc863e8ceadf3
Key [ 9]:bfca91ad9bd3d1a06c582b1d5512dddf
Key [10]:a88bc477e3fa1d5a59b5e6cf793c7a41
```

```
round  6:27031131d86cea2d747deb4f756143aa
Key [11]:f3cfb8dac8aea2a6a8ef95af3a2a2767
Key [12]:77beb90670c5300b03aa2b2232d3d40c
round  7:fc8c13d49149b1ce8d86f96e44a00065
Key [13]:b578373650af36a06e19fe335d726d32
Key [14]:6bcee918c7d0d24dfdf42237fcf99d53
round  8:04ef5f5a7ddf846cda0a07782fc23866
Key [15]:399f158241eb3e079f45d7b96490e7ea
Key [16]:1bcfbe98ecde2add52aa63ea79fb917a
Key [17]:ee8bc03ec08722bc2b075492873374af
round  1:d989d7a40cde7032d17b52f8117b69d5
Key [ 1]:2ecc6cc797cc41a2ab02007f6af396ae
Key [ 2]:acfaef7609c12567d537ae1cf9dc2198
round  2:8e76eb9a29b2ad5eea790db97aee37c1
Key [ 3]:079c8ff9b73d428df879906a0b87a6c8
Key [ 4]:19f2710baf403a494193d201f3a8c439
round  3:346bb7c35b2539676375aafe3af69089
added ->:edf48e675703a955b2f0fc062b71f95c
Key [ 5]:d623a6498f915cb2c8002765247b2f5a
Key [ 6]:900109093319bc30108b3d9434a77a72
round  4:fafb6c1f3ebbd2477be2da49dd923f69
Key [ 7]:e28e2ee6e72e7f4e5b5c11f10d204228
Key [ 8]:8e455cd11f8b9073a2dfa5413c7a4bc5
round  5:7c72230df588060a3cf920f9b0a08f06
Key [ 9]:28afb26e2c7a64238c41cefc16c53e74
Key [10]:d08dcafc2096395ba0d2dddd0e471f4d
round  6:55991df991db26ff00073a12baa3031d
Key [11]:fcffdcc3ad8faae091a7055b934f87c1
Key [12]:f8df082d77060252c02d91e55bd6a7d6
round  7:70ec682ff864375f63701fa4f6be5377
Key [13]:bef3706e523d708e8a44147d7508bc35
Key [14]:3e98ab283ca2422d56a56cf8b06caeb3
round  8:172f12ec933da85504b4ea5c90f8f0ea
Key [15]:87ad9625d06645d22598dd5ef811ea2c
Key [16]:8bd3db0cc8168009e5da90877e13a36f
Key [17]:0e74631d813a8351ac7039b348c41b42
sres    :00507e5f
aco     :2a5f19fbf60907e69f39ca9f
----------------------------------------
rand    :0ecd61782b4128480c05dc45542b1b8c
address :f428f0e624b3
key     :35949a914225fabad91995d226de1d92
round  1:0ecd61782b4128480c05dc45542b1b8c
Key [ 1]:35949a914225fabad91995d226de1d92
Key [ 2]:ea6b3dcccc8ee5d88de349fa5010404f
round  2:8935e2e263fbc4b9302cabdfc06bce3e
Key [ 3]:920f3a0f2543ce535d4e7f25ad80648a
Key [ 4]:ad47227edf9c6874e80ba80ebb95d2c9
round  3:b4c8b878675f184a0c72f3dab51f8f05
Key [ 5]:81a941ca7202b5e884ae8fa493ecac3d
Key [ 6]:bcde1520bee3660e86ce2f0fb78b9157
round  4:77ced9f2fc42bdd5c6312b87fc2377c5
```

```
Key [ 7]:c8eee7423d7c6efa75ecec0d2cd969d3
Key [ 8]:910b3f838a02ed441fbe863a02b4a1d0
round  5:fe28e8056f3004d60bb207e628b39cf2
Key [ 9]:56c647c1e865eb078348962ae070972d
Key [10]:883965da77ca5812d8104e2b640aec0d
round  6:1f2ba92259d9e88101518f145a33840f
Key [11]:61d4cb7e4f8868a283327806a9bd8d4d
Key [12]:9f57de3a3ff310e21dc1e696ce060304
round  7:cc9b5d0218d29037e88475152ebebb2f
Key [13]:7aa1d8adc1aeed7127ef9a18f6eb2d8e
Key [14]:b4db9da3bf865912acd14904c7f7785d
round  8:b04d352bedc02682e4a7f59d7cda1dba
Key [15]:a13d7141ef1f6c7d867e3d175467381b
Key [16]:08b2bc058e50d6141cdd566a307e1acc
Key [17]:057b2b4b4be5dc0ac49e50489b8006c9
round  1:5cfacc773bae995cd7f1b81e7c9ec7df
Key [ 1]:1e717950f5828f3930fe4a9395858815
Key [ 2]:d1623369b733d98bbc894f75866c544c
round  2:d571ffa21d9daa797b1a0a3c962fc64c
Key [ 3]:4abf27664ae364cc8a7e5bcf88214cc4
Key [ 4]:2aaedda8dc4933dd6aeaf6e5c0d5a482
round  3:e17c8e498a00f125bf654c938c23f36d
added ->:bd765a3eb1ae8a796856048df0c1bab2
Key [ 5]:bc7f8ab2d86000f47b1946cc8d7a7a2b
Key [ 6]:6b28544cb13ec6c5d98470df2cf900b7
round  4:a9727c26f2f06bd9920e83c8605dcd76
Key [ 7]:1be840d9107f2c9523f66bb19f5464a1
Key [ 8]:61d6fb1aa2f0c2b26fb2a3d6de8c177c
round  5:aeff751f146eab7e4626b2e2c9e2fb39
Key [ 9]:adabfc82570c568a233173099f23f4c2
Key [10]:b7df6b55ad266c0f1ff7452101f59101
round  6:cf412b95f454d5185e67ca671892e5bd
Key [11]:8e04a7282a2950dcbaea28f300e22de3
Key [12]:21362c114433e29bda3e4d51f803b0cf
round  7:16165722fe4e07ef88f8056b17d89567
Key [13]:710c8fd5bb3cbb5f132a7061de518bd9
Key [14]:0791de7334f4c87285809343f3ead3bd
round  8:28854cd6ad4a3c572b15490d4b81bc3f
Key [15]:4f47f0e5629a674bfcd13770eb3a3bd9
Key [16]:58a6d9a16a284cc0aead2126c79608a1
Key [17]:a564082a0a98399f43f535fd5cefad34
sres    :80e5629c
aco     :a6fe4dcde3924611d3cc6ba1


========================================
```

9.2 FOUR TESTS OF E21

```
rand    :00000000000000000000000000000000
address :000000000000
round  1:00000000000000000000000000000000
```

```
Key [ 1]:00000000000000000000000000000006
Key [ 2]:4697b1baa3b7100ac537b3c95a28dc94
round  2:98611307ab76bbde9a86af1ce8cad412
Key [ 3]:ecabaac66795580df89af66e665d863d
Key [ 4]:8ac3d8896ae9364943bfebd4a2a768a0
round  3:820999ad2e6618f4b578974beeedf9e7
added ->:820999ad2e6618f4b578974beeedf9e7
Key [ 5]:5d57921fd5715cbb22c1bedb1c996394
Key [ 6]:2a61b8343219fdfb1740e9541d41448f
round  4:acd6edec87581ac22dbdc64ea4ced3a2
Key [ 7]:dd0480dee731d67f01ba0f39da6f23ca
Key [ 8]:3ad01cd1303e12a18dcfe0a8af82592c
round  5:1c7798732f09fbfe25795a4a2fbc93c2
Key [ 9]:7dadb2efc287ce7b0c1302904f2e7233
Key [10]:c08dcfa981e2f4572f6c7a9f52e11538
round  6:c05b88b56aa70e9c40c79bb81cd911bd
Key [11]:fc2042c708658a555e8c147660ffdfd7
Key [12]:fa0b21002605a6b9e89e624cd99150d2
round  7:abacc71b481c84c798d1bdf3d62f7e20
Key [13]:18b407e44a5ba4c80ecb48694b4e9c35
Key [14]:454d57e8253c0c4a8b3fcca7db6baef4
round  8:e8204e1183ae85cf19edb2c86215b700
Key [15]:2d0b946d9db52674f29353b0f011ed83
Key [16]:76c316733b1e8e70bd861b477e2456f1
Key [17]:8e4697b1baa3b7100ac537b3c95a28ac
Ka      :d14ca028545ec262cee700e39b5c39ee
----------------------------------------
rand    :2dd9a5503431913040l3b2d7e1189d09
address :cac4364303b6
round  1:cac4364303b6cac4364303b6cac43643
Key [ 1]:2dd9a5503431913040l3b2d7e1189d0f
Key [ 2]:14c4335b2c43910c5dcc71d81a14242b
round  2:e169f788aad45a9011f11db5270b1277
Key [ 3]:55bfb712cba168d1a48f6e74cd9f4388
Key [ 4]:2a2b3aacca695caef2821b0fb48cc253
round  3:540f9c76652e92c44987c617035037bf
added ->:9ed3d23566e45c007fcac9a1c9146dfc
Key [ 5]:a06aab22d9a287384042976b4b6b00ee
Key [ 6]:c229d054bb72e8eb230e6dcdb32d16b7
round  4:83659a41675f7171ea57909dc5a79ab4
Key [ 7]:23c4812ab1905ddf77dedaed4105649a
Key [ 8]:40d87e272a7a1554ae2e85e3638cdf52
round  5:0b9382d0ed4f2fccdbb69d0db7b130a4
Key [ 9]:bdc064c6a39f6b84fe40db359f62a3c4
Key [10]:58228db841ce3cee983aa721f36aa1b9
round  6:c6ebda0f8f489792f09c189568226c1f
Key [11]:a815bacd6fa747a0d4f52883ac63ebe7
Key [12]:a9ce513b38ea006c333ecaaefcf1d0f8
round  7:75a8aba07e69c9065bcd831c40115116
Key [13]:3635e074792d4122130e5b824e52cd60
Key [14]:511bdb61bb28de72a5d794bffbf407df
round  8:57a6e279dcb764cf7dd6a749dd60c735
```

```
Key [15]:a32f5f21044b6744b6d913b13cdb4c0a
Key [16]:9722bbaeef281496ef8c23a9d41e92f4
Key [17]:807370560ad7e8a13a054a65a03b4049
Ka      :e62f8bac609139b3999aedbc9d228042
----------------------------------------
rand    :dab3cffe9d5739d1b7bf4a667ae5ee24
address :02f8fd4cd661
round  1:02f8fd4cd66102f8fd4cd66102f8fd4c
Key [ 1]:dab3cffe9d5739d1b7bf4a667ae5ee22
Key [ 2]:e315a8a65d809ec7c289e69c899fbdcc
round  2:ef85ff081b8709405e19f3e275cec7dc
Key [ 3]:df6a119bb50945fc8a3394e7216448f3
Key [ 4]:87fe86fb0d58b5dd0fb3b6b1dab51d07
round  3:aa25c21bf577d92dd97381e3e9edcc54
added ->:a81dbf5723d8dbd524bf5782ebe5c918
Key [ 5]:36cc253c506c0021c91fac9d8c469e90
Key [ 6]:d5fda00f113e303809b7f7d78a1a2b0e
round  4:9e69ce9b53caec3990894d2baed41e0d
Key [ 7]:c14b5edc10cabf16bc2a2ba4a8ae1e40
Key [ 8]:74c6131afc8dce7e11b03b1ea8610c16
round  5:a5460fa8cedca48a14fd02209e01f02e
Key [ 9]:346cfc553c6cbc9713edb55f4dcbc96c
Key [10]:bddf027cb059d58f0509f8963e9bdec6
round  6:92b33f11eadcacc5a43dd05f13d334dd
Key [11]:8eb9e040c36c4c0b4a7fd3dd354d53c4
Key [12]:c6ffecdd5e135b20879b9dfa4b34bf51
round  7:fb0541aa5e5df1a61c51aef606eb5a41
Key [13]:bf12f5a6ba08dfc4fda4bdfc68c997d9
Key [14]:37c4656b9215f3c959ea688fb64ad327
round  8:f0bbd2b94ae374346730581fc77a9c98
Key [15]:e87bb0d86bf421ea4f779a8eee3a866c
Key [16]:faa471e934fd415ae4c0113ec7f0a5ad
Key [17]:95204a80b8400e49db7cf6fd2fd40d9a
Ka      :b0376d0a9b338c2e133c32b69cb816b3
----------------------------------------
rand    :13ecad08ad63c37f8a54dc56e82f4dc1
address :9846c5ead4d9
round  1:9846c5ead4d99846c5ead4d99846c5ea
Key [ 1]:13ecad08ad63c37f8a54dc56e82f4dc7
Key [ 2]:ad04f127bed50b5e671d6510d392eaed
round  2:97374e18cdd0a6f7a5aa49d1ac875c84
Key [ 3]:57ad159e5774fa222f2f3039b9cd5101
Key [ 4]:9a1e9e1068fede02ef90496e25fd8e79
round  3:9dd3260373edd9d5f4e774826b88fd2d
added ->:0519ebe9a7c6719331d1485bf3cec2c7
Key [ 5]:378dce167db62920b0b392f7cfca316e
Key [ 6]:db4277795c87286faee6c9e9a6b71a93
round  4:40ec6563450299ac4e120d88672504d6
Key [ 7]:ec01aa2f5a8a793b36c1bb858d254380
Key [ 8]:2921a66cfa5bf74ac535424564830e98
round  5:57287bbb041bd6a56c2bd931ed410cd4
Key [ 9]:07018e45aab61b3c3726ee3d57dbd5f6
```

```
Key [10]:627381f0fa4c02b0c7d3e7dfbffc3333
round  6:66affa66a8dcd36e36bf6c3f1c6a276e
Key [11]:33b57c925bd5551999f716e138efbe79
Key [12]:a6dc7f9aa95bcc9243aebd12608f657a
round  7:450e65184fd8c72c578d5cdecb286743
Key [13]:a6a6db00fd8c72a28ea57ea542f6e102
Key [14]:dcf3377daeb2e24e61f0ad6620951c1f
round  8:e5eb180b519a4e673f21b7c4f4573f3d
Key [15]:621240b9506b462a7fa250da41844626
Key [16]:ae297810f01f43dc35756cd119ee73d6
Key [17]:b959835ec2501ad3894f8b8f1f4257f9
Ka      :5b61e83ad04d23e9d1c698851fa30447
========================================
```

9.3 THREE TESTS OF E22

(for K_master and overlay generation)

```
rand    :001de169248850245a5f7cc7f0d6d633
PIN     :d5a51083a04a1971f18649ea8b79311a
round  1:001de169248850245a5f7cc7f0d6d623
Key [ 1]:d5a51083a04a1971f18649ea8b79311a
Key [ 2]:7317cdbff57f9b99f9810a2525b17cc7
round  2:5f05c143347b59acae3cb002db23830f
Key [ 3]:f08bd258adf1d4ae4a54d8ccb26220b2
Key [ 4]:91046cbb4ccc43db18d6dd36ca7313eb
round  3:c8f3e3300541a25b6ac5a80c3105f3c4
added ->:c810c45921c9f27f302424cbc1dbc9e7
Key [ 5]:67fb2336f4d9f069da58d11c82f6bd95
Key [ 6]:4fed702c75bd72c0d3d8f38707134c50
round  4:bd5e0c3a97fa55b91a3bbbf306ebb978
Key [ 7]:41c947f80cdc0464c50aa89070af314c
Key [ 8]:680eecfa8daf41c7109c9a5cb1f26d75
round  5:21c1a762c3cc33e75ce8976a73983087
Key [ 9]:6e33fbd94d00ff8f72e8a7a0d2cebc4c
Key [10]:f4d726054c6b948add99fabb5733ddc3
round  6:56d0df484345582f6b574a449ba155eb
Key [11]:4eda2425546a24cac790f49ef2453b53
Key [12]:cf2213624ed1510408a5a3e00b7333df
round  7:120cf9963fe9ff22993f7fdf9600d9b8
Key [13]:d04b1a25b0b8fec946d5ecfa626d04c9
Key [14]:01e5611b0f0e140bdb64585fd3ae5269
round  8:a6337400ad8cb47fefb91332f5cb2713
Key [15]:f15b2dc433f534f61bf718770a3698b1
Key [16]:f990d0273d8ea2b9e0b45917a781c720
Key [17]:f41b3cc13d4301297bb6bdfcb3e5a1dd
Ka      :539e4f2732e5ae2de1e0401f0813bd0d
----------------------------------------
rand    :67ed56bfcf99825f0c6b349369da30ab
PIN     :7885b515e84b1f082cc499976f1725ce
round  1:67ed56bfcf99825f0c6b349369da30bb
Key [ 1]:7885b515e84b1f082cc499976f1725ce
```

```
Key [ 2]:72445901fdaf506beb036f4412512248
round  2:6b160b66a1f6c26c1f3432f463ef5aa1
Key [ 3]:59f0e4982e97633e5e7fd133af8f2c5b
Key [ 4]:b4946ec77a41bf7c729d191e33d458ab
round  3:3f22046c964c3e5ca2a26ec9a76a9f67
added ->:580f5ad359e5c003ae0da25ace44cfdc
Key [ 5]:eb0b839f97bdf534183210678520bbef
Key [ 6]:cff0bc4a94e5c8b2a2d24d9f59031e19
round  4:87aa61fc0ff88e744c195249b9a33632
Key [ 7]:592430f14d8f93db95dd691af045776d
Key [ 8]:3b55b404222bf445a6a2ef5865247695
round  5:83dcf592a854226c4dcd94e1ecf1bc75
Key [ 9]:a9714b86319ef343a28b87456416bd52
Key [10]:e6598b24390b3a0bf2982747993b0d78
round  6:dee0d13a52e96bcf7c72045a21609fc6
Key [11]:62051d8c51973073bff959b032c6e1e2
Key [12]:29e94f4ab73296c453c833e217a1a85b
round  7:08488005761e6c7c4dbb203ae453fe3a
Key [13]:0e255970b3e2fc235f59fc5acb10e8ce
Key [14]:d0dfbb3361fee6d4ffe45babf1cd7abf
round  8:0d81e89bddde7a7065316c47574feb8f
Key [15]:c12eee4eb38b7a171f0f736003774b40
Key [16]:8f962523f1c0abd9a087a0dfb11643d3
Key [17]:24be1c66cf8b022f12f1fb4c60c93fd1
Ka      :04435771e03a9daceb8bb1a493ee9bd8
----------------------------------------
rand    :40a94509238664f244ff8e3d13b119d3
PIN     :1ce44839badde30396d03c4c36f23006
round  1:40a94509238664f244ff8e3d13b119c3
Key [ 1]:1ce44839badde30396d03c4c36f23006
Key [ 2]:6dd97a8f91d628be4b18157af1a9dcba
round  2:0eac5288057d9947a24eabc1744c4582
Key [ 3]:fef9583d5f55fd4107ad832a725db744
Key [ 4]:fc3893507016d7c1db2bd034a230a069
round  3:60b424f1082b0cc3bd61be7b4c0155f0
added ->:205d69f82bb17031f9604c465fb26e33
Key [ 5]:0834d04f3e7e1f7f85f0c1db685ab118
Key [ 6]:1852397f9a3723169058e9b62bb3682b
round  4:2c6b65a49d66af6566675afdd6fa7d7d
Key [ 7]:6c10da21d762ae4ac1ba22a96d9007b4
Key [ 8]:9aa23658b90470a78d686344b8a9b0e7
round  5:a2c537899665113a42f1ac24773bdc31
Key [ 9]:137dee3bf879fe7bd02fe6d888e84f16
Key [10]:466e315a1863f47d0f93bc6827cf3450
round  6:e26982980d79b21ed3e20f8c3e71ba96
Key [11]:0b33cf831465bb5c979e6224d7f79f7c
Key [12]:92770660268ede827810d707a0977d73
round  7:e7b063c4e2e3110b89b7e1631c762dd5
Key [13]:7be30ae4961cf24ca17625a77bb7a9f8
Key [14]:be65574a33ae30e6e82dbd2826d3cc1a
round  8:7a963e37b2c2e76b489cfe40a2cf00e5
Key [15]:ed0ba7dd30d60a5e69225f0a33011e5b
```

Appendix IV - Sample Data

```
Key [16]:765c990f4445e52b39e6ed6105ad1c4f
Key [17]:52627bf9f35d94f30d5b07ef15901adc
Ka      :9cde4b60f9b5861ed9df80858bac6f7f


========================================
```

9.4 TESTS OF E22 WITH PIN AUGMENTING

for PIN lengths 1,...,16 bytes

```
rand    :24b101fd56117d42c0545a4247357048
PIN length =16 octets
PIN     :fd397c7f5c1f937cdf82d8816cc377e2
round  1:24b101fd56117d42c0545a4247357058
Key [ 1]:fd397c7f5c1f937cdf82d8816cc377e2
Key [ 2]:0f7aac9c9b53f308d9fdbf2c78e3c30e
round  2:838edfe1226266953ccba8379d873107
Key [ 3]:0b8ac18d4bb44fad2efa115e43945abc
Key [ 4]:887b16b062a83bfa469772c25b456312
round  3:8cd0c9283120aba89a7f9d635dd4fe3f
added ->:a881cad5673128ea5ad3f7211a096e67
Key [ 5]:2248cbe6d299e9d3e8fd35a91178f65b
Key [ 6]:b92af6237385bd31f8fb57fb1bdd824e
round  4:2648d9c618a622b10ef80c4dbaf68b99
Key [ 7]:2bf5ffe84a37878ede2d4c30be60203b
Key [ 8]:c9cb6cec60cb8a8f29b99fcf3e71e40f
round  5:b5a7d9e96f68b14ccebf361de3914d0f
Key [ 9]:5c2f8a702e4a45575b103b0cce8a91c6
Key [10]:d453db0c9f9ddbd11e355d9a34d9b11b
round  6:632a091e7eefe1336857ddafd1ff3265
Key [11]:32805db7e59c5ed4acabf38d27e3fece
Key [12]:fde3a8eedfa3a12be09c1a8a00890fd7
round  7:048531e9fd3efa95910540150f8b137b
Key [13]:def07eb23f3a378f059039a2124bc4c2
Key [14]:2608c58f23d84a09b9ce95e5caac1ab4
round  8:461814ec7439d412d0732f0a6f799a6a
Key [15]:0a7ed16481a623e56ee1442ffa74f334
Key [16]:12add59aca0d19532f1516979954e369
Key [17]:dd43d02d39ffd6a386a4b98b4ac6eb23
Ka      :a5f2adf328e4e6a2b42f19c8b74ba884
----------------------------------------
rand    :321964061ac49a436f9fb9824ac63f8b
PIN length =15 octets
PIN     :ad955d58b6b8857820ac1262d617a6
address :0314c0642543
round  1:321964061ac49a436f9fb9824ac63f9b
Key [ 1]:ad955d58b6b8857820ac1262d617a603
Key [ 2]:f281736f68e3d30b2ac7c67f125dc416
round  2:7c4a4ece1398681f4bafd309328b7770
Key [ 3]:43c157f4c8b360387c32ab330f9c9aa8
Key [ 4]:3a3049945a298f6d076c19219c47c3cb
round  3:9672b00738bdfaf9bd92a855bc6f3afb
added ->:a48b1401228194bad23161d7f6357960
Key [ 5]:c8e2eaa6d73b7de18f3228ab2173bc69
Key [ 6]:8623f44488222e66a293677cf30bf2bb
round  4:9b30247aad3bf133712d034b46d21c68
Key [ 7]:f3e500902fba31db9bae50ef30e484a4
Key [ 8]:49d4b1137c18f4752dd9955a5a8d2f43
```

```
round  5:4492c25fda08083a768b4b5588966b23
Key [ 9]:9d59c451989e74785cc097eda7e42ab8
Key [10]:251de25f3917dcd99c18646107a641fb
round  6:21ae346635714d2623041f269978c0ee
Key [11]:80b8f78cb1a49ec0c3e32a238e60fddf
Key [12]:beb84f4d20a501e4a24ecfbde481902b
round  7:9b56a3d0f8932f20c6a77a229514fb00
Key [13]:852571b44f35fd9d9336d3c1d2506656
Key [14]:d0a0d510fb06ba76e69b8ee3ebc1b725
round  8:6cd8492b2fd31a86978bcdf644eb08a8
Key [15]:c7ffd523f32a874ed4a93430a25976de
Key [16]:16cdcb25e62964876d951fdcc07030d3
Key [17]:def32c0e12596f9582e5e3c52b303f52
Ka      :c0ec1a5694e2b48d54297911e6c98b8f
----------------------------------------
rand    :d4ae20c80094547d7051931b5cc2a8d6
PIN length =14 octets
PIN     :e1232e2c5f3b833b3309088a87b6
address :fabecc58e609
round  1:d4ae20c80094547d7051931b5cc2a8c6
Key [ 1]:e1232e2c5f3b833b3309088a87b6fabe
Key [ 2]:5f0812b47cd3e9a30d7707050fffa1f2
round  2:1f45f16be89794bef33e4547c9c0916a
Key [ 3]:77b681944763244ffa3cd71b248b79b5
Key [ 4]:e2814e90e04f485958ce58c9133e2be6
round  3:b10d2f4ac941035263cee3552d774d2f
added ->:65bb4f82c9d5572f131f764e7139f5e9
Key [ 5]:520acad20801dc639a2c6d66d9b79576
Key [ 6]:c72255cdb61d42be72bd45390dd25ba5
round  4:ead4dc34207b6ea721c62166e155aaad
Key [ 7]:ebf04c02075bf459ec9c3ec06627d347
Key [ 8]:a1363dd2812ee800a4491c0c74074493
round  5:f507944f3018e20586d81d7f326aae9d
Key [ 9]:b0b6ba79493dc833d7f425be7b8dadb6
Key [10]:08cd23e536b9b9b53e85eb004cba3111
round  6:fff450f4302a2b3571e8405e148346da
Key [11]:fec22374c6937dcd26171f4d2edfada3
Key [12]:0f1a8ef5979c69ff44f620c2e007b6e4
round  7:de558779589897f3402a90ee78c3f921
Key [13]:901fb66f0779d6aad0c0fba1fe812cb5
Key [14]:a0cab3cd15cd23603adc8d4474efb239
round  8:b2df0aa0c9f07fbbaa02f510a29cf540
Key [15]:18edc3f4296dd6f1dea13f7c143117a1
Key [16]:8d3d52d700a379d72ded81687f7546c7
Key [17]:5927badfe602f29345f840bb53e1dea6
Ka      :d7b39be13e3692c65b4a9e17a9c55e17
----------------------------------------
rand    :272b73a2e40db52a6a61c6520549794a
PIN length =13 octets
PIN     :549f2694f353f5145772d8ae1e
address :20487681eb9f
round  1:272b73a2e40db52a6a61c6520549795a
```

```
Key [ 1]:549f2694f353f5145772d8ae1e204876
Key [ 2]:42c855593d66b0c458fd28b95b6a5fbf
round  2:d7276dc8073f7677c31f855bde9501e2
Key [ 3]:75d0a69ae49a2da92e457d767879df52
Key [ 4]:b3aa7e7492971afaa0fb2b64827110df
round  3:71aae503831133d19bc452da4d0e409b
added ->:56d558a1671ee8fbf12518884857b9c1
Key [ 5]:9c8cf1604a98e9a503c342e272de5cf6
Key [ 6]:d35bc2df6b85540a27642106471057d9
round  4:f41a709c89ea80481aa3d2b9b2a9f8ca
Key [ 7]:b454dda74aeb4eff227ba48a58077599
Key [ 8]:bcba6aec050116aa9b7c6a9b7314d796
round  5:20fdda20f4a26b1bd38eb7f355a7be87
Key [ 9]:d41f8a9de0a716eb7167a1b6e321c528
Key [10]:5353449982247782d168ab43f17bc4d8
round  6:a70e316997eeed49a5a9ef9ba5e913b5
Key [11]:32cbc9cf1a81e36a45153972347ce4ac
Key [12]:5747619006cf4ef834c749f2c4b9feb6
round  7:e66f2317a825f589f76b47b6aa6e73fb
Key [13]:f9b68beba0a09d2a570a7dc88cc3c3c2
Key [14]:55718f9a4f0b1f9484e8c6b186a41a4b
round  8:5f68f940440a9798e074776019804ada
Key [15]:4ecc29be1b4d78433f6aa30db974a7fb
Key [16]:8470a066ffb00cda7b08059599f919f5
Key [17]:f39a36d74e960a051e1ca98b777848f4
Ka      :9ac64309a37c25c3b4a584fc002a1618
----------------------------------------
rand    :7edb65f01a2f45a2bc9b24fb3390667e
PIN length =12 octets
PIN     :2e5a42797958557b23447ca8
address :04f0d2737f02
round  1:7edb65f01a2f45a2bc9b24fb3390666e
Key [ 1]:2e5a42797958557b23447ca804f0d273
Key [ 2]:18a97c856561eb23e71af8e9e1be4799
round  2:3436e12db8ffdc1265cb5a86da2fed0b
Key [ 3]:7c0908dcbc73201e17c4f7aa1ab8aec8
Key [ 4]:7cb58833602fbe4194c7cc797ce8c454
round  3:caed6af4226f67e4ad1914620803ef2a
added ->:b4c8cf04389eac4611b438993b935544
Key [ 5]:f4dce7d607b5234562d0ebb2267b08b8
Key [ 6]:560b75c5545751fd8fa99fa4346e654b
round  4:ee67c87d6f74bb75db98f68bff0192c1
Key [ 7]:32f10cefd8d3e6424c6f91f1437808af
Key [ 8]:a934a46045be30fb3be3a5f3f7b18837
round  5:792398dcbeb8d10bdb07ae3c819e943c
Key [ 9]:a0f12e97c677a0e8ac415cd2c8a7ca88
Key [10]:e27014c908785f5ca03e8c6a1da3bf13
round  6:e778b6e0c3e8e7edf90861c7916d97a8
Key [11]:1b4a4303bcc0b2e0f41c72d47654bd9f
Key [12]:4b1302a50046026d6c9054fc8387965a
round  7:1fafddc7efa5f04c1dec1869d3f2d9bb
Key [13]:58c334bb543d49eca562cdbe0280e0fc
```

```
Key [14]:bdb60d383c692d06476b76646c8dec48
round  8:3d7c326d074bd6aa222ea050f04a3c7f
Key [15]:78c0162506be0b5953e8403c01028f93
Key [16]:24d7dbbe834dbd7b67f57fcf0d39d60f
Key [17]:2e74f1f3331c0f6585e87b2f715e187e
Ka      :d3af4c81e3f482f062999dee7882a73b
----------------------------------------
rand    :26a92358294dce97b1d79ec32a67e81a
PIN length =11 octets
PIN     :05fbad03f52fa9324f7732
address :b9ac071f9d70
round  1:26a92358294dce97b1d79ec32a67e80a
Key [ 1]:05fbad03f52fa9324f7732b9ac071f9d
Key [ 2]:2504c9691c04a18480c8802e922098c0
round  2:0be20e3d76888e57b6bf77f97a8714fb
Key [ 3]:576b2791d1212bea8408212f2d43e77e
Key [ 4]:90ae36dcce8724adb618f912d1b27297
round  3:1969667060764453257d906b7e58bd5b
added ->:3f12892849c312c494542ea854bfa551
Key [ 5]:bc492c42c9e87f56ec31af5474e9226e
Key [ 6]:c135d1dbed32d9519acfb4169f3e1a10
round  4:ac404205118fe771e54aa6f392da1153
Key [ 7]:83ccbdbbaf17889b7d18254dc9252fa1
Key [ 8]:80b90a1767d3f2848080802764e21711
round  5:41795e89ae9a0cf776ffece76f47fd7a
Key [ 9]:cc24e4a86e8eed129118fd3d5223a1dc
Key [10]:7b1e9c0eb9dab083574be7b7015a62c9
round  6:29ca9e2f87ca00370ef1633505bfba4b
Key [11]:888e6d88cf4beb965cf7d4f32b696baa
Key [12]:6d642f3e5510b0b043a44daa2cf5eec0
round  7:81fc891c3c6fd99acc00028a387e2366
Key [13]:e224f85da2ab63a23e2a3a036e421358
Key [14]:c8dc22aaa739e2cb85d6a0c08226c7d0
round  8:e30b537e7a000e3d2424a9c0f04c4042
Key [15]:a969aa818c6b324bae391bedcdd9d335
Key [16]:6974b6f2f07e4c55f2cc0435c45bebd1
Key [17]:134b925ebd98e6b93c14aee582062fcb
Ka      :be87b44d079d45a08a71d15208c5cb50
----------------------------------------
rand    :0edef05327eab5262430f21fc91ce682
PIN length =10 octets
PIN     :8210e47390f3f48c32b3
address :7a3cdfe377d1
round  1:0edef05327eab5262430f21fc91ce692
Key [ 1]:8210e47390f3f48c32b37a3cdfe377d1
Key [ 2]:c6be4c3e425e749b620a94c779e33a7e
round  2:07ca3c7a7a6bcbc31d79a856d9cffc0e
Key [ 3]:2587cec2a4b8e4f996a9ed664350d5dd
Key [ 4]:70e4bf72834d9d3dbb7eb2c239216dc0
round  3:792ad2ac4e4559d1463714d2f161b6f4
added ->:7708c2ff692f0ef7626706cd387d9c66
Key [ 5]:6696e1e7f8ac037e1fff3598f0c164e2
```

```
Key [ 6]:23dbfe4d0b561bea08fbcef25e49b648
round  4:7d8c71a9d7fbdcbd851bdf074550b100
Key [ 7]:b03648acd021550edee904431a02f00c
Key [ 8]:cb169220b7398e8f077730aa4bf06baa
round  5:b6fcaa45064ffd557e4b7b30cfbb83e0
Key [ 9]:af602c2ba16a454649951274c2be6527
Key [10]:5d60b0a7a09d524143eca13ad680bc9c
round  6:b3416d391a0c26c558843debd0601e9e
Key [11]:9a2f39bfe558d9f562c5f09a5c3c0263
Key [12]:72cae8eebd7fabd9b1848333c2aab439
round  7:abe4b498d9c36ea97b8fd27d7f813913
Key [13]:15f27ea11e83a51645d487b81371d7dc
Key [14]:36083c8666447e03d33846edf444eb12
round  8:8032104338a945ba044d102eabda3b22
Key [15]:0a3a8977dd48f3b6c1668578befadd02
Key [16]:f06b6675d78ca0ee5b1761bdcdab516d
Key [17]:cbc8a7952d33aa0496f7ea2d05390b23
Ka      :bf0706d76ec3b11cce724b311bf71ff5
----------------------------------------
rand    :86290e2892f278ff6c3fb917b020576a
PIN length = 9 octets
PIN     :3dcdffcfd086802107
address :791a6a2c5cc3
round  1:86290e2892f278ff6c3fb917b0205765
Key [ 1]:3dcdffcfd086802107791a6a2c5cc33d
Key [ 2]:b4962f40d7bb19429007062a3c469521
round  2:1ec59ffd3065f19991872a7863b0ef02
Key [ 3]:eb9ede6787dd196b7e340185562bf28c
Key [ 4]:2964e58aacf7287d1717a35b100ae23b
round  3:f817406f1423fc2fe33e25152679eaaf
added ->:7e404e47861574d08f7dde02969941ca
Key [ 5]:6abf9a314508fd61e486fa4e376c3f93
Key [ 6]:6da148b7ee2632114521842cbb274376
round  4:e9c2a8fac22b8c7cf0c619e2b3f890ed
Key [ 7]:df889cc34fda86f01096d52d116e620d
Key [ 8]:5eb04b147dc39d1974058761ae7b73fc
round  5:444a8aac0efee1c02f8d38f8274b7b28
Key [ 9]:8426cc59eee391b2bd50cf8f1efef8b3
Key [10]:8b5d220a6300ade418da791dd8151941
round  6:9185f983db150b1bccab1e5c12eb63a1
Key [11]:82ba4ddef833f6a4d18b07aa011f2798
Key [12]:ce63d98794682054e73d0359dad35ec4
round  7:5eded2668f5916dfd036c09e87902886
Key [13]:da794357652e80c70ad8b0715dbe33d6
Key [14]:732ef2c0c3220b31f3820c375e27bb29
round  8:88a5291b4acbba009a85b7dd6a834b3b
Key [15]:3ce75a61d4b465b70c95d7ccd5799633
Key [16]:5df9bd2c3a17a840cdaafb76c171db7c
Key [17]:3f8364b089733d902bccb0cd3386846f
Ka      :cdb0cc68f6f6fbd70b46652de3ef3ffb
----------------------------------------
rand    :3ab52a65bb3b24a08eb6cd284b4b9d4b
```

Appendix IV - Sample Data

```
PIN length = 8 octets
PIN     :d0fb9b6838d464d8
address :25a868db91ab
round  1:3ab52a65bb3b24a08eb6cd284b4b9d45
Key [ 1]:d0fb9b6838d464d825a868db91abd0fb
Key [ 2]:2573f47b49dad6330a7a9155b7ae8ba1
round  2:ad2ffdff408fcfab44941016a9199251
Key [ 3]:d2c5b8fb80cba13712905a589adaee71
Key [ 4]:5a3381511b338719fae242758dea0997
round  3:2ddc17e570d7931a2b1d13f6ace928a5
added ->:17914180cb12b7baa5d3e0dee734c5e0
Key [ 5]:e0a4d8ac27fbe2783b7bcb3a36a6224d
Key [ 6]:949324c6864deac3eca8e324853e11c3
round  4:62c1db5cf31590d331ec40ad692e8df5
Key [ 7]:6e67148088a01c2d4491957cc9ddc4aa
Key [ 8]:557431deab7087bb4c03fa27228f60c6
round  5:9c8933bc361f4bde4d1bda2b5f8bb235
Key [ 9]:a2551aca53329e70ade3fd2bb7664697
Key [10]:05d0ad35de68a364b54b56e2138738fe
round  6:9156db34136aa06655bf28a05be0596a
Key [11]:1616a6b13ce2f2895c722e8495181520
Key [12]:b12e78a1114847b01f6ed2f5a1429a23
round  7:84dcc292ed836c1c2d523f2a899a2ad5
Key [13]:316e144364686381944e95afd8a026bb
Key [14]:1ab551b88d39d97ea7a9fe136dbfe2e1
round  8:87bdcac878d777877f4eccf042cfee5e
Key [15]:70e21ab08c23c7544524b64492b25cc9
Key [16]:35f730f2ae2b950a49a1bf5c8b9f8866
Key [17]:2f16924c22db8b74e2eadf1ba4ebd37c
Ka      :983218718ca9aa97892e312d86dd9516
----------------------------------------
rand    :a6dc447ff08d4b366ff96e6cf207e179
PIN length = 7 octets
PIN     :9c57e10b4766cc
address :54ebd9328cb6
round  1:a6dc447ff08d4b366ff96e6cf207e174
Key [ 1]:9c57e10b4766cc54ebd9328cb69c57e1
Key [ 2]:00a609f4d61db26993c8177e3ee2bba8
round  2:1ed26b96a306d7014f4e5c9ee523b73d
Key [ 3]:646d7b5f9aaa528384bda3953b542764
Key [ 4]:a051a42212c0e9ad5c2c248259aca14e
round  3:a53f526db18e3d7d53edbfc9711041ed
added ->:031b9612411b884b3ce62da583172299
Key [ 5]:d1bd5e64930e7f838d8a33994462d8b2
Key [ 6]:5dc7e2291e32435665ebd6956bec3414
round  4:9438be308ec83f35c560e2796f4e0559
Key [ 7]:10552f45af63b0f15e2919ab37f64fe7
Key [ 8]:c44d5717c114a58b09207392ebe341f8
round  5:b79a7b14386066d339f799c40479cb3d
Key [ 9]:6886e47b782325568eaf59715a75d8ff
Key [10]:8e1e335e659cd36b132689f78c147bda
round  6:ef232462228aa166438d10c34e17424b
```

```
Key [11]:8843efeedd5c2b7c3304d647f932f4d1
Key [12]:13785aaedd0adf67abb4f01872392785
round  7:02d133fe40d15f1073673b36bba35abd
Key [13]:837d7ca2722419e6be3fae35900c3958
Key [14]:93f8442973e7fccf2e7232d1d057c73a
round  8:275506a3d08c84e94cc58ed60054505e
Key [15]:8a7a9edffa3c52918bc6a45f57d91f5d
Key [16]:f214a95d777f763c56109882c4b52c84
Key [17]:10e2ee92c5ea1ddc5eb010e55510c403
Ka      :9cd6650ead86323e87cafb1ff516d1e0
----------------------------------------
rand    :3348470a7ea6cc6eb81b40472133262c
PIN length = 6 octets
PIN     :fcad169d7295
address :430d572f8842
round  1:3348470a7ea6cc6eb81b404721332620
Key [ 1]:fcad169d7295430d572f8842fcad169d
Key [ 2]:b3479d4d4fd178c43e7bc5b0c7d8983c
round  2:af976da9225066d563e10ab955e6fc32
Key [ 3]:7112462b37d82dd81a2a35d9eb43cb7c
Key [ 4]:c5a7030f8497945ac7b84600d1d161fb
round  3:d08f826ebd55a0bd7591c19a89ed9bde
added ->:e3d7c964c3fb6cd3cdac01dda820c1fe
Key [ 5]:84b0c6ef4a63e4dff19b1f546d683df5
Key [ 6]:f4023edfc95d1e79ed4bb4de9b174f5d
round  4:6cd952785630dfc7cf81eea625e42c5c
Key [ 7]:ea38dd9a093ac9355918632c90c79993
Key [ 8]:dbba01e278ddc76380727f5d7135a7de
round  5:93573b2971515495978264b88f330f7f
Key [ 9]:d4dc3a31be34e412210fafa6eca00776
Key [10]:39d1e190ee92b0ff16d92a8be58d2fa0
round  6:b3f01d5e7fe1ce6da7b46d8c389baf47
Key [11]:1eb081328d4bcf94c9117b12c5cf22ac
Key [12]:7e047c2c552f9f1414d946775fabfe30
round  7:0b833bff6106d5bae033b4ce5af5a924
Key [13]:e78e685d9b2a7e29e7f2a19d1bc38ebd
Key [14]:1b582272a3121718c4096d2d8602f215
round  8:23de0bbdc70850a7803f4f10c63b2c0f
Key [15]:8569e860530d9c3d48a0870dac33f676
Key [16]:6966b528fdd1dc222527052c8f6cf5a6
Key [17]:a34244c757154c53171c663b0b56d5c2
Ka      :98f1543ab4d87bd5ef5296fb5e3d3a21
----------------------------------------
rand    :0f5bb150b4371ae4e5785293d22b7b0c
PIN length = 5 octets
PIN     :b10d068bca
address :b44775199f29
round  1:0f5bb150b4371ae4e5785293d22b7b07
Key [ 1]:b10d068bcab44775199f29b10d068bca
Key [ 2]:aec70d1048f1bbd2c18040318a8402ad
round  2:342d2b79d7fb7cd110379742b9842c79
Key [ 3]:6d8d5cf338f29ef4420639ef488e4fa9
```

```
Key [ 4]:a1584117541b759ba6d9f7eb2bedcbba
round  3:9407e8e3e810603921bf81cfda62770a
added ->:9b6299b35c477addc437d35c088df20d
Key [ 5]:09a20676666aeed6f22176274eb433f4
Key [ 6]:840472c001add5811a054be5f5c74754
round  4:9a3ba953225a7862c0a842ed3d0b2679
Key [ 7]:fad9e45c8bf70a972fcd9bff0e8751f5
Key [ 8]:e8f30ff666dfd212263416496ff3b2c2
round  5:2c573b6480852e875df34b28a5c44509
Key [ 9]:964cdba0cf8d593f2fc40f96daf8267a
Key [10]:bcd65c11b13e1a70bcd4aafba8864fe3
round  6:21b0cc49e880c5811d24dee0194e6e9e
Key [11]:468c8548ea9653c1a10df6288dd03c1d
Key [12]:5d252d17af4b09d3f4b5f7b5677b8211
round  7:e6d6bdcd63e1d37d9883543ba86392fd
Key [13]:e814bf307c767428c67793dda2df95c7
Key [14]:4812b979fdc20f0ff0996f61673a42cc
round  8:e3dde7ce6bd7d8a34599aa04d6a760ab
Key [15]:5b1e2033d1cd549fc4b028146eb5b3b7
Key [16]:0f284c14fb8fe706a5343e3aa35af7b1
Key [17]:b1f7a4b7456d6b577fded6dc7a672e37
Ka      :c55070b72bc982adb972ed05d1a74ddb
----------------------------------------
rand    :148662a4baa73cfadb55489159e476e1
PIN length = 4 octets
PIN     :fb20f177
address :a683bd0b1896
round  1:148662a4baa73cfadb55489159e476eb
Key [ 1]:fb20f177a683bd0b1896fb20f177a683
Key [ 2]:47266cefbfa468ca7916b458155dc825
round  2:3a942eb6271c3f4e433838a5d3fcbd27
Key [ 3]:688853a6d6575eb2f6a2724b0fbc133b
Key [ 4]:7810df048019634083a2d9219d0b5fe0
round  3:9c835b98a063701c0887943596780769
added ->:8809bd3c1a0aace6d3dcdca4cf5c7d82
Key [ 5]:c78f6dcf56da1bbd413828b33f5865b3
Key [ 6]:eb3f3d407d160df3d293a76d1a513c4a
round  4:7e68c4bafa020a4a59b5a1968105bab5
Key [ 7]:d330e038d6b19d5c9bb0d7285a360064
Key [ 8]:9bd3ee50347c00753d165faced702d9c
round  5:227bad0cf0838bdb15b3b3872c24f592
Key [ 9]:9543ad0fb3fe74f83e0e2281c6d4f5f0
Key [10]:746cd0383c17e0e80e6d095a87fd0290
round  6:e026e98c71121a0cb739ef6f59e14d26
Key [11]:fa28bea4b1c417536608f11f406ea1dd
Key [12]:3aee0f4d21699df9cb8caf5354a780ff
round  7:cd6a6d8137d55140046f8991da1fa40a
Key [13]:372b71bc6d1aa6e785358044fbcf05f4
Key [14]:00a01501224c0405de00aa2ce7b6ab04
round  8:52cd7257fe8d0c782c259bcb6c9f5942
Key [15]:c7015c5c1d7c030e00897f104a006d4a
Key [16]:260a9577790c62e074e71e19fd2894df
```

```
Key [17]:c041b7a231493acd15ddcdaee94b9f52
Ka      :7ec864df2f1637c7e81f2319ae8f4671
----------------------------------------
rand    :193a1b84376c88882c8d3b4ee93ba8d5
PIN length = 3 octets
PIN     :a123b9
address :4459a44610f6
round  1:193a1b84376c88882c8d3b4ee93ba8dc
Key [ 1]:a123b94459a44610f6a123b94459a446
Key [ 2]:5f64d384c8e990c1d25080eb244dde9b
round  2:3badbd58f100831d781ddd3ccedefd3f
Key [ 3]:5abc00eff8991575c00807c48f6dbea5
Key [ 4]:127521158ad6798fb6479d1d2268abe6
round  3:0b53075a49c6bf2df2421c655fdedf68
added ->:128d22de7e3247a5decf572bb61987b4
Key [ 5]:f2a1f620448b8e56665608df2ab3952f
Key [ 6]:7c84c0af02aad91dc39209c4edd220b1
round  4:793f4484fb592e7a78756fd4662f990d
Key [ 7]:f6445b647317e7e493bb92bf6655342f
Key [ 8]:3cae503567c63d3595eb140ce60a84c0
round  5:9e46a8df925916a342f299a8306220a0
Key [ 9]:734ed5a806e072bbecb4254993871679
Key [10]:cda69ccb4b07f65e3c8547c11c0647b8
round  6:6bf9cd82c9e1be13fc58eae0b936c75a
Key [11]:c48e531d3175c2bd26fa25cc8990e394
Key [12]:6d93d349a6c6e9ff5b26149565b13d15
round  7:e96a9871471240f198811d4b8311e9a6
Key [13]:5c4951e85875d663526092cd4cbdb667
Key [14]:f19f7758f5cde86c3791efaf563b3fd0
round  8:e94ca67d3721d5fb08ec069191801a46
Key [15]:bf0c17f3299b37d984ac938b769dd394
Key [16]:7edf4ad772a6b9048588f97be25bde1c
Key [17]:6ee7ba6afefc5b561abbd8d6829e8150
Ka      :ac0daabf17732f632e34ef193658bf5d
----------------------------------------
rand    :1453db4d057654e8eb62d7d62ec3608c
PIN length = 2 octets
PIN     :3eaf
address :411fbbb51d1e
round  1:1453db4d057654e8eb62d7d62ec36084
Key [ 1]:3eaf411fbbb51d1e3eaf411fbbb51d1e
Key [ 2]:c3a1a997509f00fb4241aba607109c64
round  2:0b78276c1ebc65707d38c9c5fa1372bd
Key [ 3]:3c729833ae1ce7f84861e4dbad6305cc
Key [ 4]:c83a43c3a66595cb8136560ed29be4ff
round  3:23f3f0f6441563d4c202cee0e5cb2335
added ->:3746cbbb418bb73c2964a536cb8e83b1
Key [ 5]:18b26300b86b70acdd1c8f5cbc7c5da8
Key [ 6]:04efc75309b98cd8f1cef5513c18e41e
round  4:c61afa90d3c14bdf588320e857afdc00
Key [ 7]:517c789cecadc455751af73198749fb8
Key [ 8]:fd9711f913b5c844900fa79dd765d0e2
```

```
round  5:a8a0e02ceb556af8bfa321789801183a
Key [ 9]:bb5cf30e7d3ceb930651b1d16ee92750
Key [10]:3d97c7862ecab42720e984972f8efd28
round  6:0b58e922438d224db34b68fca9a5ea12
Key [11]:4ce730344f6b09e449dcdb64cd466666
Key [12]:38828c3a56f922186adcd9b713cdcc31
round  7:b90664c4ac29a8b4bb26debec9ffc5f2
Key [13]:d30fd865ea3e9edcff86a33a2c319649
Key [14]:1fdb63e54413acd968195ab6fa424e83
round  8:6934de3067817cefd811abc5736c163b
Key [15]:a16b7c655bbaa262c807cba8ae166971
Key [16]:7903dd68630105266049e23ca607cda7
Key [17]:888446f2d95e6c2d2803e6f4e815ddc9
Ka      :1674f9dc2063cc2b83d3ef8ba692ebef
----------------------------------------
rand    :1313f7115a9db842fcedc4b10088b48d
PIN length = 1 octets
PIN     :6d
address :008aa9be62d5
round  1:1313f7115a9db842fcedc4b10088b48a
Key [ 1]:6d008aa9be62d56d008aa9be62d56d00
Key [ 2]:46ebfeafb6657b0a1984a8dc0893accf
round  2:839b23b83b5701ab095bafd162ec0ac7
Key [ 3]:8e15595edcf058af62498ee3c1dc6098
Key [ 4]:dd409c3444e94b9cc08396ae967542a0
round  3:c0a2010cc44f2139427f093f4f97ae68
added ->:d3b5f81d9eecd97bbe6ccd8e4f1f62e2
Key [ 5]:487deff5d519f6a6481e947b926f633c
Key [ 6]:5b4b6e3477ed5c2c01f6e607d3418963
round  4:1a5517a0efad3575931d8ea3bee8bd07
Key [ 7]:34b980088d2b5fd6b6a2aceeda99c9c4
Key [ 8]:e7d06d06078acc4ecdbc8da800b73078
round  5:d3ce1fdfe716d72c1075ff37a8a2093f
Key [ 9]:7d375bad245c3b757380021af8ecd408
Key [10]:14dac4bc2f4dc4929a6cceec47f4c3a3
round  6:47e90cb55be6e8dd0f583623c2f2257b
Key [11]:66cfda3c63e464b05e2e7e25f8743ad7
Key [12]:77cfccda1ad380b9fdf1df10846b50e7
round  7:f866ae6624f7abd4a4f5bd24b04b6d43
Key [13]:3e11dd84c031a470a8b66ec6214e44cf
Key [14]:2f03549bdb3c511eea70b65ddbb08253
round  8:02e8e17cf8be4837c9c40706b613dfa8
Key [15]:e2f331229ddfcc6e7bea08b01ab7e70c
Key [16]:b6b0c3738c5365bc77331b98b3fba2ab
Key [17]:f5b3973b636119e577c5c15c87bcfd19
Ka      :38ec0258134ec3f08461ae5c328968a1

========================================
```

9.5 FOUR TESTS OF E3

```
rand    :00000000000000000000000000000000
aco     :48afcdd4bd40fef76693b113
key     :00000000000000000000000000000000
round  1:00000000000000000000000000000000
Key [ 1]:00000000000000000000000000000000
Key [ 2]:4697b1baa3b7100ac537b3c95a28ac64
round  2:78d19f9307d2476a523ec7a8a026042a
Key [ 3]:ecabaac66795580df89af66e66dc053d
Key [ 4]:8ac3d8896ae9364943bfebd4969b68a0
round  3:600265247668dda0e81c07bbb30ed503
Key [ 5]:5d57921fd5715cbb22c1be7bbc996394
Key [ 6]:2a61b8343219fdfb1740e6511d41448f
round  4:d7552ef7cc9dbde568d80c2215bc4277
Key [ 7]:dd0480dee731d67f01a2f739da6f23ca
Key [ 8]:3ad01cd1303e12a1cd0fe0a8af82592c
round  5:fb06bef32b52ab8f2a4f2b6ef7f6d0cd
Key [ 9]:7dadb2efc287ce75061302904f2e7233
Key [10]:c08dcfa981e2c4272f6c7a9f52e11538
round  6:b46b711ebb3cf69e847a75f0ab884bdd
Key [11]:fc2042c708e409555e8c147660ffdfd7
Key [12]:fa0b21001af9a6b9e89e624cd99150d2
round  7:c585f308ff19404294f06b292e978994
Key [13]:18b40784ea5ba4c80ecb48694b4e9c35
Key [14]:454d54e5253c0c4a8b3fcca7db6baef4
round  8:2665fadb13acf952bf74b4ab12264b9f
Key [15]:2df37c6d9db52674f29353b0f011ed83
Key [16]:b60316733b1e8e70bd861b477e2456f1
Key [17]:884697b1baa3b7100ac537b3c95a28ac
round  1:5d3ecb17f26083df0b7f2b9b29aef87c
Key [ 1]:e9e5dfc1b3a79583e9e5dfc1b3a79583
Key [ 2]:7595bf57e0632c59f435c16697d4c864
round  2:de6fe85c5827233fe22514a16f321bd8
Key [ 3]:e31b96afcc75d286ef0ae257cbbc05b7
Key [ 4]:0d2a27b471bc0108c6263aff9d9b3b6b
round  3:7cd335b50d09d139ea6702623af85edb
added ->:211100a2ff6954e6e1e62df913a656a7
Key [ 5]:98d1eb5773cf59d75d3b17b3bc37c191
Key [ 6]:fd2b79282408ddd4ea0aa7511133336f
round  4:991dccb3201b5b1c4ceff65a3711e1e9
Key [ 7]:331227756638a41d57b0f7e071ee2a98
Key [ 8]:aa0dd8cc68b406533d0f1d64aabacf20
round  5:18768c7964818805fe4c6ecae8a38599
Key [ 9]:669291b0752e63f806fce76f10e119c8
Key [10]:ef8bdd46be8ee0277e9b78adef1ec154
round  6:82f9aa127a72632af43d1a17e7bd3a09
Key [11]:f3902eb06dc409cfd78384624964bf51
Key [12]:7d72702b21f97984a721c99b0498239d
round  7:1543d7870bf2d6d6efab3cbf62dca97d
Key [13]:532e60bceaf902c52a06c2c283ecfa32
Key [14]:181715e5192efb2a64129668cf5d9dd4
```

```
round  8:eee3e8744a5f8896de95831ed837ffd5
Key [15]:83017c1434342d4290e961578790f451
Key [16]:2603532f365604646ff65803795ccce5
Key [17]:882f7c907b565ea58dae1c928a0dcf41
kc      :cc802aecc7312285912e90af6a1e1154
----------------------------------------
rand    :950e604e655ea3800fe3eb4a28918087
aco     :68f4f472b5586ac5850f5f74
key     :34e86915d20c485090a6977931f96df5
round  1:950e604e655ea3800fe3eb4a28918087
Key [ 1]:34e86915d20c485090a6977931f96df5
Key [ 2]:8de2595003f9928efaf37e5229935bdb
round  2:d46f5a04c967f55840f83d1cdb5f9afc
Key [ 3]:46f05ec979a97cb6ddf842ecc159c04a
Key [ 4]:b468f0190a0a83783521deae8178d071
round  3:e16ededede9cb6297f32e1203e442ac73a
Key [ 5]:8a171624dedbd552356094daaadcf12a
Key [ 6]:3085e07c85e4b99313f6e0c837b5f819
round  4:805144e55e1ece96683d23366fc7d24b
Key [ 7]:fe45c27845169a66b679b2097d147715
Key [ 8]:44e2f0c35f64514e8bec66c5dc24b3ad
round  5:edbaf77af070bd22e9304398471042f1
Key [ 9]:0d534968f3803b6af447eaf964007e7b
Key [10]:f5499a32504d739ed0b3c547e84157ba
round  6:0dab1a4c846aef0b65b1498812a73b50
Key [11]:e17e8e456361c46298e6592a6311f3fb
Key [12]:ec6d14da05d60e8abac807646931711f
round  7:1e7793cac7f55a8ab48bd33bc9c649e0
Key [13]:2b53dde3d89e325e5ff808ed505706ae
Key [14]:41034e5c3fb0c0d4f445f0cf23be79b0
round  8:3723768baa78b6a23ade095d995404da
Key [15]:e2ca373d405a7abf22b494f28a6fd247
Key [16]:74e09c9068c0e8f1c6902d1b70537c30
Key [17]:767a7f1acf75c3585a55dd4a428b2119
round  1:39809afb773efd1b7510cd4cb7c49f34
Key [ 1]:1d0d48d485abddd3798b483a82a0f878
Key [ 2]:aed957e600a5aed5217984dd5fef6fd8
round  2:6436ddbabe92655c87a7d0c12ae5e5f6
Key [ 3]:fee00bb0de89b6ef0a289696a4faa884
Key [ 4]:33ce2f4411db4dd9b7c42cc586b8a2ba
round  3:cec690f7e0aa5f063062301e049a5cc5
added ->:f7462a0c97e85c1d4572fd52b35efbf1
Key [ 5]:b5116f5c6c29e05e4acb4d02a46a3318
Key [ 6]:ff4fa1f0f73d1a3c67bc2298abc768f9
round  4:dcdfe942e9f0163fc24a4718844b417d
Key [ 7]:5453650c0819e001e48331ad0e9076e0
Key [ 8]:b4ff8dda778e26c0dce08349b81c09a1
round  5:265a16b2f766afae396e7a98c189fda9
Key [ 9]:f638fa294427c6ed94300fd823b31d10
Key [10]:1ccfa0bd86a9879b17d4bc457e3e03d6
round  6:628576b5291d53d1eb8611c8624e863e
Key [11]:0eaee2ef4602ac9ca19e49d74a76d335
```

```
Key [12]:6e1062f10a16e0d378476da3943842e9
round  7:d7b9c2e9b2d5ea5c27019324cae882b3
Key [13]:40be960bd22c744c5b23024688e554b9
Key [14]:95c9902cb3c230b44d14ba909730d211
round  8:97fb6065498385e47eb3df6e2ca439dd
Key [15]:10d4b6e1d1d6798aa00aa2951e32d58d
Key [16]:c5d4b91444b83ee578004ab8876ba605
Key [17]:1663a4f98e2862eddd3ec2fb03dcc8a4
kc      :c1beafea6e747e304cf0bd7734b0a9e2
----------------------------------------
rand    :6a8ebcf5e6e471505be68d5eb8a3200c
aco     :658d791a9554b77c0b2f7b9f
key     :35cf77b333c294671d426fa79993a133
round  1:6a8ebcf5e6e471505be68d5eb8a3200c
Key [ 1]:35cf77b333c294671d426fa79993a133
Key [ 2]:c4524e53b95b4bf2d7b2f095f63545fd
round  2:ade94ec585db0d27e17474b58192c87a
Key [ 3]:c99776768c6e9f9dd3835c52cea8d18a
Key [ 4]:f1295db23823ba2792f21217fc01d23f
round  3:da8dc1a10241ef9e6e069267cd2c6825
Key [ 5]:9083db95a6955235bbfad8aeefec5f0b
Key [ 6]:8bab6bc253d0d0c7e0107feab728ff68
round  4:e6665ca0772ceecbc21222ff7be074f8
Key [ 7]:2fa1f4e7a4cf3ccd876ec30d194cf196
Key [ 8]:267364be247184d5337586a19df8bf84
round  5:a857a9326c9ae908f53fee511c5f4242
Key [ 9]:9aef21965b1a6fa83948d107026134c7
Key [10]:d2080c751def5dc0d8ea353cebf7b973
round  6:6678748a1b5f21ac05cf1b117a7c342f
Key [11]:d709a8ab70b0d5a2516900421024b81e
Key [12]:493e4843805f1058d605c8d1025f8a56
round  7:766c66fe9c460bb2ae39ec01e435f725
Key [13]:b1ed21b71daea03f49fe74b2c11fc02b
Key [14]:0e1ded7ebf23c72324a0165a698c65c7
round  8:396e0ff7b2b9b7a3b35c9810882c7596
Key [15]:b3bf4841dc92f440fde5f024f9ce8be9
Key [16]:1c69bc6c2994f4c84f72be8f6b188963
Key [17]:bb7b66286dd679a471e2792270f3bb4d
round  1:45654f2f26549675287200f07cb10ec9
Key [ 1]:1e2a5672e66529e4f427b0682a3a34b6
Key [ 2]:974944f1ce0037b1febcf61a2bc961a2
round  2:990cd869c534e76ed4f4af7b3bfbc6c8
Key [ 3]:8147631fb1ce95d624b480fc7389f6c4
Key [ 4]:6e90a2db33d284aa13135f3c032aa4f4
round  3:ceb662f875aa6b94e8192b5989abf975
added ->:8b1bb1d753fe01e1c08b2ba9f55c07bc
Key [ 5]:cbad246d24e36741c46401e6387a05f9
Key [ 6]:dcf52aaec5713110345a41342c566fc8
round  4:d4e000be5de78c0f56ff218f3c1df61b
Key [ 7]:8197537aa9d27e67d17c16b182c8ec65
Key [ 8]:d66e00e73d835927a307a3ed79d035d8
round  5:9a4603bdef954cfaade2052604bed4e4
```

```
Key [ 9]:71d46257ecc1022bcd312ce6c114d75c
Key [10]:f91212fa528379651fbd2c32890c5e5f
round  6:09a0fd197ab81eb933eece2fe0132dbb
Key [11]:283acc551591fadce821b02fb9491814
Key [12]:ca5f95688788e20d94822f162b5a3920
round  7:494f455a2e7a5db861ece816d4e363e4
Key [13]:ba574aef663c462d35399efb999d0e40
Key [14]:6267afc834513783fef1601955fe0628
round  8:37a819f91c8380fb7880e640e99ca947
Key [15]:fdcd9be5450eef0f8737e6838cd38e2b
Key [16]:8cfbd9b8056c6a1ce222b92b94319b38
Key [17]:4f64c1072c891c39eeb95e63318462e0
kc      :a3032b4df1cceba8adc1a04427224299
----------------------------------------
rand    :5ecd6d75db322c75b6afbd799cb18668
aco     :63f701c7013238bbf88714ee
key     :b9f90c53206792b1826838b435b87d4d
round  1:5ecd6d75db322c75b6afbd799cb18668
Key [ 1]:b9f90c53206792b1826838b435b87d4d
Key [ 2]:15f74bbbde4b9d1e08f858721f131669
round  2:72abb85fc80c15ec2b00d72873ef9ad4
Key [ 3]:ef7fb29f0b01f82706c7439cc52f2dab
Key [ 4]:3003a6aecdee06b9ac295cce30dcdb93
round  3:2f10bab93a0f73742183c68f712dfa24
Key [ 5]:5fcdbb3afdf7df06754c954fc6340254
Key [ 6]:ddaa90756635579573fe8ca1f93d4a38
round  4:183b145312fd99d5ad08e7ca4a52f04e
Key [ 7]:27ca8a7fc703aa61f6d7791fc19f704a
Key [ 8]:702029d8c6e42950762317e730ec5d18
round  5:cbad52d3a026b2e38b9ae6fefffecc32
Key [ 9]:ff15eaa3f73f4bc2a6ccfb9ca24ed9c5
Key [10]:034e745246cd2e2cfc3bda39531ca9c5
round  6:ce5f159d0a1acaacd9fb4643272033a7
Key [11]:0a4d8ff5673731c3dc8fe87e39a34b77
Key [12]:637592fab43a19ac0044a21afef455a2
round  7:8a49424a10c0bea5aba52dbbffcbcce8
Key [13]:6b3fde58f4f6438843cdbe92667622b8
Key [14]:a10bfa35013812f39bf2157f1c9fca4e
round  8:f5e12da0e93e26a5850251697ec0b917
Key [15]:2228fe5384e573f48fdd19ba91f1bf57
Key [16]:5f174db2bc88925c0fbc6b5485bafc08
Key [17]:28ff90bd0dc31ea2bb479feb7d8fe029
round  1:0c75eed2b54c1cfb9ff522daef94ed4d
Key [ 1]:a21ceb92d3c027326b4de775865fe8d0
Key [ 2]:26f64558a9f0a1652f765efd546f3208
round  2:48d537ac209a6aa07b70000016c602e8
Key [ 3]:e64f9ef630213260f1f79745a0102ae5
Key [ 4]:af6a59d7cebfd0182dcca9a537c4add8
round  3:8b6d517ac893743a401b3fb7911b64e1
added ->:87e23fa87ddf90c1df10616d7eaf51ac
Key [ 5]:9a6304428b45da128ab64c8805c32452
Key [ 6]:8af4d1e9d80cb73ec6b44e9b6e4f39d8
```

Appendix IV - Sample Data

```
round  4:9f0512260a2f7a5067efc35bf1706831
Key [ 7]:79cc2d138606f0fca4e549c34a1e6d19
Key [ 8]:803dc5cdde0efdbee7a1342b2cd4d344
round  5:0cfd7856edfafac51f29e86365de6f57
Key [ 9]:e8fa996448e6b6459ab51e7be101325a
Key [10]:2acc7add7b294acb444cd933f0e74ec9
round  6:2f1fa34bf352dc77c0983a01e8b7d622
Key [11]:f57de39e42182efd6586b86a90c86bb1
Key [12]:e418dfd1bb22ebf1bfc309cd27f5266c
round  7:ee4f7a53849bf73a747065d35f3752b1
Key [13]:80a9959133856586370854db6e0470b3
Key [14]:f4c1bc2f764a0193749f5fc09011a1ae
round  8:8fec6f7249760ebf69e370e9a4b80a92
Key [15]:d036cef70d6470c3f52f1b5d25b0c29d
Key [16]:d0956af6b8700888a1cc88f07ad226dc
Key [17]:1ce8b39c4c7677373c30849a3ee08794
kc      :ea520cfc546b00eb7c3a6cea3ecb39ed

========================================
```

Appendix V

BLUETOOTH AUDIO

This recommendation outlines some general guidelines for voice transmission over the Bluetooth air interface. The recommendation is formed such that a smooth audio interface of a Bluetooth terminal to other audio, electronic devices and cellular terminal equipment can be carried out.

CONTENTS

1 General Audio Recommendations....1008

1.1 Maximum Sound Pressure....1008

1.2 Other Telephony Network Requirements....1008

1.3 Audio Levels for Bluetooth....1008

1.4 Microphone Path....1009

1.4.1 SLR measurement model....1009

1.5 Loudspeaker Path....1009

1.5.1 RLR measurement model....1009

1.6 Bluetooth Voice Interface....1009

1.7 Frequency Mask....1011

1 GENERAL AUDIO RECOMMENDATIONS

1.1 MAXIMUM SOUND PRESSURE

It is the sole responsibility of each manufacturer to design their audio products in a safe way with regards to injury to the human ear. Bluetooth doesn't specify maximum sound pressure from an audio device.

1.2 OTHER TELEPHONY NETWORK REQUIREMENTS

It is the sole responsibility of each manufacturer to design the Bluetooth audio product so that it meets the regulatory requirements of all telephony networks that it may be connected to.

1.3 AUDIO LEVELS FOR BLUETOOTH

Audio levels shall be calculated as Send Loudness Rating, SLR, and Receive Loudness Rating, RLR. The calculation methods are specified in ITU-T Recommendation P.79.

The physical test set-up for Handsets and Headsets is described in ITU-T Recommendation P.51 andP.57

The physical test set-up for speakerphones and "Vehicle handsfree systems" is specified in ITU-T Recommendation P.34.

A general equation for computation of loudness rating (LR) for telephone sets is given by ITU-T recommendations P.79 and is given by

$$LR = -\frac{10}{m}\log_{10}\left(\sum_{i=N_1}^{N_2} 10^{m(s_i - w_i)/10}\right), \qquad \text{(EQ 1)}$$

where

m is a constant (~ 0.2).

w_i = weighting coefficient (different for the various LRs).

S_i = the sensitivity at frequency F_i, of the electro-acoustic path

N_1,N_2, consecutive filter bank numbers (Art. Index: 200-4000 Hz)

(EQ 1) is used for calculating the (SLR) according to Figure 1.1:, and (RLR) according to Figure 1.2:. When calculating LRs one must only include those parts of the frequency band where an actual signal transmission can occur in order to ensure that the additive property of LRs is retained. Therefore ITU-T P.79 uses only the frequency band 200-4000 Hz in LR computations.

1.4 MICROPHONE PATH

1.4.1 SLR measurement model

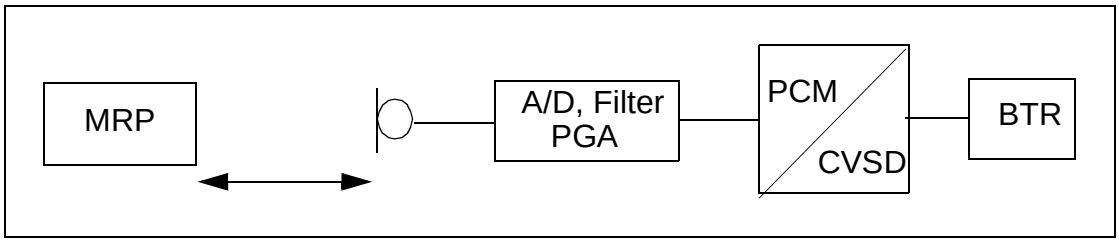


Figure 1.1: SLR measurement set-up.

1.5 LOUDSPEAKER PATH

1.5.1 RLR measurement model

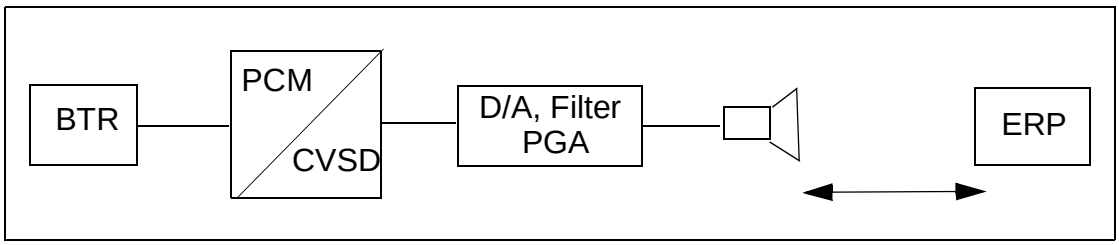


Figure 1.2: RLR measurement set-up.

1.6 BLUETOOTH VOICE INTERFACE

The specification for the Bluetooth voice interface should follow in the first place the *ITU-T Recommendations P.79*, which specifies the loudness ratings for telephone sets. These recommendations give general guidelines and specific algorithms used for calculating the loudness ratings of the audio signal with respect to Ear Reference Point (ERP).

For Bluetooth voice interface to the different cellular system terminals, loudness and frequency recommendations based on the cellular standards should be used. For example, GSM 03.50 gives recommendation for both the loudness ratings and frequency mask for a GSM terminal interconnection with Bluetooth.

1- The output of the CVSD decoder are 16-bit linear PCM digital samples, at a sampling frequency of 8 ksample/second. Bluetooth also supports 8-bit log PCM samples of A-law and μ-law type. The sound pressure at the ear reference point for a given 16-bit CVSD sample, should follow the sound pressure level given in the cellular standard specification.

2- A maximum sound pressure which can be represented by a 16-bit linear PCM sample at the output of the CVSD decoder should be specified according

to the loudness rating, in ITU P.79 and at PGA value of 0 dB. Programmable Gain Amplifiers (PGAs) are used to control the audio level at the terminals by the user. For conversion between various PCM representations: A-law, μ-law and linear PCM, ITU-T G.711, G.712, G.714 give guidelines and PCM value relationships. Zero-code suppression based on ITU-T G.711 is also recommended to avoid network mismatches.

1.7 FREQUENCY MASK

For interfacing a Bluetooth terminal to a digital cellular mobile terminal, a compliance of the CVSD decoder signal to the frequency mask given in the cellular standard, is recommended to guarantee correct function of the speech coders. A recommendation for a frequency mask is given in Table 1.1. Figure 1.3: shows a plot of the frequency mask for Bluetooth (solid line). The GSM frequency mask (dotted line) is shown in Figure 1.3: for comparison.

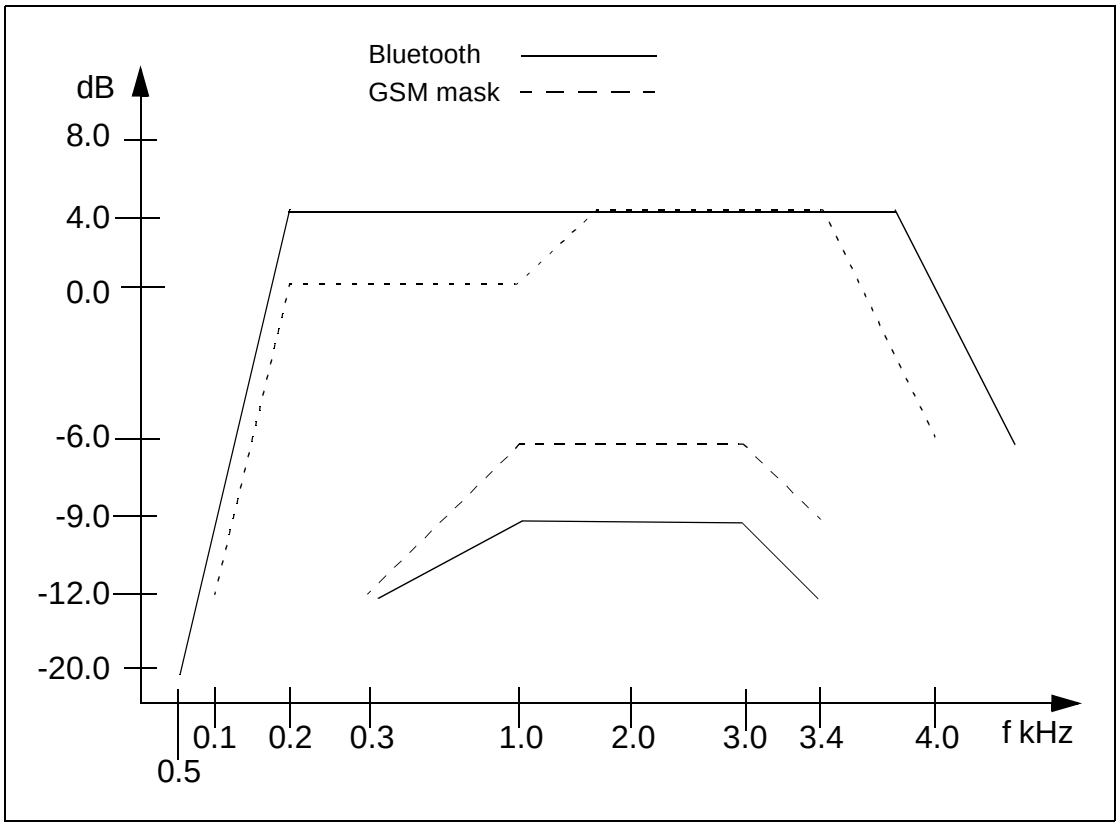


Figure 1.3: Plot of recommended frequency mask for Bluetooth. The GSM send frequency mask is given for comparison (dotted line)

Frequency (Hz)	Upper Limit (dB)	Lower Limit (dB)
50	-20	-
300	4	-12
1000	4	-9
2000	4	-9
3000	4	--9
3400	4	-12
4000	0	-

Table 1.1: Recommended Frequency Mask for Bluetooth

Appendix VI

BASEBAND TIMERS

This appendix contains a list of all timers defined the Baseband Specification.

CONTENTS

1 Baseband Timers 1016

1.1 LIST OF TIMERS 1016

1.1.1 inquiryTO 1016

1.1.2 pageTO 1016

1.1.3 pagerespTO 1016

1.1.4 inqrespTO 1016

1.1.5 newconnectionTO 1017

1.1.6 supervisionTO 1017

1 BASEBAND TIMERS

This appendix contains a list of all timers defined in this specification. Definitions and default values of the timers are listed below.

All timer values are given in slots.

1.1 LIST OF TIMERS

1.1.1 inquiryTO

The *inquiryTO* defines the number of slots the **inquiry** substate will last. Its value is determined by an HCI command.

1.1.2 pageTO

The *pageTO* defines the number of slots the **page** substate can last before a response is received. Its value is determined by an HCI command.

1.1.3 pagerespTO

In the slave, it defines the number of slots the slave awaits the master's response, FHS packet, after sending the page acknowledgment ID packet. In the master, *pagerespTO* defines the number of slots the master should wait for the FHS packet acknowledgment before returning to **page** substate. Both master and slave units should use the same value for this timeout, to ensure common page/scan intervals after reaching *pagerespTO*.

The *pagerespTO*value is 8 slots.

1.1.4 inqrespTO

In the inquiry scan substate, when a device triggers on an inquiry, it waits a RAND random number of slots and returns to inquiry scan. The *inqRespTO* defines the number of slots the device will stay in the inquiry scan substate without triggering on an inquiry after the RAND wait period. The timeout value should preferably be in multiples of an inquiry train period. Upon reaching the *inqrespTO*, the device returns to **CONNECTION** or **STANDBY** state.

The *inqrespTO* value is 128 slots.

1.1.5 newconnectionTO

Every time a new connection is started through paging, scanning, master-slave switch or unparking, the master sends a POLL packet as the first packet in the new connection. Transmission and acknowledgment of this POLL packet is used to confirm the new connection. If the POLL packet is not received by the slave or the response packet is not received by the master for *newconnectionTO* number of slots, both the master and the slave will return to the previous substate.

newconnectionTO value is 32 slots.

1.1.6 supervisionTO

The *supervisionTO* is used by both the master and slave to monitor link loss. If a device does not receive any packets that pass the HEC check and have the proper AM_ADDR for a period of *supervisionTO*, it will reset the link *supervisionTO* will work through hold and sniff periods.

The *supervisionTO* value is determined by an HCI command. At the baseband level a default value that is equivalent to 20 seconds will be used.

Appendix VII

OPTIONAL PAGING SCHEMES

CONTENTS

1 General 1023

2 Optional Paging Scheme I 1024

2.1 Page 1024

2.2 Page Scan 1026

2.3 Page Response Procedures 1026

2.4 Train Tracing 1027

1 GENERAL

For the access procedure, several paging schemes may be used. There is one mandatory paging scheme which has to be supported by all Bluetooth devices. This scheme has been described in Baseband Specification Section 10.6 on page 96 In addition to the mandatory scheme, a Bluetooth unit may support one or more optional paging schemes. The method used for page scan is indicated in the FHS payload, see Baseband Specification Section 4.4.1.4 on page 56. Three additional optional paging schemes are possible; only optional paging scheme ***I*** has been defined yet.

2 OPTIONAL PAGING SCHEME I

In this section the first optional paging scheme is described which may be used according to the rules specified in Baseband Specification Section 10 on page 92 and LMP Specification Section 3.23 on page 230. The paging code for optional scheme ***I*** is 1 (0 is used for the mandatory scheme), see also Baseband Specification Section 4.4.1.4 on page 56

The main difference between the first optional paging scheme and the mandatory scheme is the construction of the page train sent by the pager. In addition to transmission in the even master slots, the master is transmitting in the odd master slots as well. This allows the slave unit to reduce the scan window.

2.1 PAGE

The same 32 frequencies that are used for transmitting ID-packets in the mandatory paging scheme are used in the optional paging scheme ***I*** (for the construction of page trains, see Baseband Specification Section 11.3.2 on page 134. The 32 frequencies are also split into an **A-train** and **B train**. In contrast to the mandatory scheme, the same 32 frequencies that are used for transmitting are also used for reception trials, to catch the response from the addressed device.

The construction of the page train in optional page scheme ***I*** differs from the page train in the mandatory scheme in two ways:

- the page train consists of 10 slots, or 6.25 ms
- the first 8 slots of the train are used to transmit the ID packets, the 9th slot is used to send a marker packet, and the 10th slot is used for the return of a slave response

The marker packets precede the return slot, indicating the position where the slave can respond, and with which frequency. For the marker codes M_ID, bit-inverted page access codes are used. If a marker code is received at T_m with frequency f_k, a return is expected at nominally T_m+625µs at frequency f_k.

Note: The bit-inverted code M_ID to be used as marker code is beneficial for the implementation of the correlators, because the sign of the correlation peak can be used to identify the mark code during page scanning. Still, the transmitting party is uniquely identified, since inverted ID packets are not identical to the ID packets for the device with bit-wise inverted LAP.

The frequency ordering in the train and the frequencies used for the marker and receive slots change after every train. After 8 trains, all of which have a different appearance, the entire procedure is repeated. It is, therefore, more appropriate to talk about subtrains, each with length 6.25ms. Eight subtrains form a supertrain, which is repeated. An example of a supertrain with the eight subtrains is

illustrated in Figure 2.1. The supertrain length is 50ms. In this example, the **A-train** is assumed with an estimated frequency of f_8; as a consequence, the frequencies selected for the train range from f_0 to f_{15}. The marker codes M_ID are indicated as **M**; the receive (half) slots are indicated as **R**.

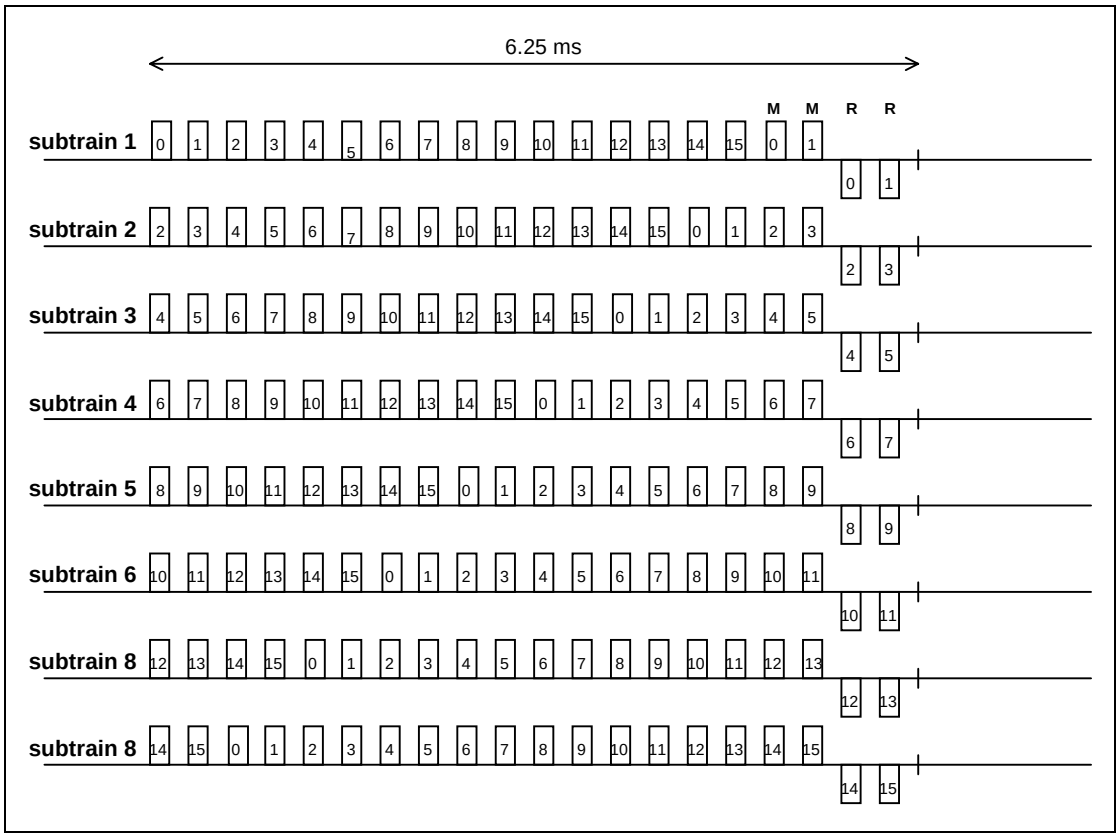


Figure 2.1: Example of train configuration for optional page scheme I.

Corresponding to the paging modes R0, R1 and R2 of the mandatory scheme, the optional scheme supports the same three modes as described for the mandatory scheme in Baseband Specification Section 10.6.2 on page 96

Since the subtrain length is now 10 slots, the 1.28s interval does not cover a multiple of (sub)trains any longer. Therefore, in contrast to the mandatory scheme, the exchange from **A-train** to **B-train** and vice versa is not based on the 1.28s interval, but instead on a multiple number of supertrains. For the R1 and R2 modes, the repetition of a supertrain N_{sup} is indicated in Table 2.1 below.

mode	No SCO link	One SCO link (HV3)	Two SCO links (HV3)
R1	N_{sup}=26	N_{sup}=52	N_{sup}=77
R2	N_{sup}=52	N_{sup}=103	N_{sup}=154

*Table 2.1: Relation between repetition duration of **A**- and **B**-trains and paging modes R1 and R2 when SCO links are present*

In accordance with the phase input to the hop selection scheme *Xp* in (EQ 4) on page 135 in the Baseband Specification (Section 11.3.2), the phase input *Xp_opt* in the optional mode is determined by:

$$Xp\_opt=[k_{offset\_opt} + \mathrm{ST}(cnt)] \bmod 32 \qquad \text{(EQ A1)}$$

where k_{offset_opt} is determined by the A/B selection and the clock estimation of the recipient:

$$k_{offset\_opt} = \begin{cases} \mathrm{CLKE}_{16\text{-}12} + 24 & \text{A-train} \\ \mathrm{CLKE}_{16\text{-}12} + 8 & \text{B-train} \end{cases} \qquad \text{(EQ A2)}$$

and ST is a function determining the structure of the sub- and supertrain:

$$\mathrm{ST}(cnt) = (cnt \bmod 160 - 2*\mathrm{INT}[((cnt \bmod 160) + 18)/20]) \bmod 16 \qquad \text{(EQ A3)}$$

k_{offset_opt} is determined once at the beginning of the repetition period. The CLKE value as is found at the beginning of the repetition interval is taken (the repetition interval being the interval in which the same supertrain is repeated all the time). As long as no train change takes place, k_{offset_opt} is not updated. *cnt* is a counter which is reset to zero at the beginning of the repetition interval and is incremented at the half-slot rate (3200 cycles/s)

The first two ID-packets of a train are transmitted in an even numbered slot.

2.2 PAGE SCAN

The basic page scanning is identical to the mandatory scheme except that a scan duration of 9.5 · 0.625 = 5.9375 ms is sufficient at the slave side.

If a device wants to scan concurrently for the mandatory and optional mode (e.g. after an inquiry response was sent), the device shall try to identify whether the paging party uses the optional scheme after an ID packet was caught. This can be done by train tracing; i.e. the device can determine whether transmission takes place in consecutive slots (optional paging scheme ***I***) or in every over slot (mandatory paging scheme), and/or whether mark codes are sent.

2.3 PAGE RESPONSE PROCEDURES

The page response procedures at the master and slave sides are almost identical to the procedures described in the mandatory mode (see Baseband Specification Section 10.6.4 on page 100). There are two differences:

- The page response routine starts after the transmission and reception of the marker code M_ID
- The ID packet sent by recipient is identical to the frequency in which the marker code was received

For the page response timing, see Figure 2.2 and Figure 2.3.

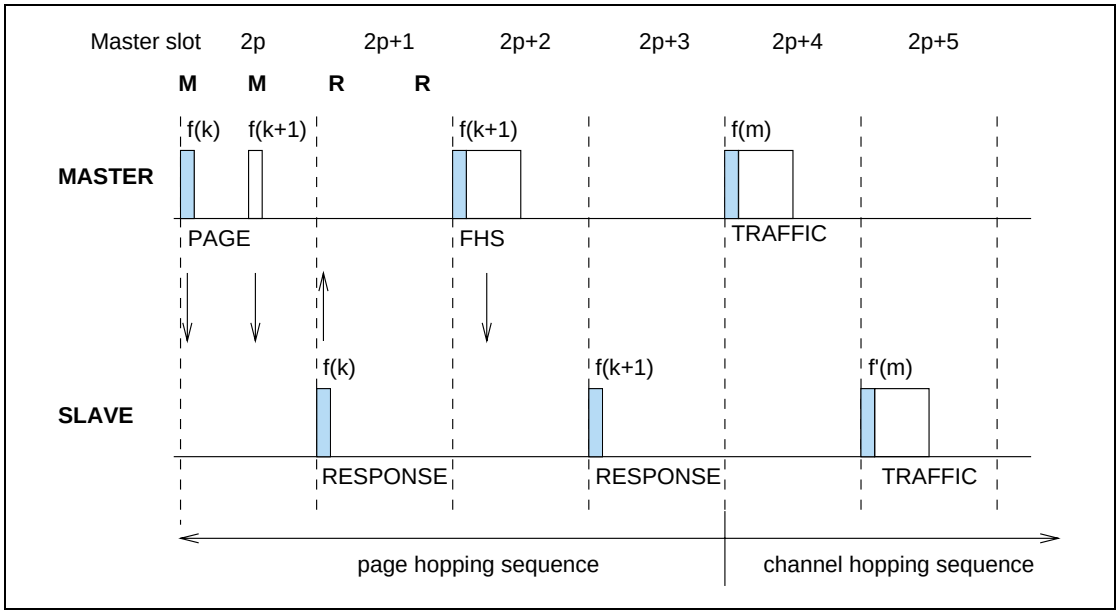


Figure 2.2: Messaging when marker code is received in first half slot of even master slot

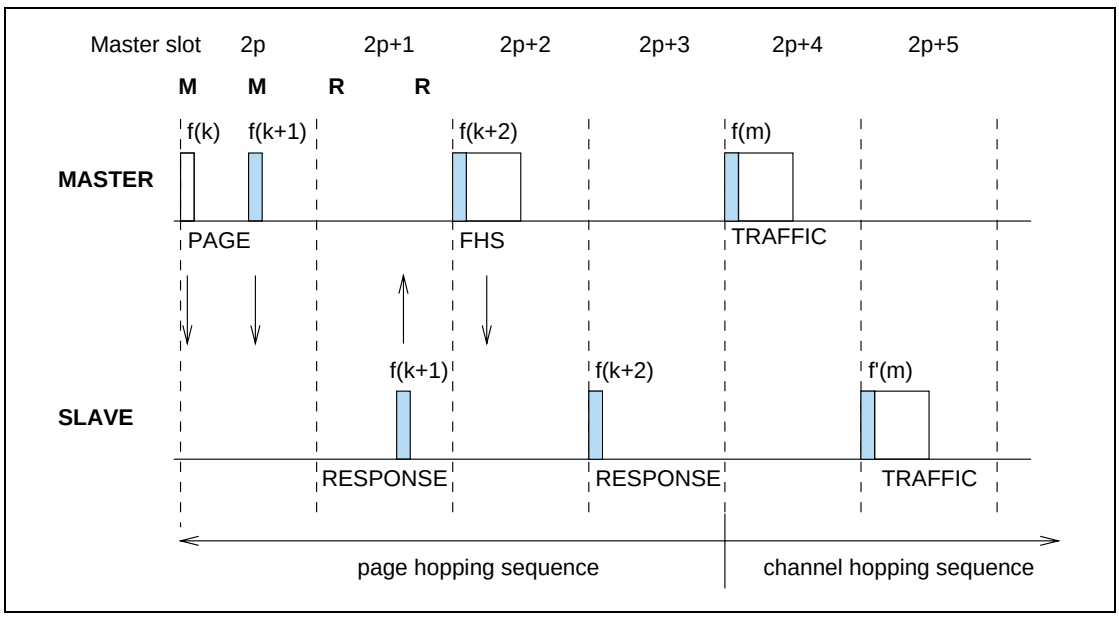


Figure 2.3: Messaging when marker code is received in second half slot of even master slot

2.4 TRAIN TRACING

This section outlines how a slave may search for the mark code although the current partitioning into A- and B-trains at the master side is not known. Train tracing means that the slave tries to receive as many page access codes from the train as possible, to catch a mark code as soon as possible. When searching for the mark codes, or trying to distinguish between the mandatory paging mode and the optional paging mode, a unit shall set up a hopping pattern for train tracing after the reception of the first access code. The hopping pattern shall ensure that the transmission and reception is performed with a 50% probability on the same frequency regardless of the actual frequency set (16 frequencies) used for paging.

Appendix IX

MESSAGE SEQUENCE CHARTS

Between Host and Host Controller/Link Manager

This document shows examples of inter-working between HCI Commands and LM Protocol Data Units in form of message sequence charts. It helps to understand and to correctly use the HCI Commands.

CONTENTS

1 Introduction 1033

2 Services without connection request 1034

2.1 Remote Name Request 1034

2.2 One-Time Inquiry 1035

2.3 Periodic Inquiry 1036

3 ACL connection establishment and detachment 1038

3.1 ACL Connection Request phase 1039

3.2 ACL Connection Setup phase 1041

3.2.1 Pairing 1041

3.2.2 Authentication 1042

3.3 Encryption and Connection Setup Complete 1043

3.4 ACL Disconnection 1045

4 Optional activities after ACL Connection establishment 1046

4.1 Authentication Requested 1046

4.2 Set Connection Encryption 1047

4.3 Change Connection Link Key 1049

4.4 Master Link Key 1049

4.5 Read Remote Supported Features 1051

4.6 Read Clock Offset 1051

4.7 Read Remote Version Information 1052

4.8 QoS Setup 1053

4.9 Switch Role 1054

5 SCO Connection establishment and detachment 1055

5.1 SCO Connection setup 1055

5.1.1 Master activates the SCO Connection setup 1055

5.1.2 Slave activates the SCO Connection setup 1056

5.2 SCO Disconnection 1057

6 Special modes: sniff, hold, park 1058

6.1 Sniff Mode 1058

6.2 Hold Mode 1059

6.3 Park Mode 1061

6.3.1 Enter park mode 1061

6.3.2 Exit Park Mode 1062

7 Buffer management, flow control 1064

8 Loopback Mode 1066

8.1 Local Loopback Mode 1066

8.2 Remote Loopback Mode 1068

9 List of Acronyms and Abbreviations 1069

10 List of Figures 1070

11 List of Tables 1071

12 References 1072

1 INTRODUCTION

The goal of this document is to show the interworkings of HCI-Commands and LM-PDUs. It focuses on the message sequence charts for the procedures specified in [3] “Bluetooth Host Controller Interface Functional Specification” with regard to LM Procedures from [2] “Link Manager Protocol”.

We illustrate here the most useful scenarios, but we do not cover all possible alternatives. Furthermore, the message sequence charts do not consider the transfer error over Air Interface or Host Interface. In all message sequence charts it is assumed that all events are not masked, so the Host Controller will not filter out any events.

Notation used in the message sequence charts:

Box:

- Replaces a group of transactions
- Indicates the start of a procedure or a sub-scenario

Note: in a message sequence chart where several sub-scenarios exist, the sub-scenarios can be executed optionally, consequently, exclusively or independently from each other.

Hexagon:

- Indicates a condition that is needed to start the transaction below this hexagon

Arrow:

- Represents a message, signal or transaction

Comment:

- “/* … */” indicates editor comments

2 SERVICES WITHOUT CONNECTION REQUEST

2.1 REMOTE NAME REQUEST

The service Remote Name Request is used to find out the name of the remote BT Device without an explicit ACL Connection request.

Sending an HCI_Remote_Name_Request (BD_ADDR, Page_Scan_ Repetition_Mode, Page_Scan_Mode, Clock_Offset), the Host expects that its local BT Device will automatically try to connect to the remote BT Device (with the specified BD_ADDR). Then the local BT Device should try to get the name, to disconnect, and finally to return the name of the remote BT Device back to the Host (see Figure 2.1 Remote Name Request: sub-scenario 1).

Note: if an ACL Connection already exists (see Figure 2.1 Remote Name Request: sub-scenario 2), the Remote Name Request procedure will be executed like an optional service. No Paging and no ACL Detachment need to be done.

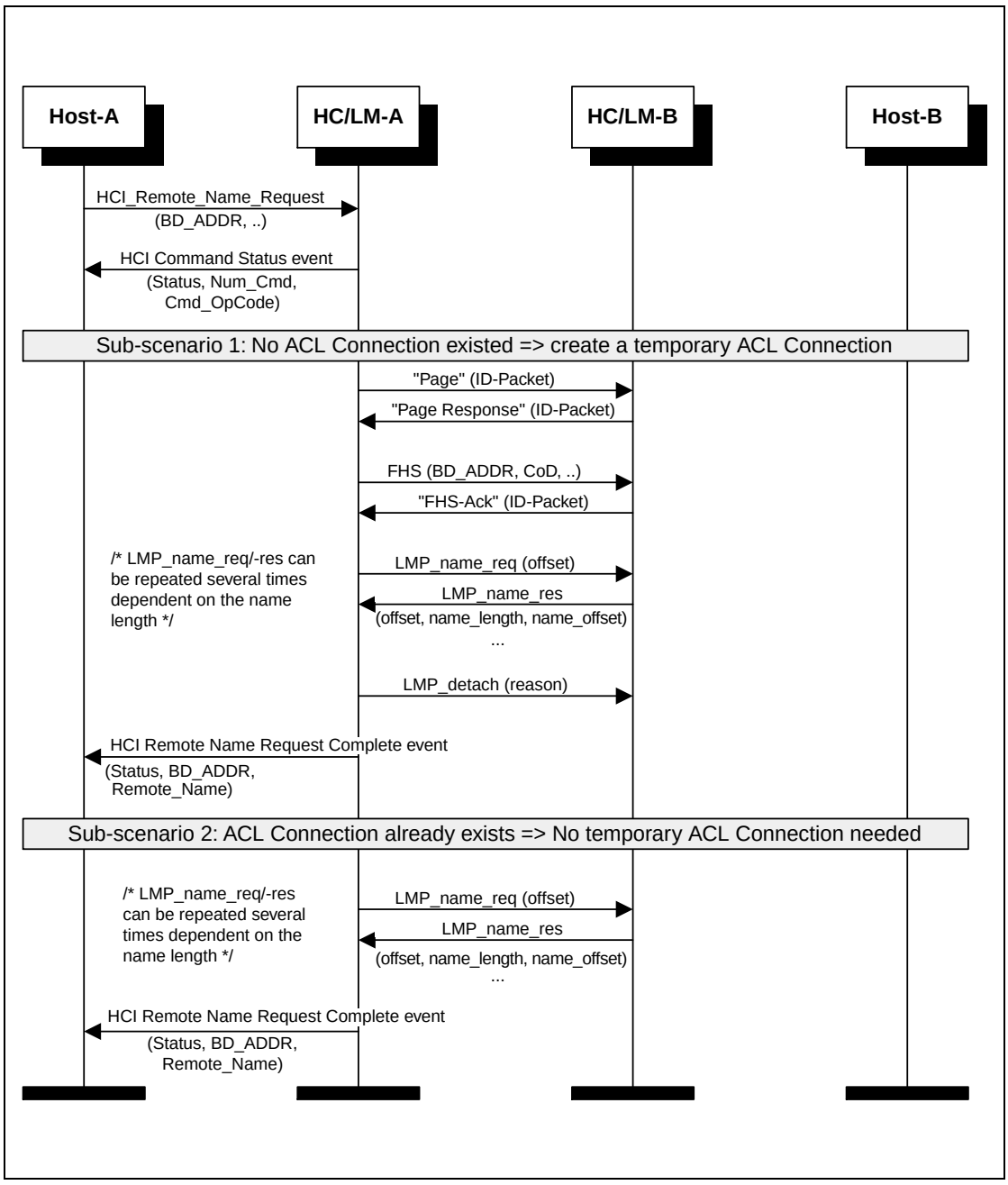


Figure 2.1: Remote Name Request

2.2 ONE-TIME INQUIRY

Inquiry is used to detect and collect nearby BT Devices. When receiving the command HCI_Inquiry (LAP, Inquiry_Length, Num_Responses), HC will start the baseband inquiry procedure with an Inquiry Access Code (derived from the specified LAP) and Inquiry Length. When Inquiry Responses are received, HC will filter out and then return the information related to the found BT Devices using one or several Inquiry Result events (Num_Responses, BD_ADDR[I], Page_Scan_Repetition_Mode[i], Page_Scan_Period_Mode[i], Page_Scan_Mode[i], Class_Of_Device[i], Clock_Offset[i]) to the Host.

The filtering of found BT Devices is specified in HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition) with the Filter_Type = Inquiry Result. When the Inquiry procedure is completed, Inquiry Complete event (Status) must be returned to the Host. Otherwise, the command HCI_Inquiry_Cancel( ) will be used to directly stop the inquiry procedure.

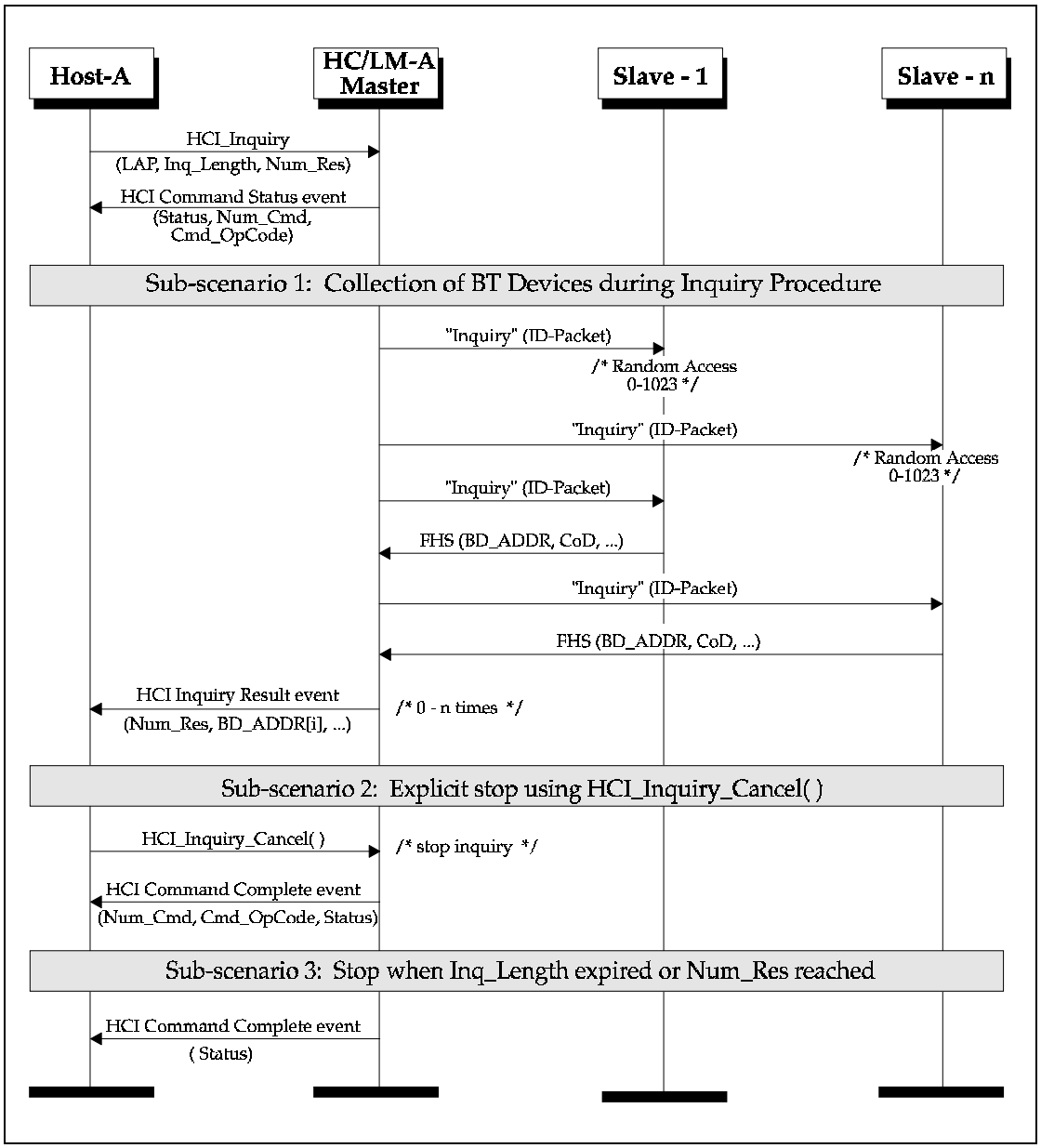


Figure 2.2: One-Time Inquiry

2.3 PERIODIC INQUIRY

Periodic inquiry is needed when the inquiry procedure is to be repeated periodically. Receipt of the command HCI_Periodic_Inquiry_Mode (Max_Period_Length, Min_Period_Length, LAP, Inquiry_Length, Num_Responses) HC will start the periodic Inquiry Mode with the specified parameters Max_Period_Length, Min_Period_Length, Inquiry_Access_code

(derived from LAP) and Inquiry_Length. As in the one-time Inquiry procedure, only BT Devices that are specified in the HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition) with the Filter_Type = Inquiry Result will not be filtered out. Therefore, in the inquiry cycle, one or several Inquiry Result events (Num_Responses, BD_ADDR[i], Page_Scan_Repetition_ Mode[i], Page_Scan_Period_Mode[i], Page_Scan_Mode[i], Class_Of_ Device[i], Clock_Offset[i]) and Inquiry Complete event (Status) will be returned to the Host with one, or a list of, found BT Devices. The periodic Inquiry can be stopped using HCI_Exit_Periodic_ Inquiry_Mode( ).

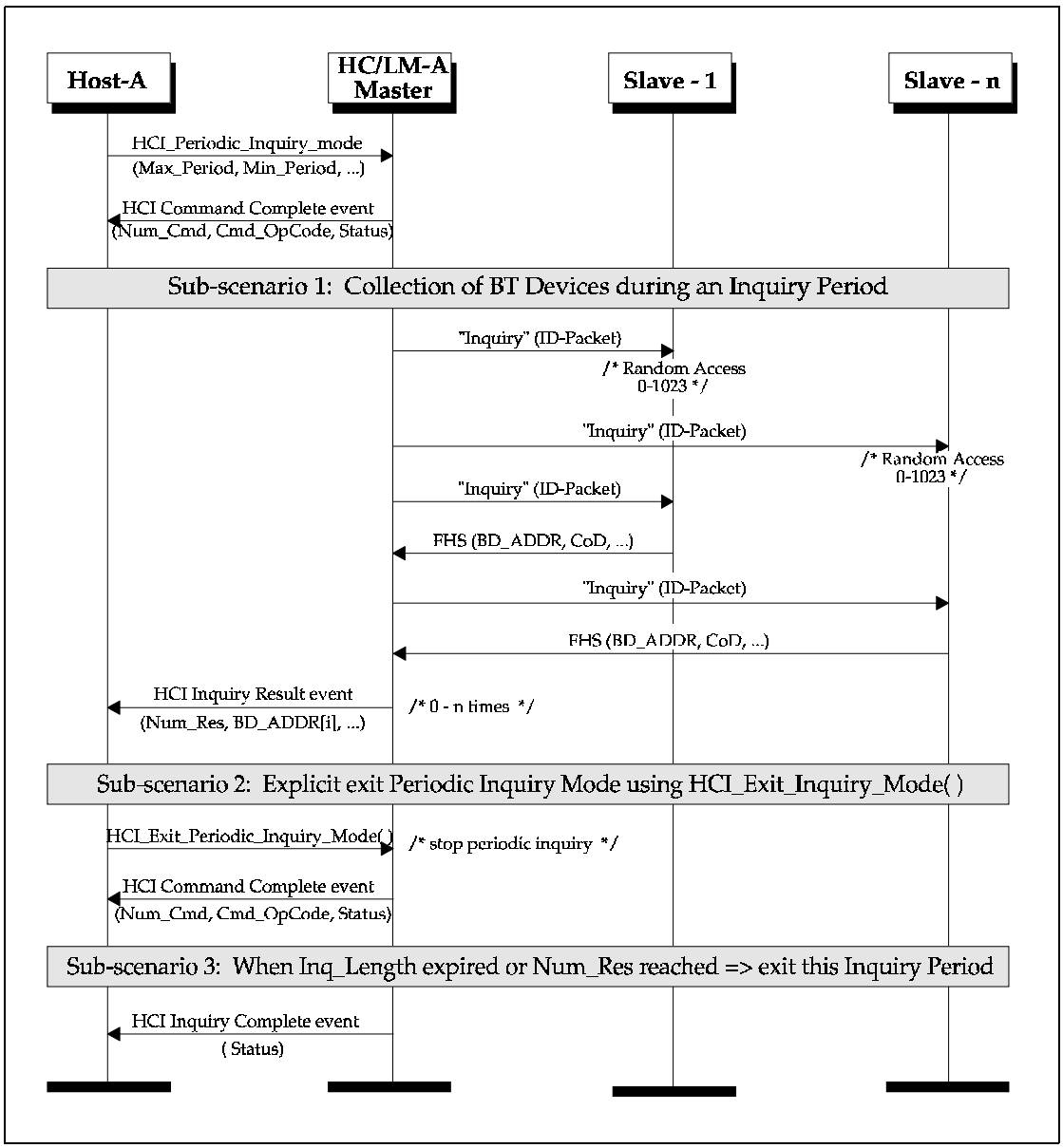


Figure 2.3: Periodic Inquiry

3 ACL CONNECTION ESTABLISHMENT AND DETACHMENT

The overview of the ACL Connection establishment and detachment is shown in Figure 3.1 Overview of ACL Connection establishment and detachment.

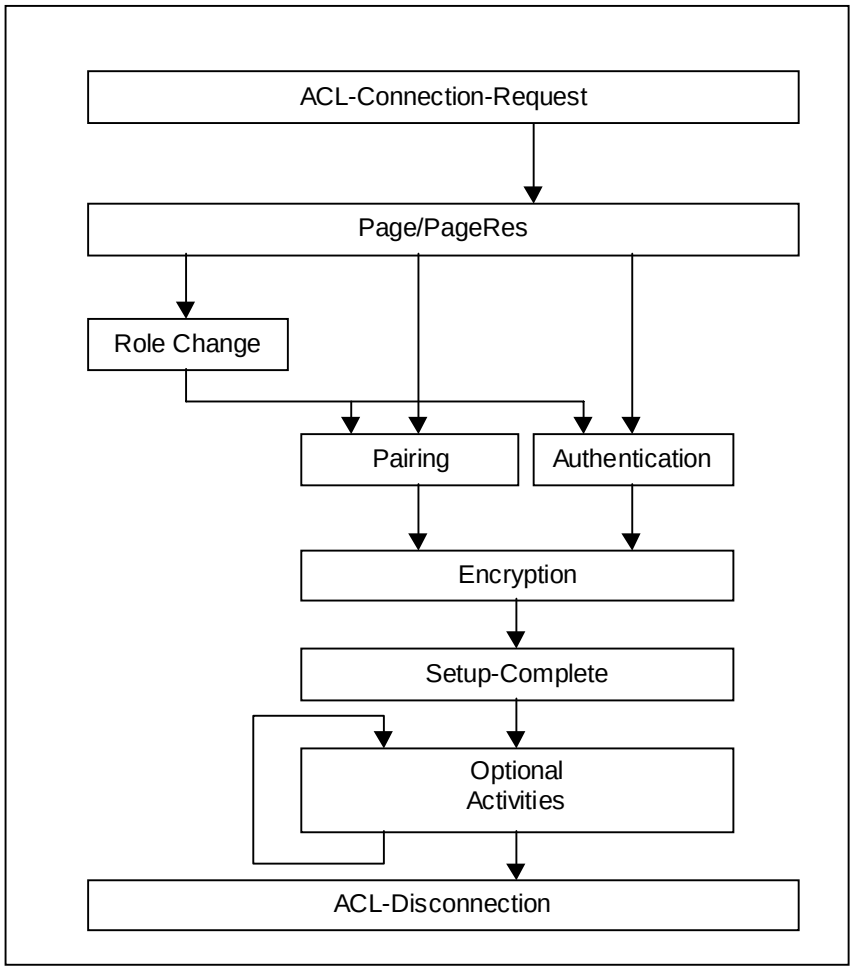


Figure 3.1: Overview of ACL Connection establishment and detachment

3.1 ACL CONNECTION REQUEST PHASE

The ACL Connection Request phase is identified between the HCI_Create_Connection (BD_ADDR, Packet_Type, Page_Scan_Repetition_Mode, Page_Scan_Mode, Clock_Offset, Allow_Role_Switch) from the master side and the response from the slave side with rejection or acceptation on the LM level. Three alternative sub-scenarios are shown in Figure 3.2, “ACL Connection Request phase,” on page1046 .

Sub-scenario 1: Slave rejects ACL Connection Request

If the ACL Connection request is rejected by slave, a Connection Complete event (Status, Connection_Handle, BD_ADDR, Link_Type, Encryption_Mode) will be then returned to Host, whereby the Status will be copied from the Reason parameter of the command HCI_Reject_Connection_ Request (Reason, BD_ADDR). The parameters Connection_Handle and Encryption_Mode will be meaningless.

Sub-scenario 2: Slave accepts ACL Connection Request

When the slave responds with LMP_accepted ( ) correspondent to LMP_host_ connection_req ( ), the ACL Connection Request is accepted. The master will continue with the ACL Connection Setup, where pairing, authentication or encryption will be executed.

Sub-scenario 3: Slave accepts ACL Connection Request with Role Change

This case is identified when the slave sends an LMP_switch_req ( ) to initiate Role Change. If the master accepts, the baseband Master-Slave Switch will be executed. Thereafter, the ACL Connection Setup will continue.

Note: on the slave side, an incoming connection request can be automatically accepted by using HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition) with the Filter_Type = 0x02 /*Connection_Setup*/.

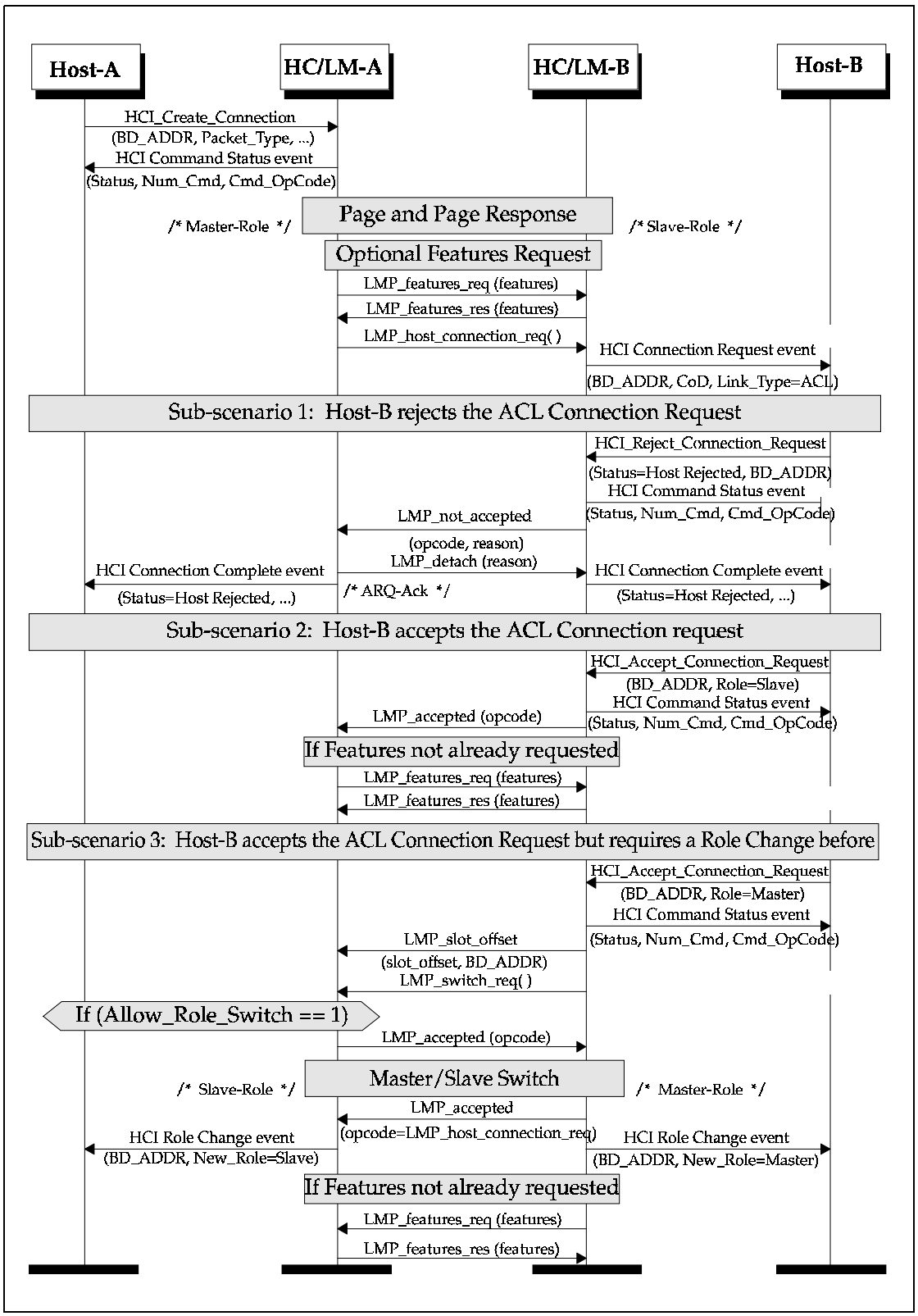


Figure 3.2: ACL Connection Request phase

3.2 ACL CONNECTION SETUP PHASE

If the ACL Connection Request phase was successful, the ACL Connection Setup phase will start, with the goal of executing security procedures like pairing, authentication and encryption. The ACL Connection Setup phase is successfully finished when LMP_setup_complete ( ) is exchanged and the Connection Complete event (Status=0x00, Connection_Handle, BD_ADDR, Link_Type, Encryption_Mode) is sent to the Host.

3.2.1 Pairing

If authentication is required and the BT Devices to be connected don't have a common link key, the pairing procedure on LM Level will be executed using the PIN Input from Host.During the pairing, the link key creation - and mutual authentication procedures will be done. Note: the created Link Key can be stored either in the BT Device or in the Host.

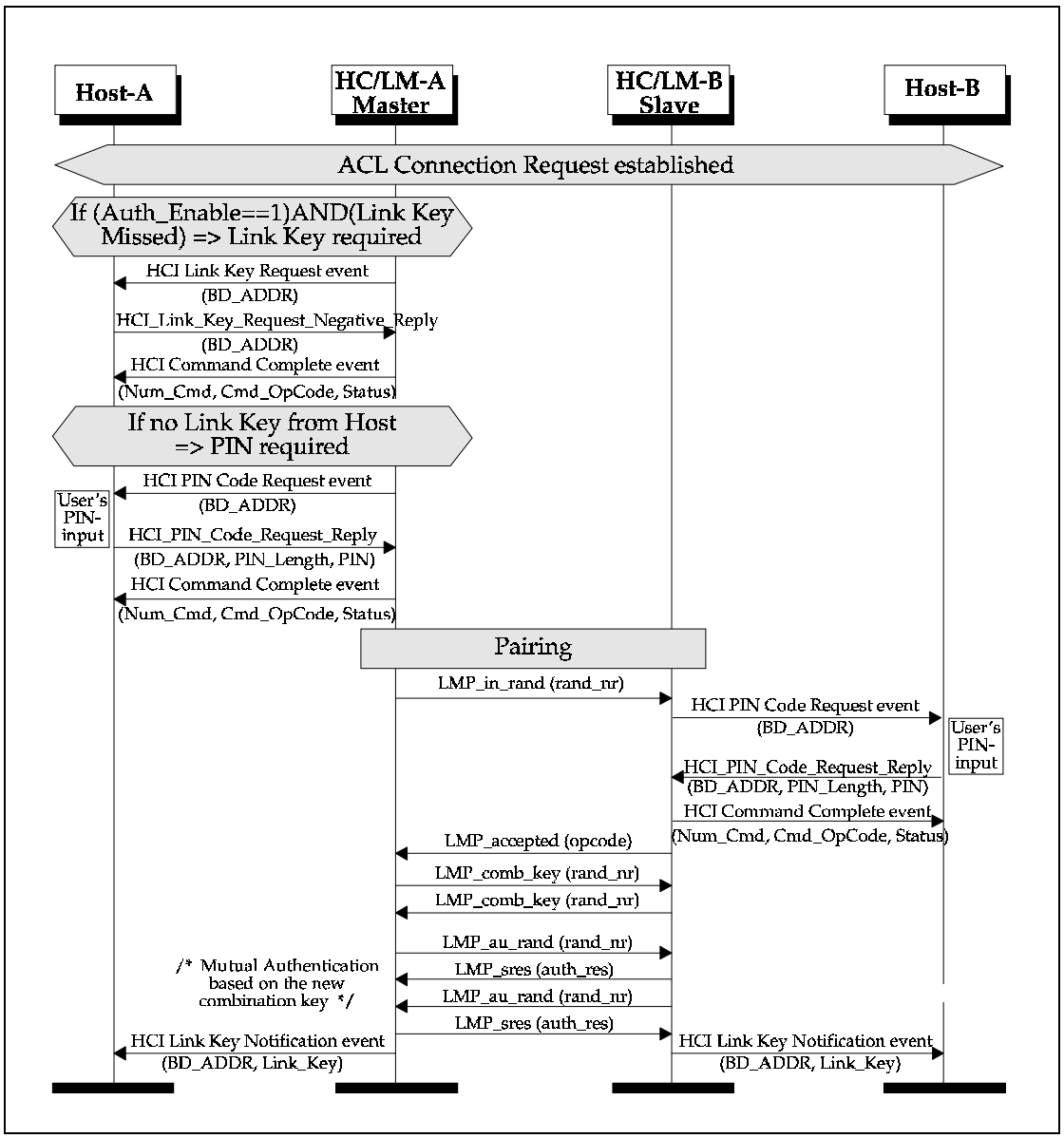


Figure 3.3: ACL Connection setup with pairing

3.2.2 Authentication

If a common link key already exists between the BT Devices, pairing is not needed. Note: a Link Key created during pairing can be stored either in the BT Device or in the Host. If the parameter Authentication_Enable is set, the authentication procedure has to be executed. Here, the MSC only shows the case when Authentication_Enable is set on both sides.

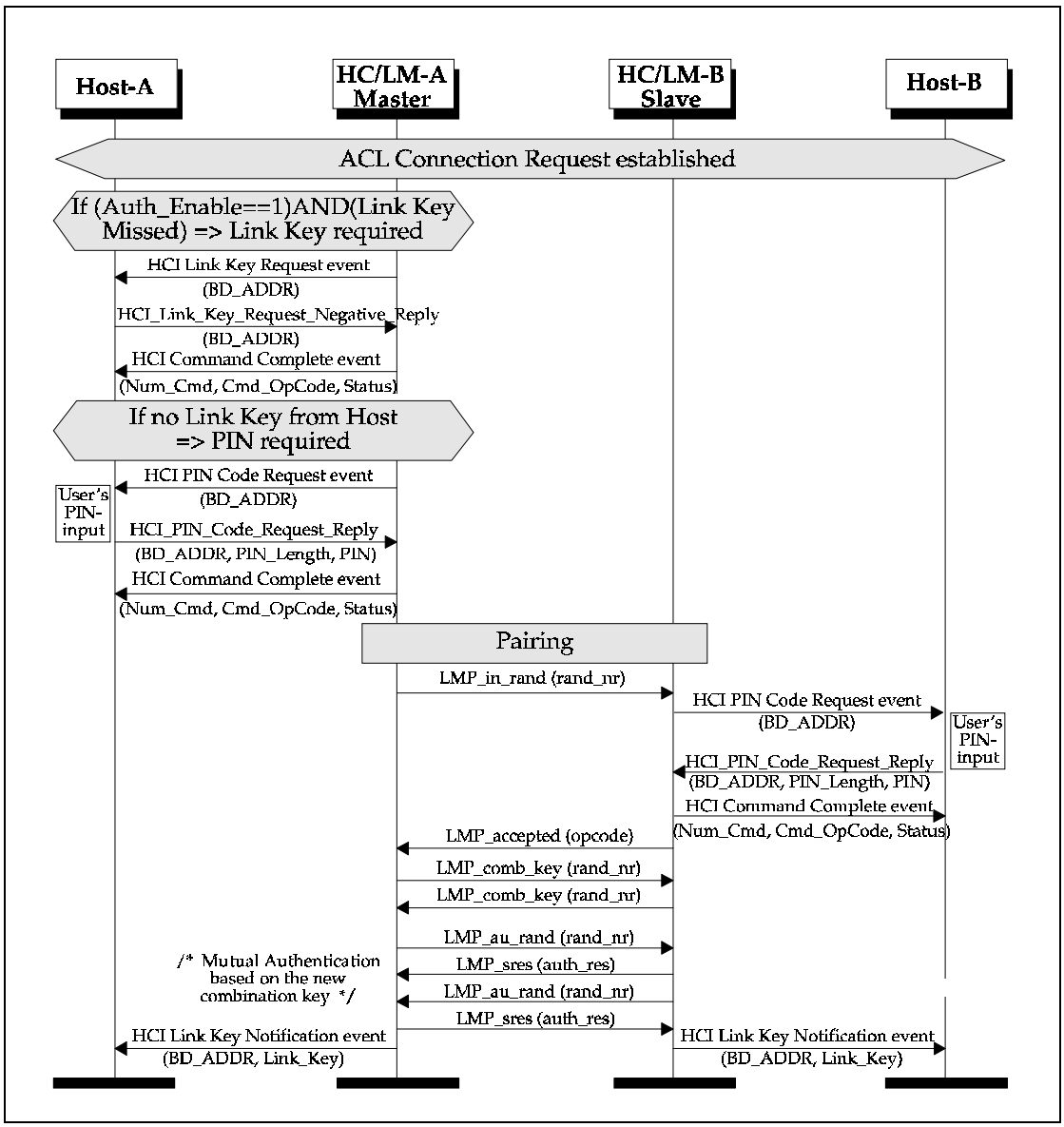


Figure 3.4: ACL Connection setup with authentication

3.3 ENCRYPTION AND CONNECTION SETUP COMPLETE

Once the pairing/authentication procedure is successful, the encryption procedure will be started. Here, the MSC only shows how to set up an encrypted point-to-point connection (Encryption_Mode = 1 /*point-to-point/). Note: an encrypted connection requires an established common link key.

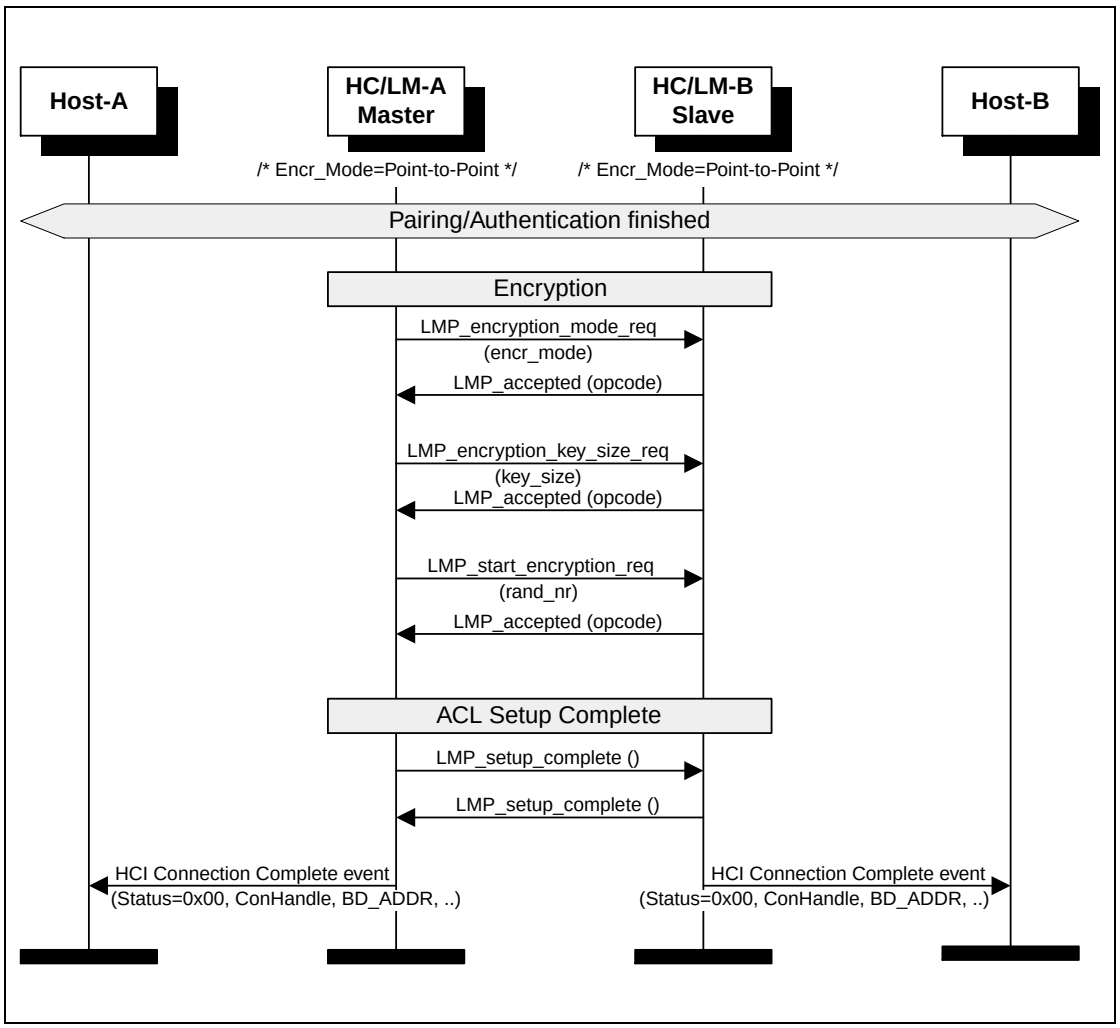


Figure 3.5: Encryption and Setup complete

3.4 ACL DISCONNECTION

At any time, an established ACL Connection can be detached by an HCI_Disconnect (Connection_Handle, Reason). If one or several SCO Connections exist, they must first be detached before the ACL Connection can be released.

Note: the disconnection procedure is one-sided and doesn't need an explicit acknowledgment from the remote LM. So the ARQ Acknowledgment from the LC is needed, to ensure that the remote LM has received the LMP_detach (reason).

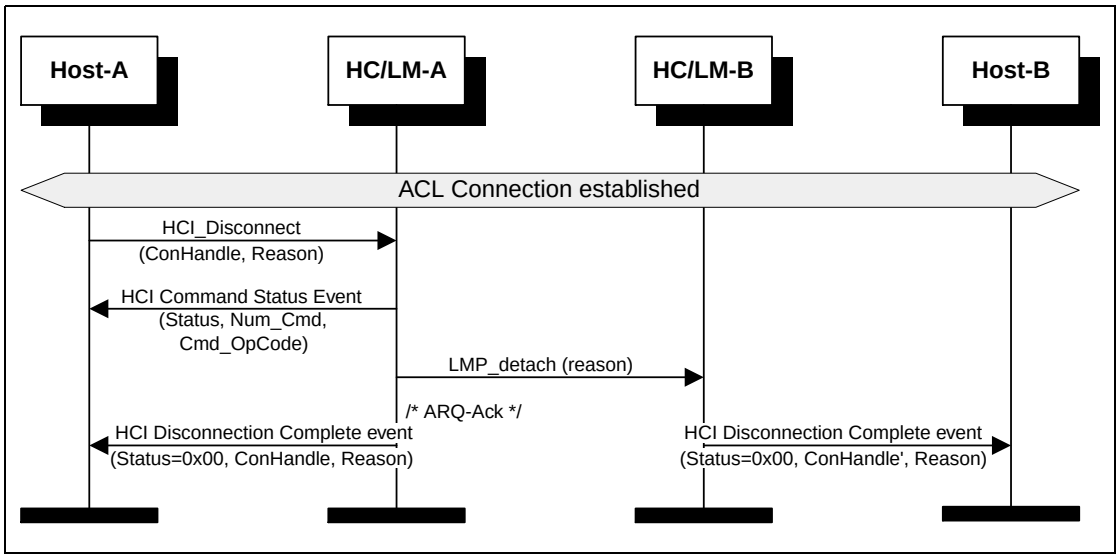


Figure 3.6: ACL Disconnection

4 OPTIONAL ACTIVITIES AFTER ACL CONNECTION ESTABLISHMENT

4.1 AUTHENTICATION REQUESTED

Authentication can be explicitly executed at any time after an ACL Connection has been established. If the Link Key was missed in HC/LM, the Link Key will be required from the Host, as in the authentication procedure (see 3.2.2).

Note: if the HC/LM and the Host don't have the Link Key a PIN Code Request event will be sent to the Host to request a PIN Code for pairing. A procedure identical to ACL Connection Setup with Pairing (see 3.2.1) will be used.

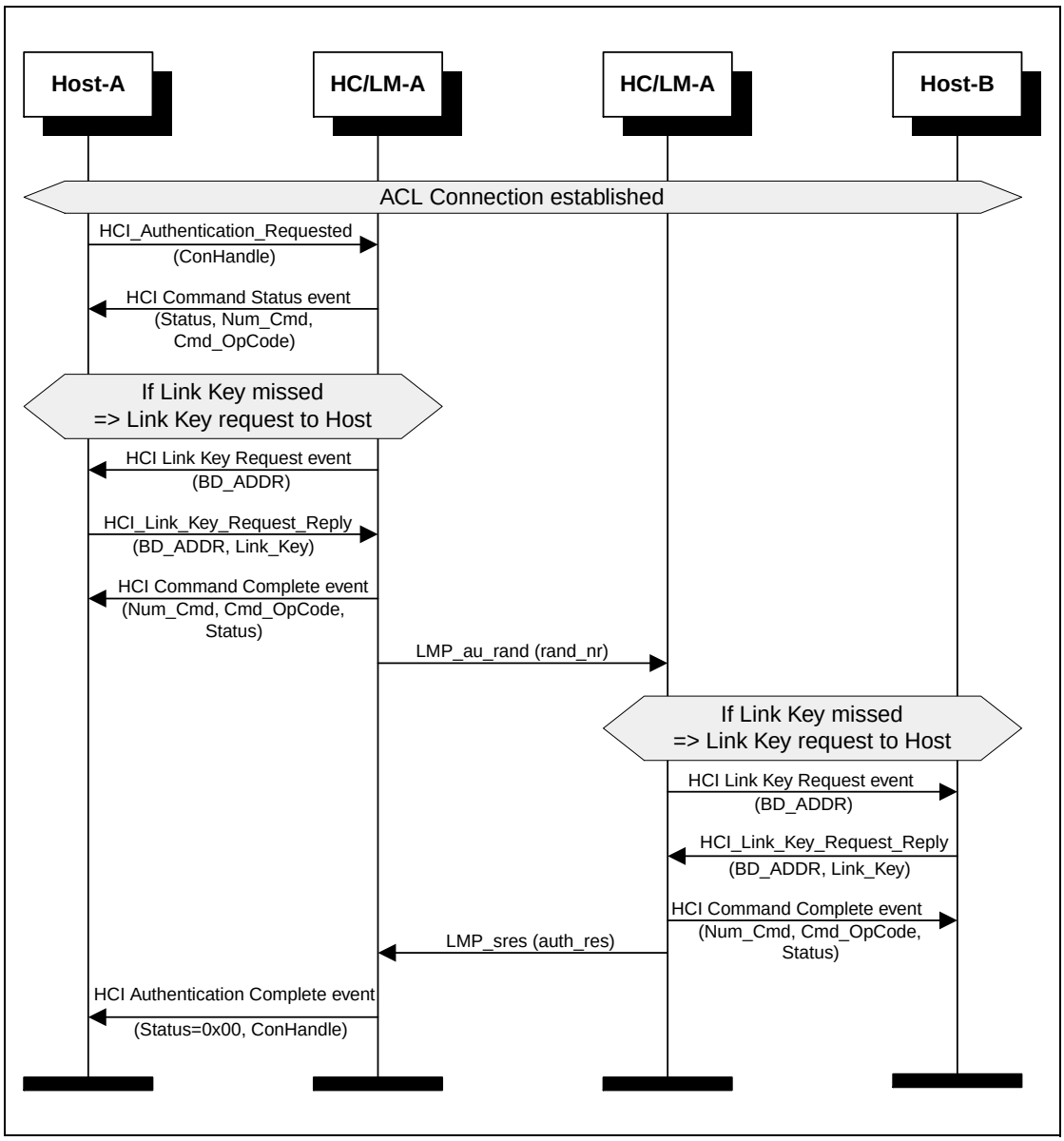


Figure 4.1: Authentication Requested

4.2 SET CONNECTION ENCRYPTION

Using the command HCI_Set_Connection_Encryption (Connection_Handle, Encryption_Enable), the Host is able to switch the encryption of a connection with the specified Connection_Handle to ON/OFF. This command can be applied on both the master- and slave sides (only the master side is shown in Figure 4.2 Set Connection Encryption). If this command occurs on the slave side, the only difference is that LMP_encryption_mode_req (encryption_mode) will be sent from the HC/LM Slave. LMP_encryption_key_size_req (key_size) and LMP_start_encryption_req (rand_nr) will still be requested from the HC/LM master.

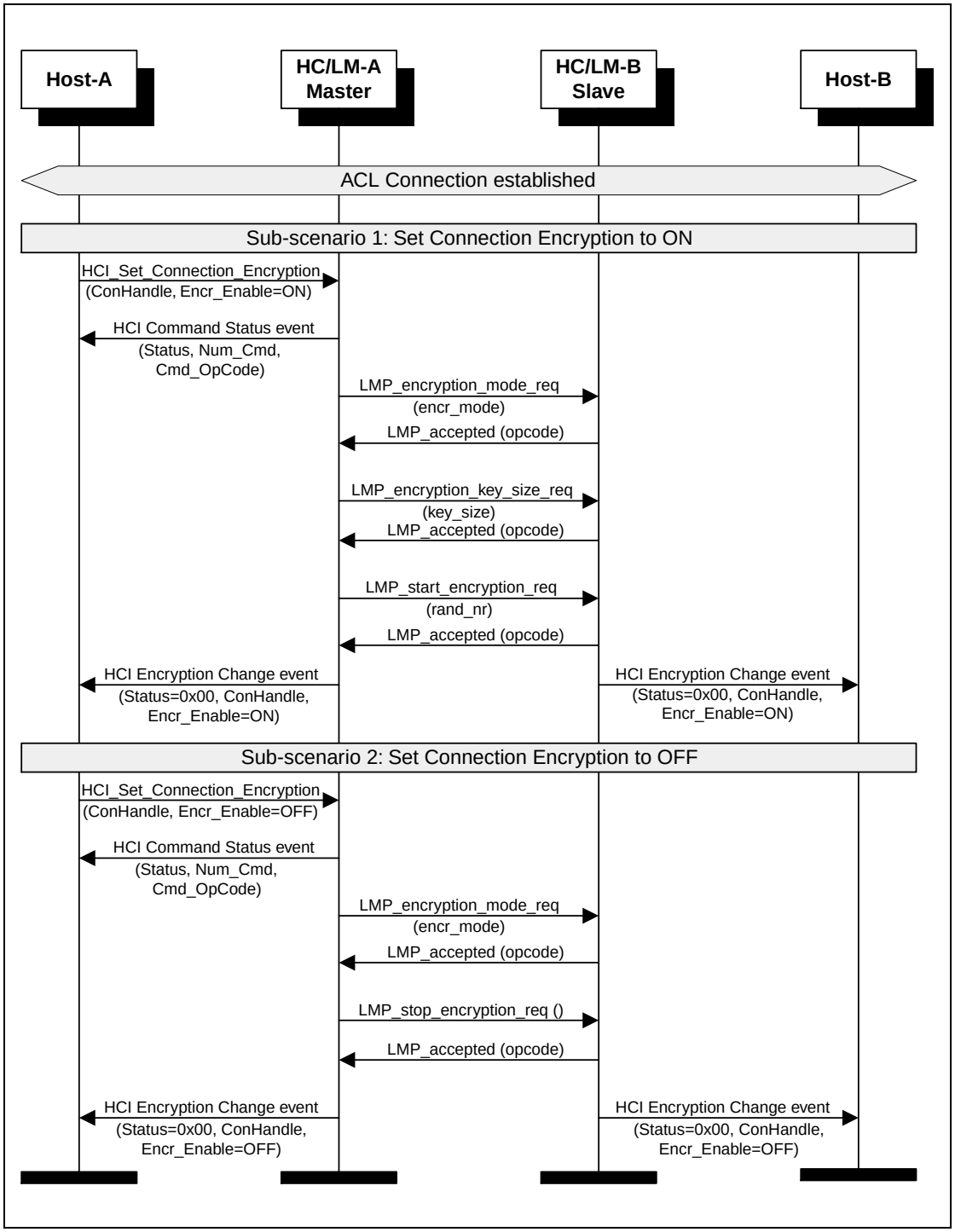


Figure 4.2: Set Connection Encryption

4.3 CHANGE CONNECTION LINK KEY

Using the command HCI_Change_Connection_Link_Key (Connection_Handle), the Host can explicitly change the common link key shared between the BT Devices.

Note: if the connection encryption was enabled and the temporary link key was used, it is the task of the BT Master to automatically restart the encryption (first stop and then restart) after the link key is successfully changed.

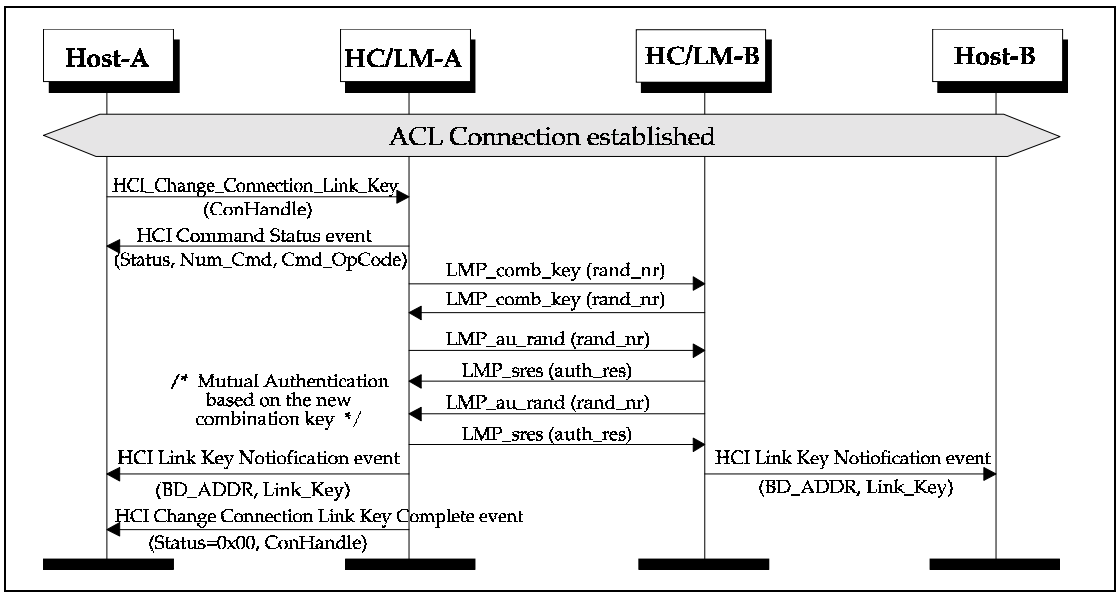


Figure 4.3: Change Connection Link Key

4.4 MASTER LINK KEY

The Figure 4.4 Master Link Key shows how the Host can explicitly switch between the temporary Link Key and the semi-permanent Link Key. Since this command can only be used for the BT Master, the Link Key switch will affect all connections.

Note: if encryption was enabled, it is the task of the BT Master to restart the encryption separately for each slave.

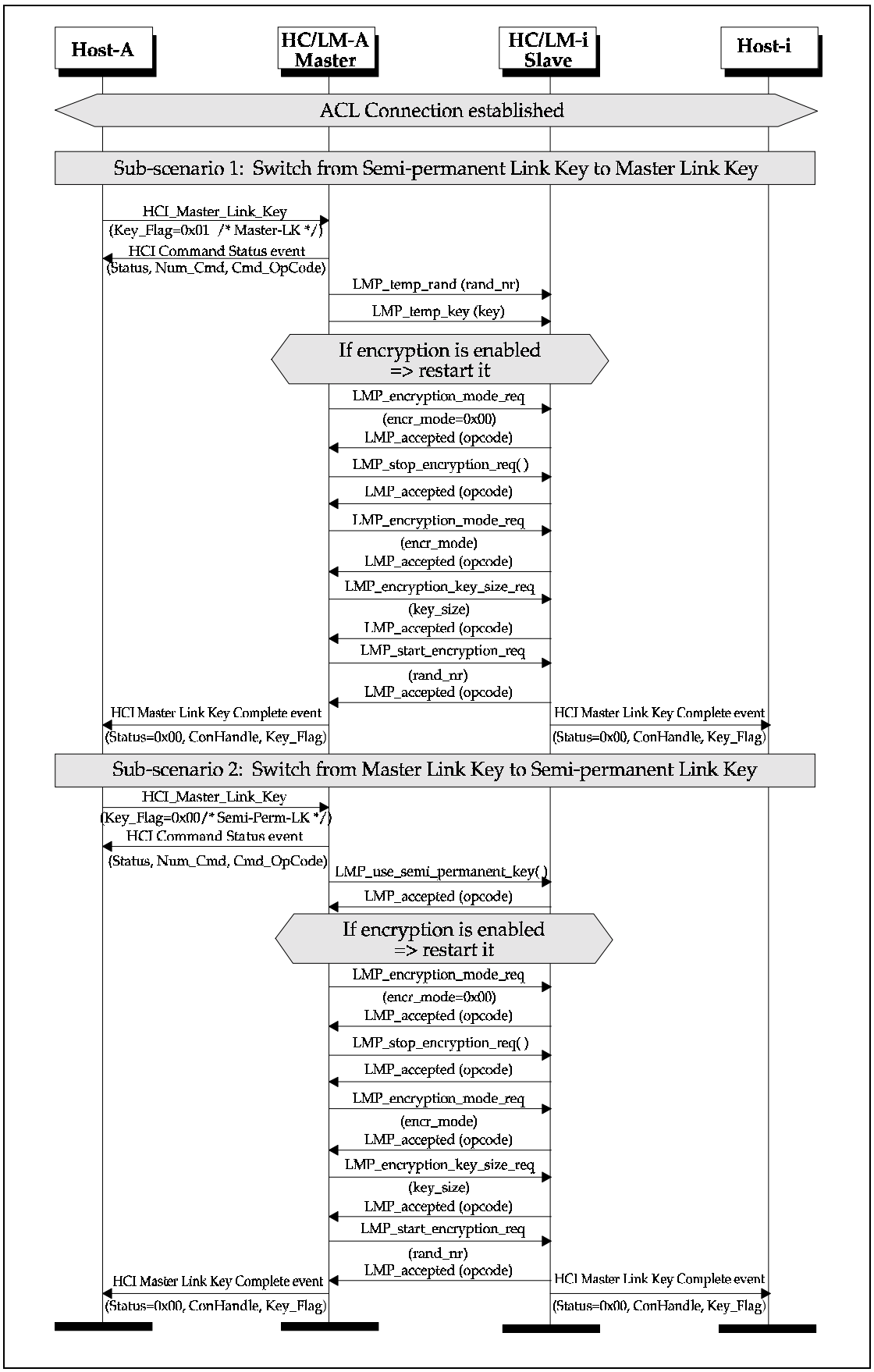


Figure 4.4: Master Link Key

4.5 READ REMOTE SUPPORTED FEATURES

Using the command HCI_Read_Remote_Supported_Features (Connection_Handle) the supported LMP Features of a remote BT Device can be read. These features contain supported packet types, supported modes, supported audio coding modes, etc.

Note: if the LMP Features was exchanged during ACL Connection Setup, the HC/LM A may return the Read Remote Supported Features Complete event (Status, Connection_Handle, LMP_Features) without exchange of LMP PDUs.

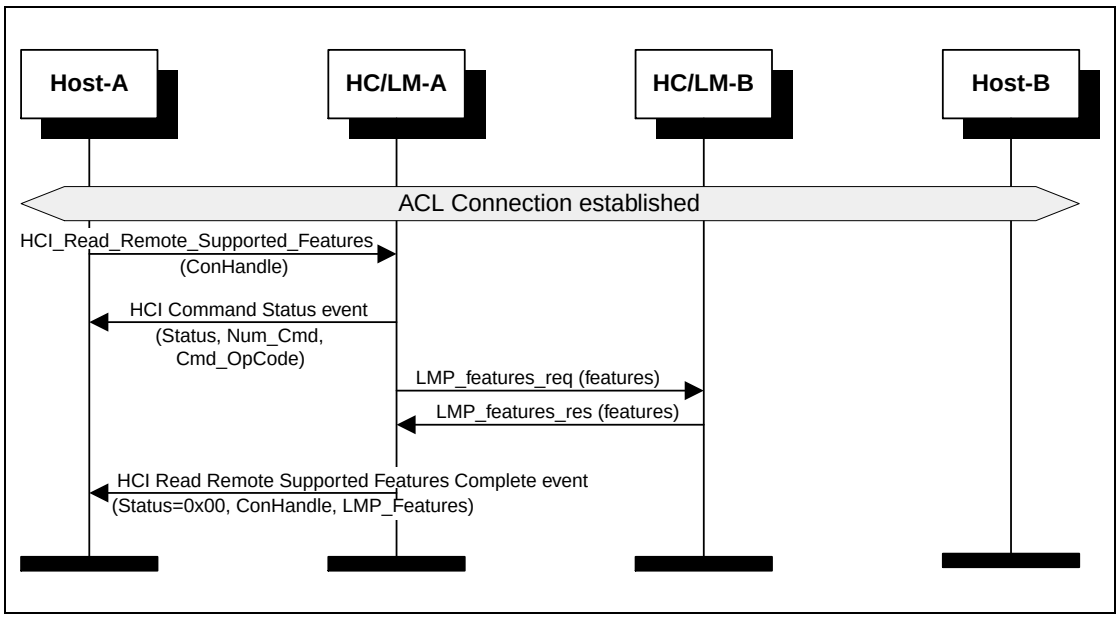


Figure 4.5: Read Remote Supported Features

4.6 READ CLOCK OFFSET

Using the command HCI_Read_Clock_Offset (Connection_Handle) the BT Master can read the Clock Offset of the BT Slaves. The Clock Offset can be used to speed up the paging procedure in a later connection attempt. If the command is requested from the slave device, the HC/LM Slave will directly return a Command Status event and an Read Clock Offset Complete event without exchange of LMP PDUs.

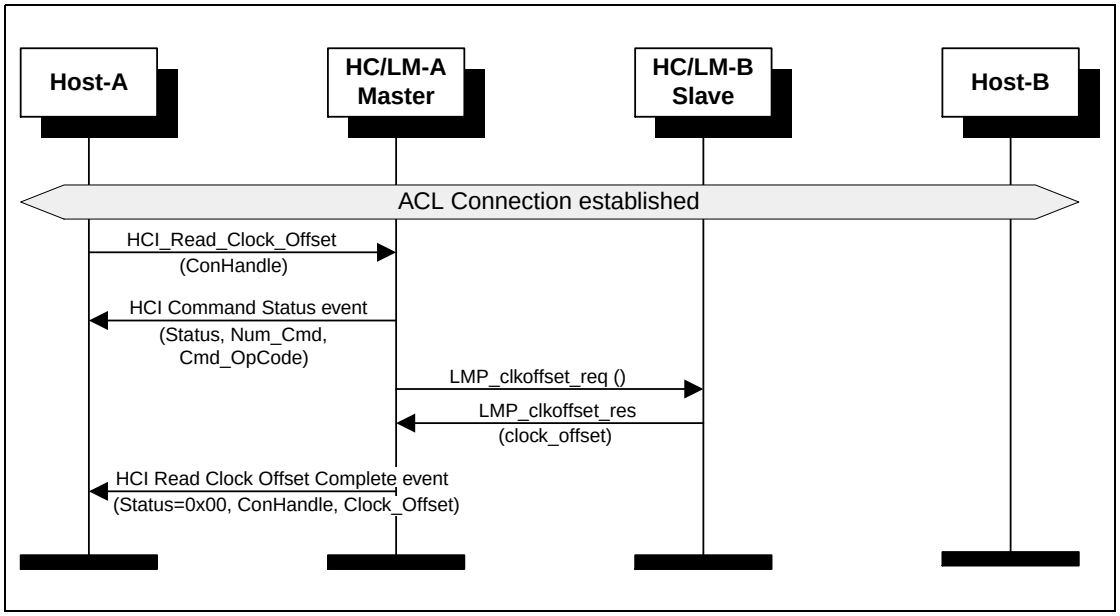


Figure 4.6: Read Clock Offset

4.7 READ REMOTE VERSION INFORMATION

Using HCI_Read_Remote_Version_Information (Connection_Handle) the version information consisting of LMP_Version, Manufacturer_Name and LMP_Subversion from the remote BT Device can be read.

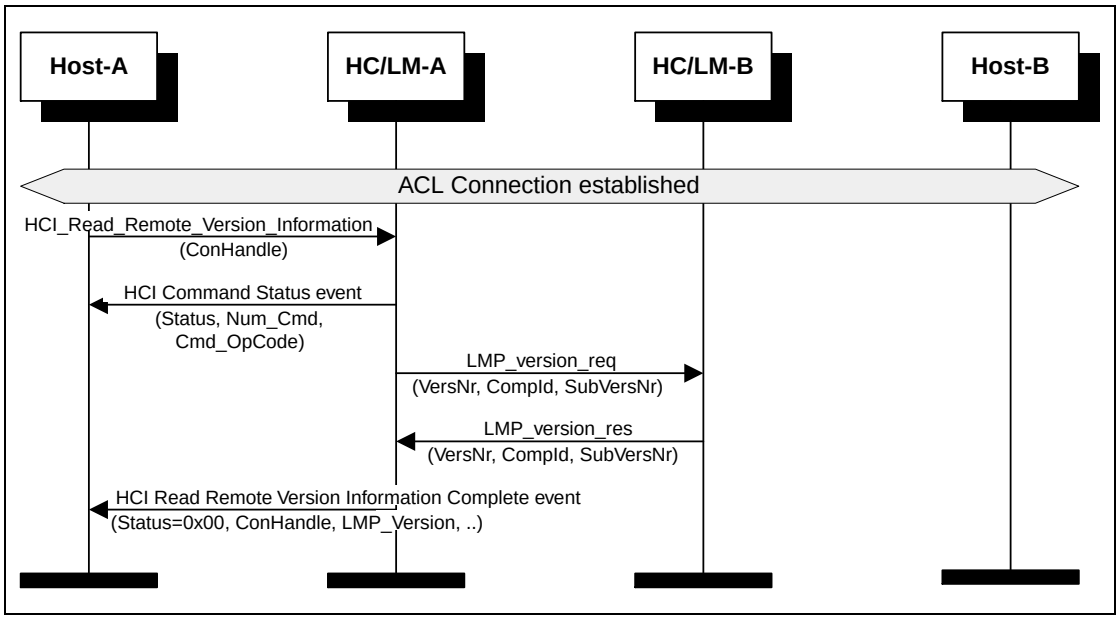


Figure 4.7: Read Remote Version Information

4.8 QOS SETUP

To set up the Quality of Service, the command HCI_QoS_Setup (Connection_Handle, Flags, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation) is used.

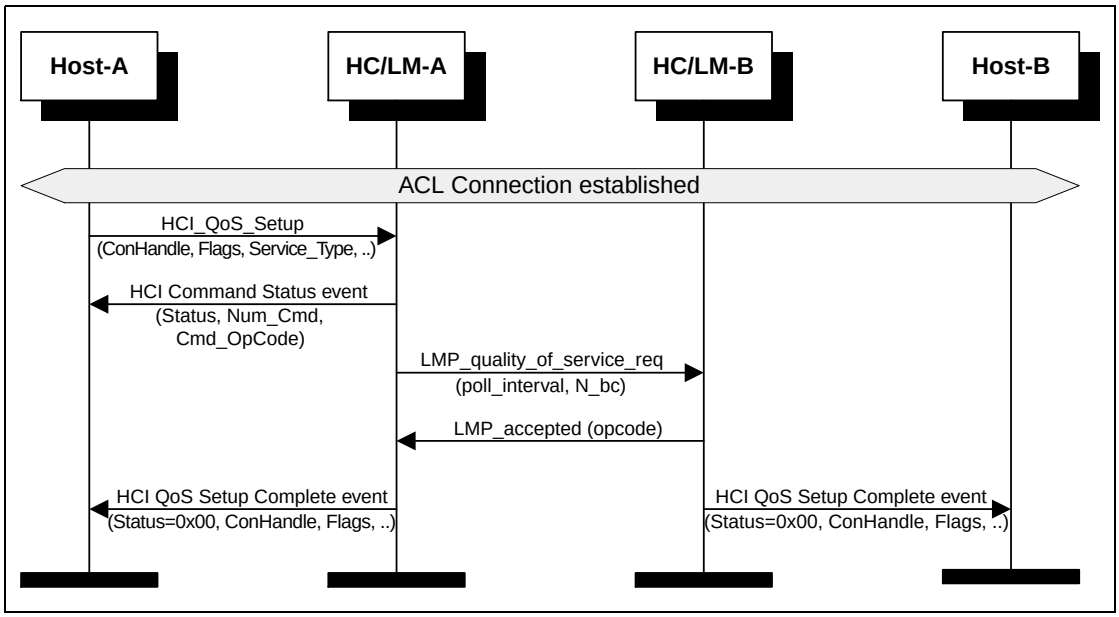


Figure 4.8: QoS Setup

4.9 SWITCH ROLE

The command HCI_Switch_Role (BD_ADDR, Role) can be used to explicitly switch the current role of the local BT Device for a particular connection with the specified BT Device (BD_ADDR). The local HC/LM has to check whether the switch is performed or not.

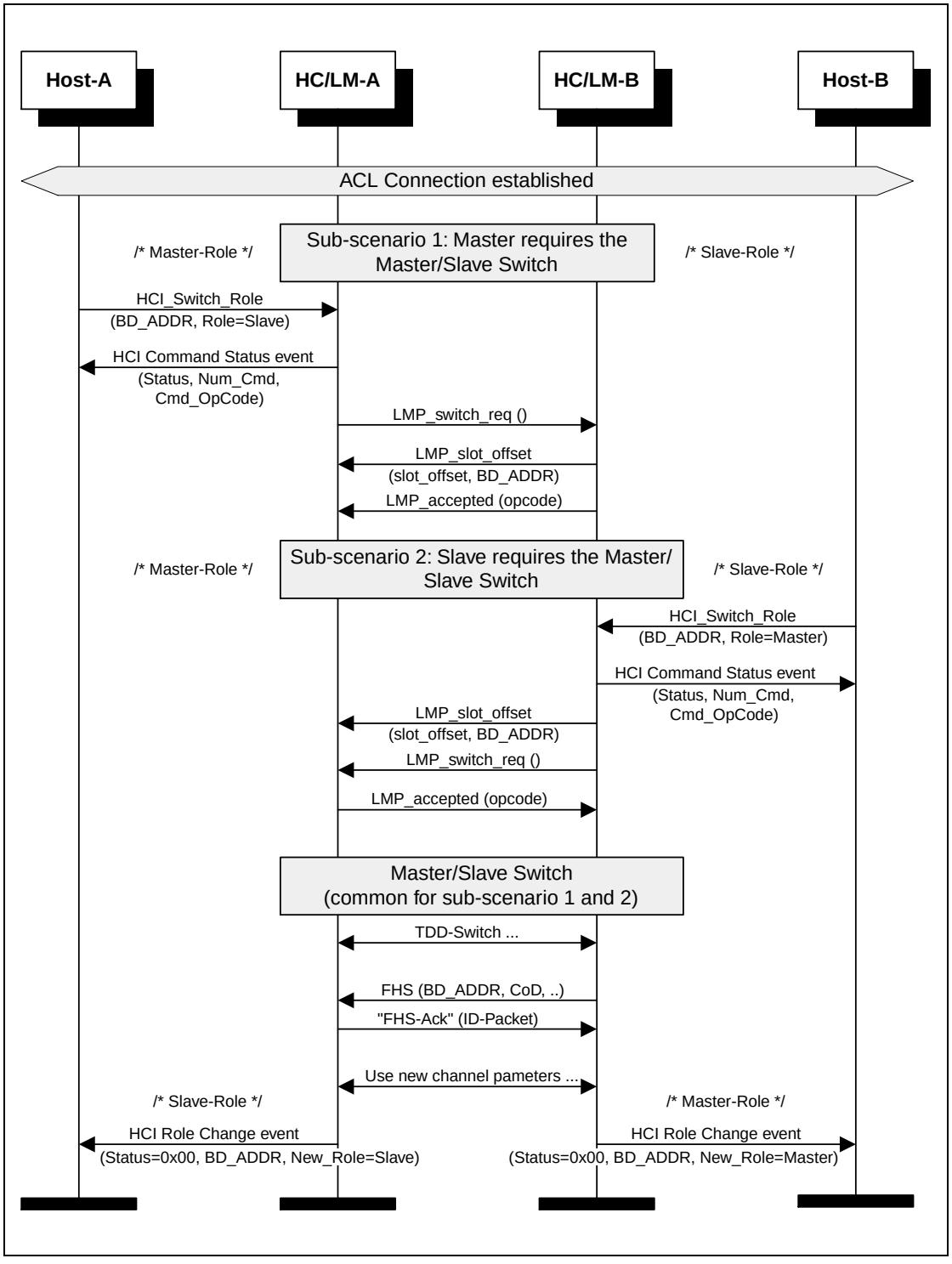


Figure 4.9: Switch Role

5 SCO CONNECTION ESTABLISHMENT AND DETACHMENT

5.1 SCO CONNECTION SETUP

SCO Connection setup requires an established ACL Connection. It is the task of the Host to create an ACL Connection first and then the SCO Link.

Note: On the slave side, an incoming connection request can be automatically accepted by using HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition) with the Filter_Type = 0x02 /*Connection_Setup*/. Furthermore, for each SCO Link to a BT Device, a separate SCO Connection Handle is needed.

5.1.1 Master activates the SCO Connection setup

To set up an SCO Connection, the HCI_Add_SCO_Connection (Connection_Handle, Packet_Type) command is used. The specified Connection_Handle is related to the ACL Connection that must have been created before the HCI_Add_SCO_Connection is issued.

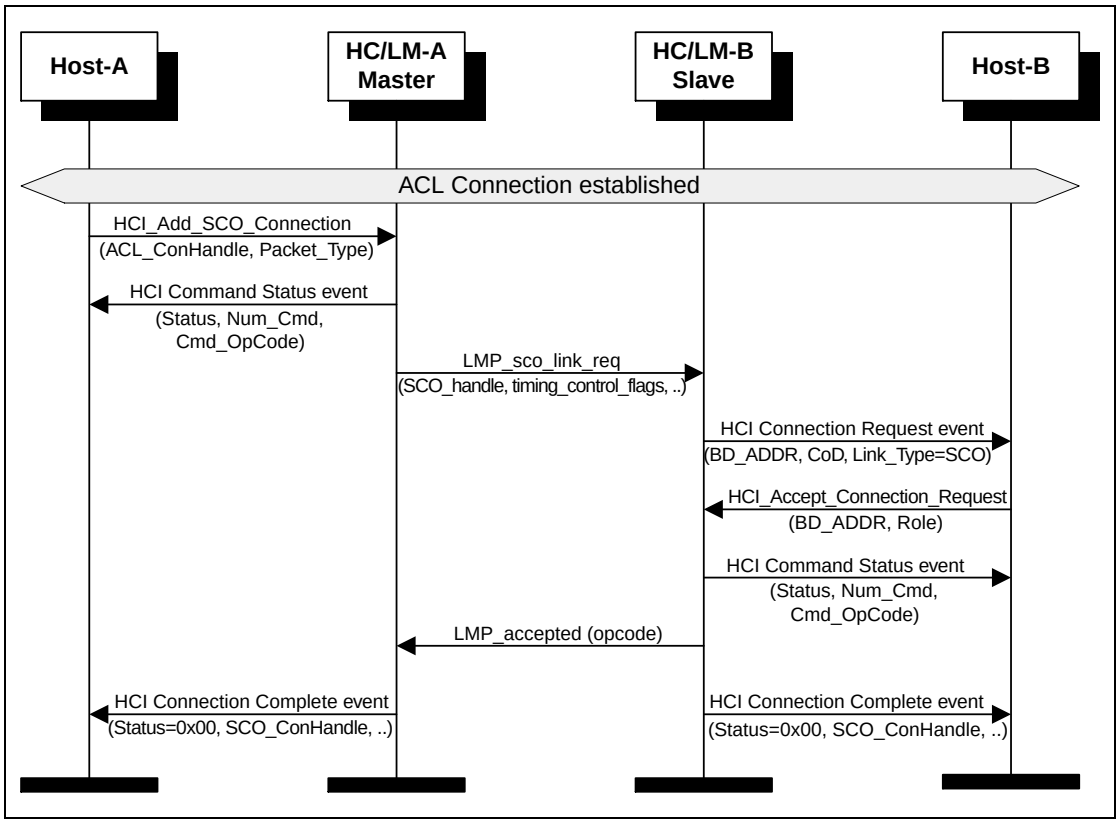


Figure 5.1: SCO Connection setup (activated from master)

5.1.2 Slave activates the SCO Connection setup

The same command HCI_Add_SCO_Connection (Connection_Handle, Packet_Type) can be used to create an SCO Link when the local BT Device is a BT Slave. Here the specified Connection_Handle belongs to the established ACL Connection between the BT Devices. Compared to 5.1.1, the only difference is that the HC/LM Slave starts the SCO Setup with LMP_sco_link_req first.

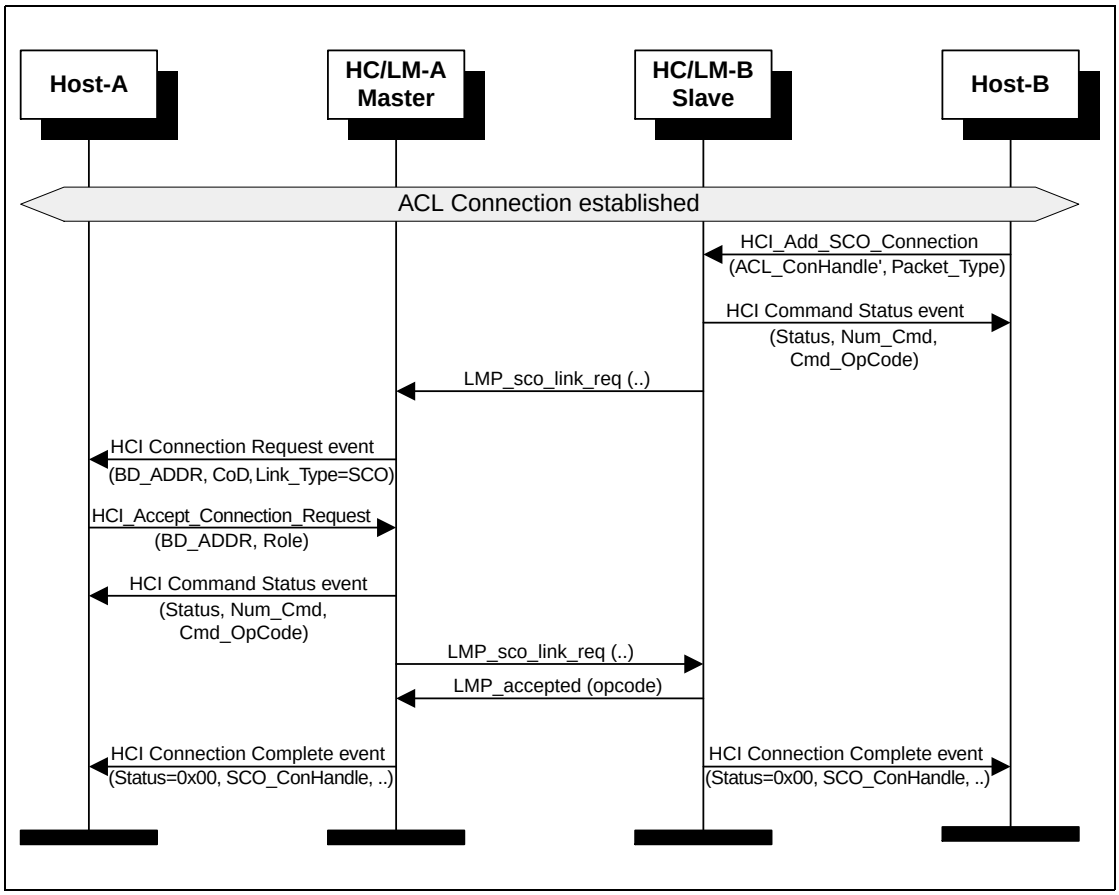


Figure 5.2: SCO Connection setup (activated from slave)

5.2 SCO DISCONNECTION

An established SCO Connection can be detached at any time. Since several SCO Connections can exist between a BT Master and a BT Slave, an SCO Disconnection only removes the SCO Link with the specified SCO Connection Handle. The other SCO Connections will still exist.

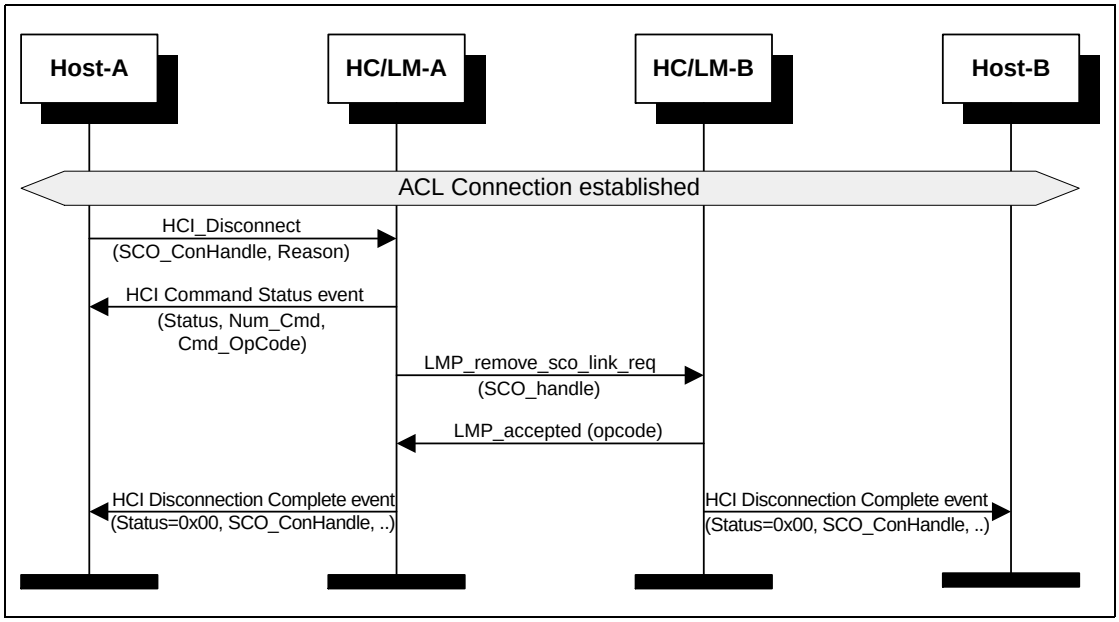


Figure 5.3: SCO Disconnection

6 SPECIAL MODES: SNIFF, HOLD, PARK

Entry into sniff, hold or park mode requires an established ACL Connection. The following table summarizes the modes and the BT Role that can request, force, activate or exit the modes.

	Sniff	**Hold**	**Park**
Request	Master/Slave	Master/Slave	Master/Slave
Force	Master	Master/Slave	Master
Activation	Master	Master/Slave	Master
Release	Master/Slave	Automatic	Master/Slave

Table 6.1: Summary of modes (Sniff, Hold, Park)

6.1 SNIFF MODE

Sniff Mode is used when a slave shall participate in the piconet only in a sniff interval. For the Sniff Mode negotiation, the Host specifies the Sniff_Max_Interval and the Sniff_Min_Interval so that HC/LM will be able to choose the one sniff interval in this range. The used command is HCI_Sniff_Mode (Connection_Handle, Sniff_Max_Interval, Sniff_Min_Interval, Sniff_Attempt, Sniff_Timeout).
Since Sniff Mode is a periodic mode, the command HCI_Exit_Sniff_Mode (Connection_Handle) is needed to return to Active Mode.

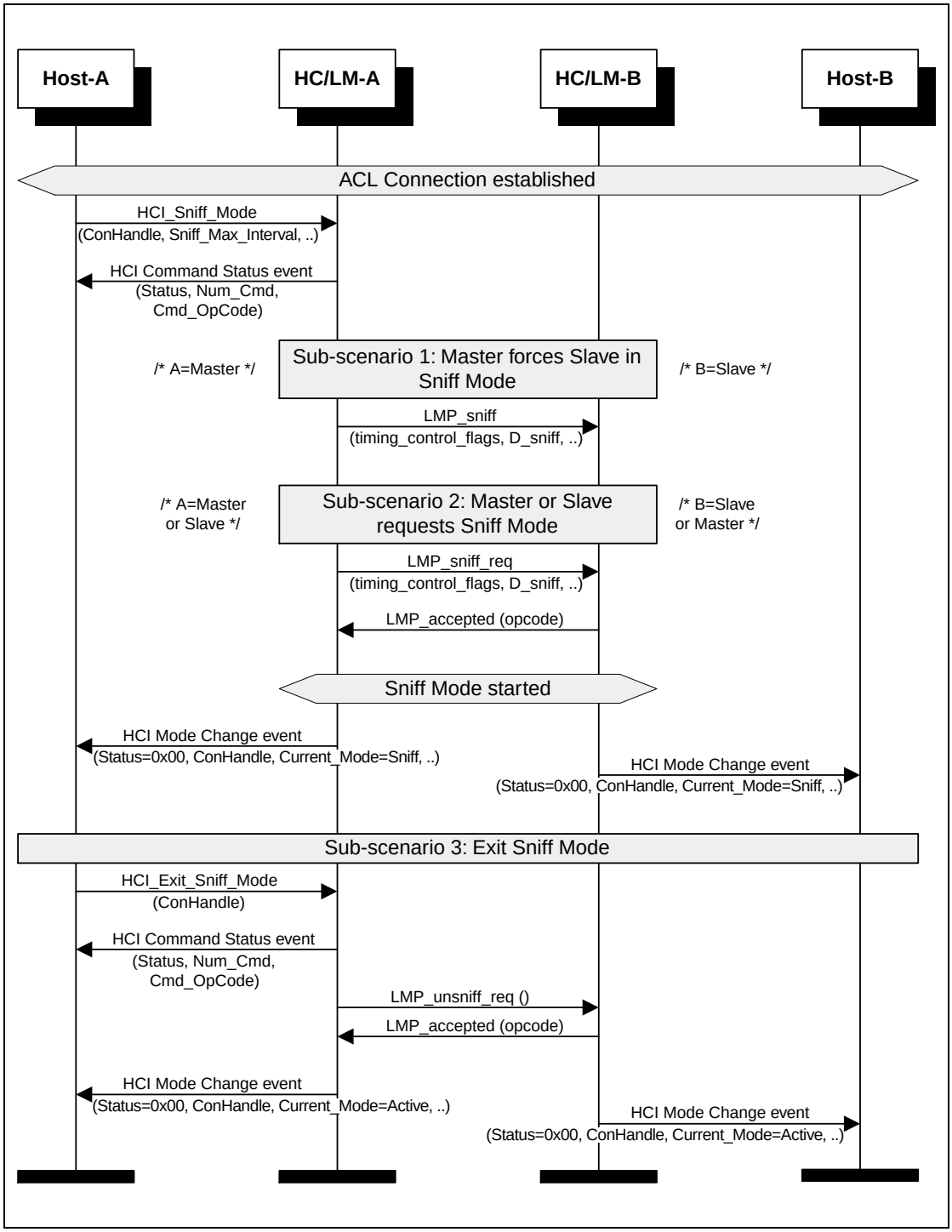


Figure 6.1: Sniff Mode

6.2 HOLD MODE

Hold Mode is useful when a BT Device doesn't want to participate in the connection for a Hold Mode Length. Using the command HCI_Hold_Mode (Connection_Handle, Hold_Max_Length, Hold_Min_Length), the Host specifies the Hold_Max_Length and Hold_Min_Length. The HC/LM will then be able to negotiate a Hold Mode Length in this range. When the hold mode is started

or complete, Mode Change event (Status, Connection_ Handle, Current_Mode, Interval) will be used to inform the Host about the actual mode.

Note: the Hold Mode is exited when the Hold Mode Length has expired, so it is no guarantee that the remote BT Device is immediately active.

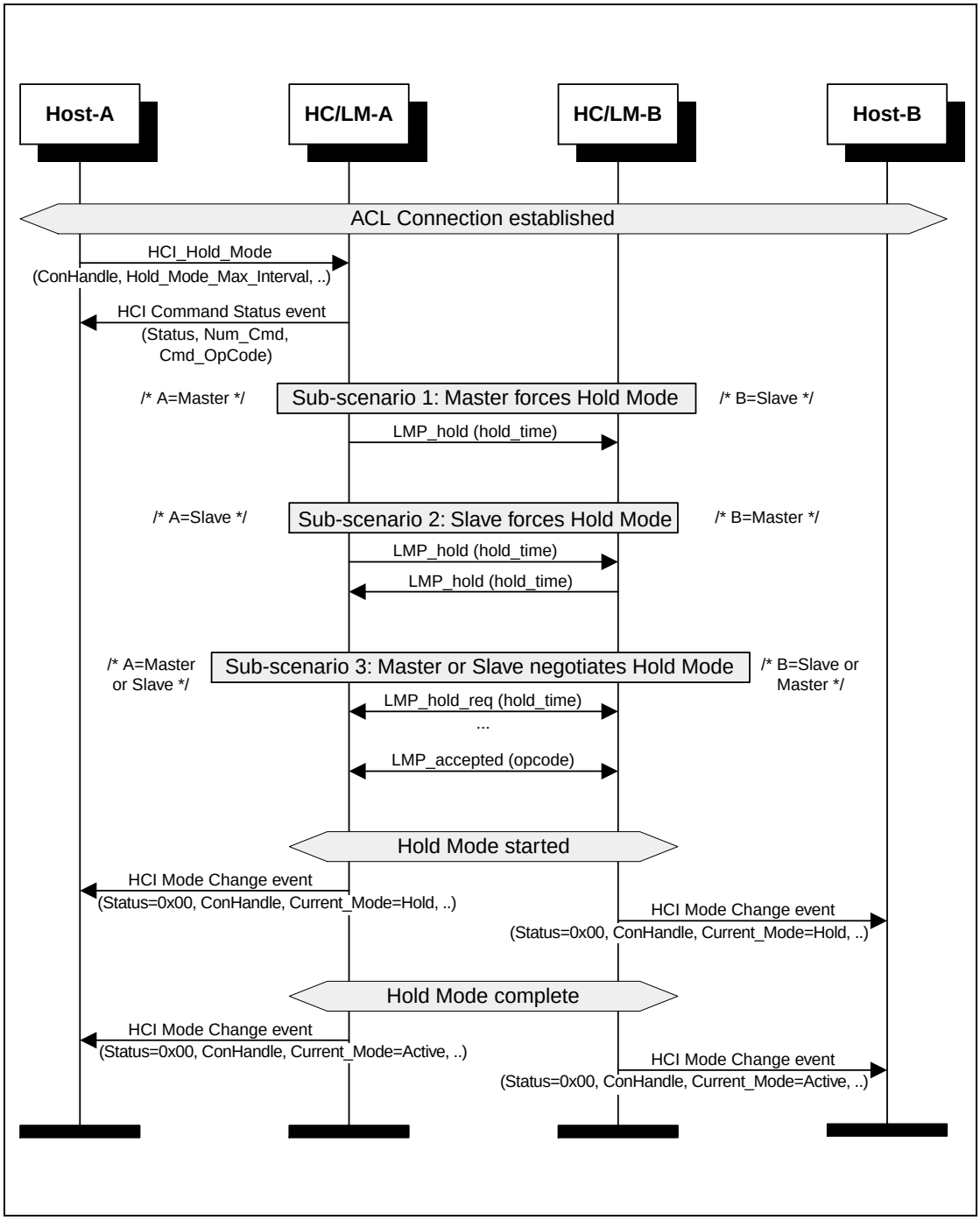


Figure 6.2: Hold Mode

6.3 PARK MODE

Park Mode is used to render the slaves inactive but still synchronized to the master using the beacon interval. In park mode, broadcast is performed.

6.3.1 Enter park mode

Using the command HCI_Park_Mode (Connection_Handle, Beacon_Max_Interval, Beacon_Min_Interval) the Host specifies the Beacon_Max_Interval and Beacon_Min_Interval so that HC/LM can set up a Beacon-Interval in this range for the BT Slaves. In Park Mode, the BT Slave gives up its AM_ADDR.

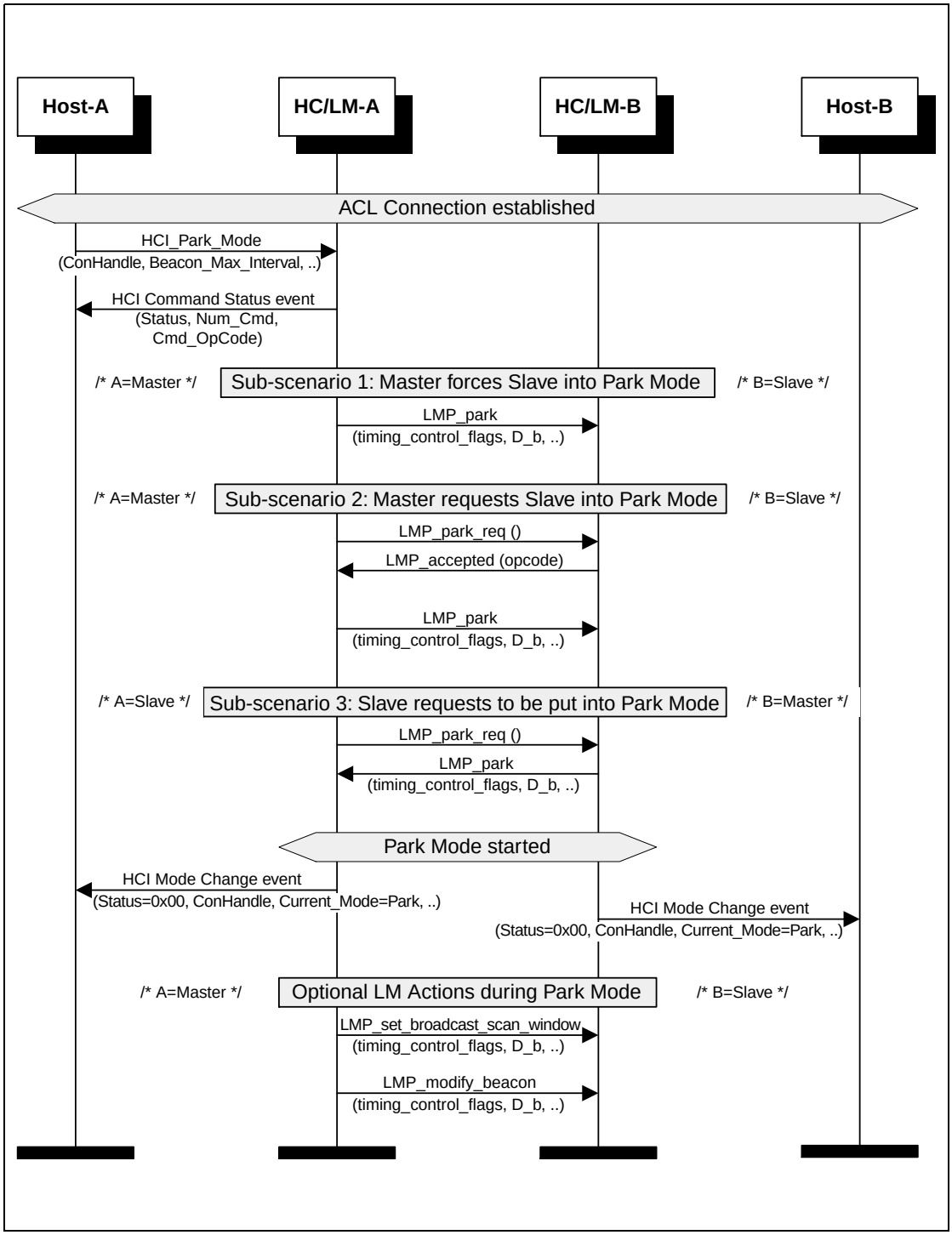


Figure 6.3: Enter Park Mode

6.3.2 Exit Park Mode

Since Park Mode is a periodic mode, the command HCI_Exit_Park_Mode (Connection_Handle) will be used to return to Active Mode. A parked BT Slave can send an Access_Request_Message to request to leave the Park Mode. It is the task of master HC/LM to use LMP_unpark_PM_ADDR_req (..) or LMP_unpark_BD_ADDR_req (..) to unpark a BT Slave.

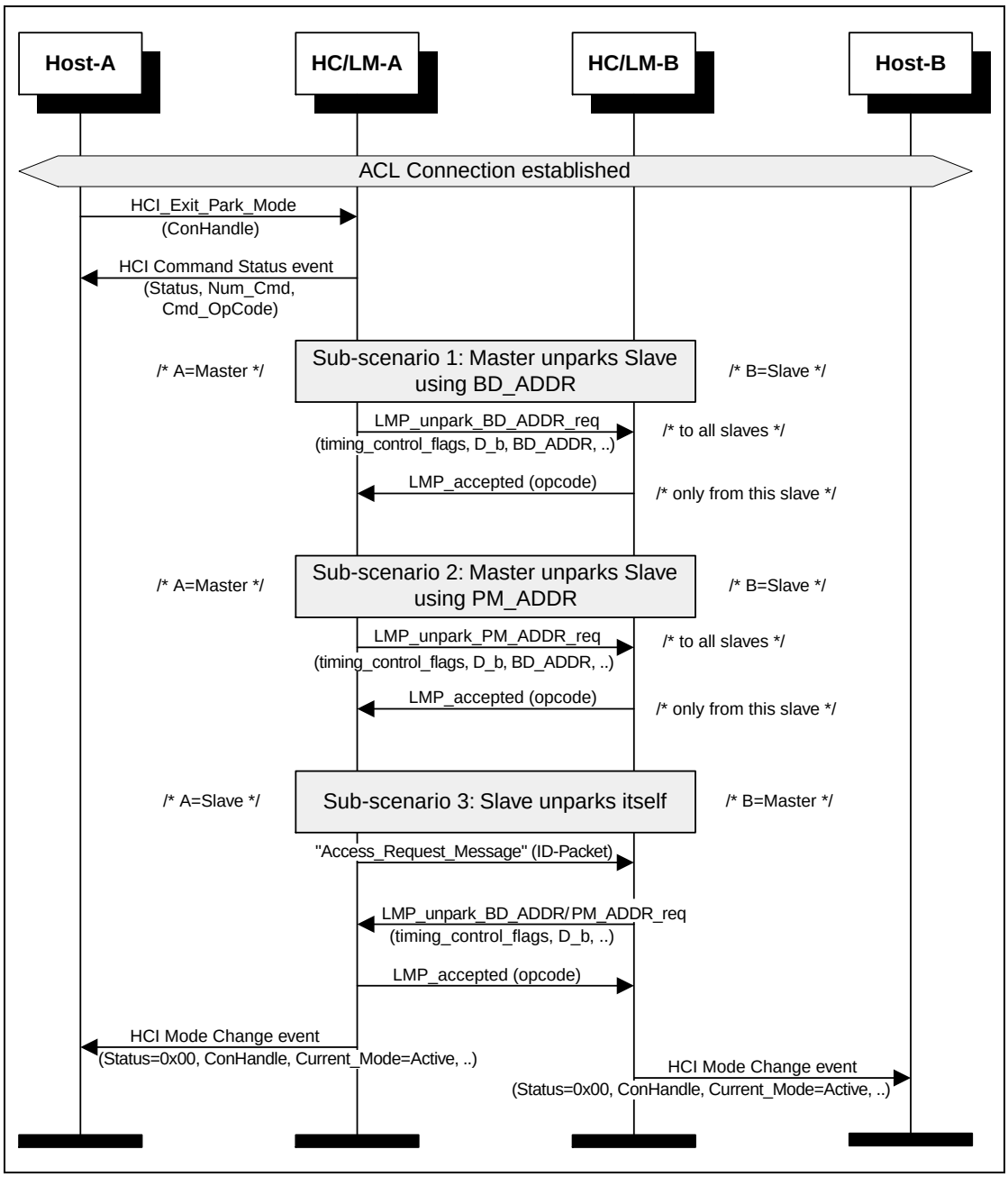


Figure 6.4: Exit Park Mode

7 BUFFER MANAGEMENT, FLOW CONTROL

The HC Data buffers are configured by the HC and managed by the Host. On initialization, the Host will issue HCI_Read_Buffer_Size. This specifies the maximum allowed length of HCI data packets sent from the Host to the HC, and the maximum number of ACL and SCO data packets that the HC can store in its buffer. After a connection is created, HC will frequently inform the Host about the number of sent packets using Number Of Completed Packets event (Number_of_Handles, Connection_Handle[I], HC_Num_Of_ Completed_Packets[i]) (see Figure 7.1 Host-to-HC flow control).

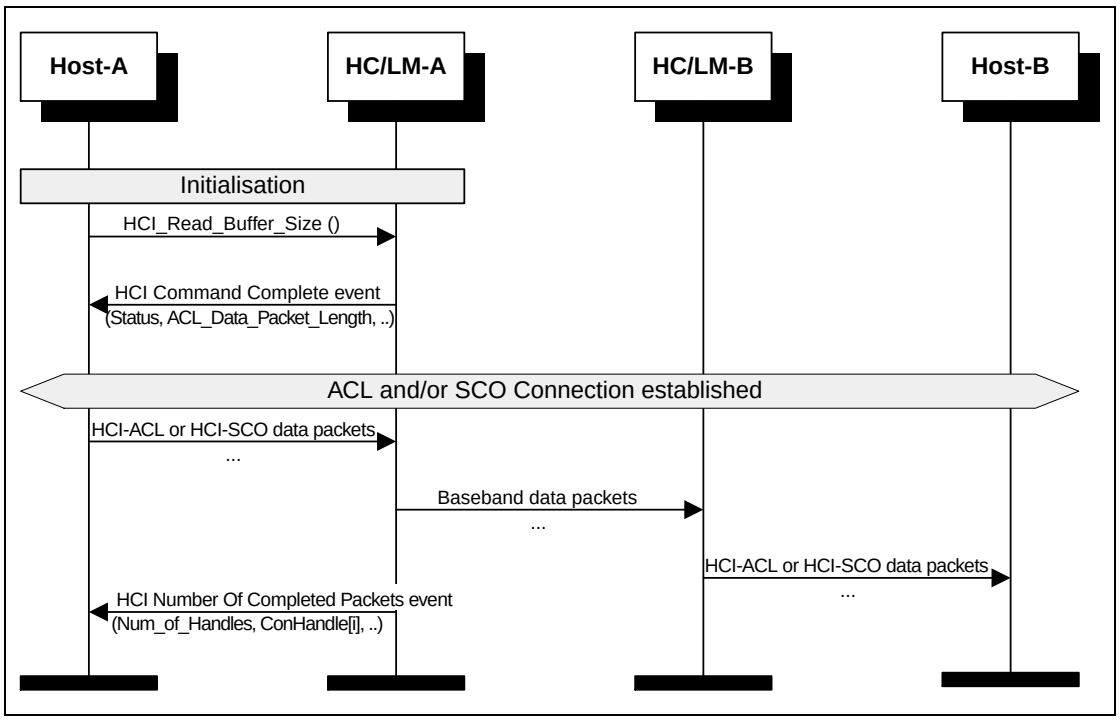


Figure 7.1: Host to HC flow control

Accordingly the HC to Host flow control can be applied in the same way so that during initialization the Host configures the Buffer Size and later the Host Controller will manage the Host Buffers.
Using HCI_Set_Host_Controller_To_Host_Flow_Control (Flow_Control_Enable) the Host can decide to apply the HC to Host flow control or not. For flow control itself HCI_Host_Buffer_Size (Host_ACL_Data_Packet_Length, Host_SCO_Data_Packet_Length, Host_Total_Num_ACL_Data_Packets, Host_Total_Num_SCO_Data_Packets) and HCI_Host_Number_Of_Completed_Packets (Number_of_Handles, Connection_Handle[I], Host_Num_Of_Completed_Packets[I]) will be used (for details see Figure 7.2 HC to Host Flow Control).

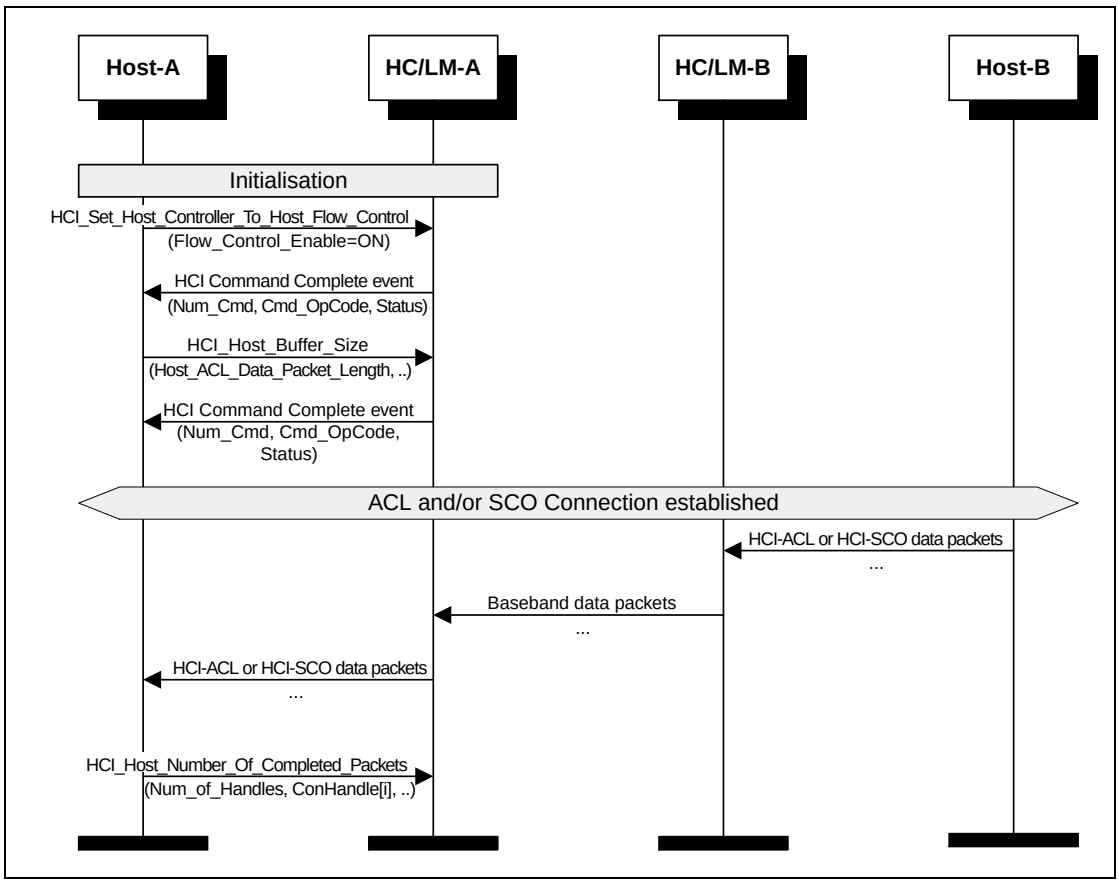


Figure 7.2: HC to Host Flow Control

8 LOOPBACK MODE

8.1 LOCAL LOOPBACK MODE

The local Loopback Mode is used to loopback received HCI Commands, and HCI ACL and HCI SCO packets sent from the Host.

The HC will send four Connection Complete events (one for ACL, three for SCO Connections) so that the Host can use the Connection_Handles to re-send HCI ACL and HCI SCO Packet to HC. To exit the local Loopback Mode, HCI_Write_Loopback_Mode (Loopback_Mode=0x00) or HCI_Reset ( ) will be used.

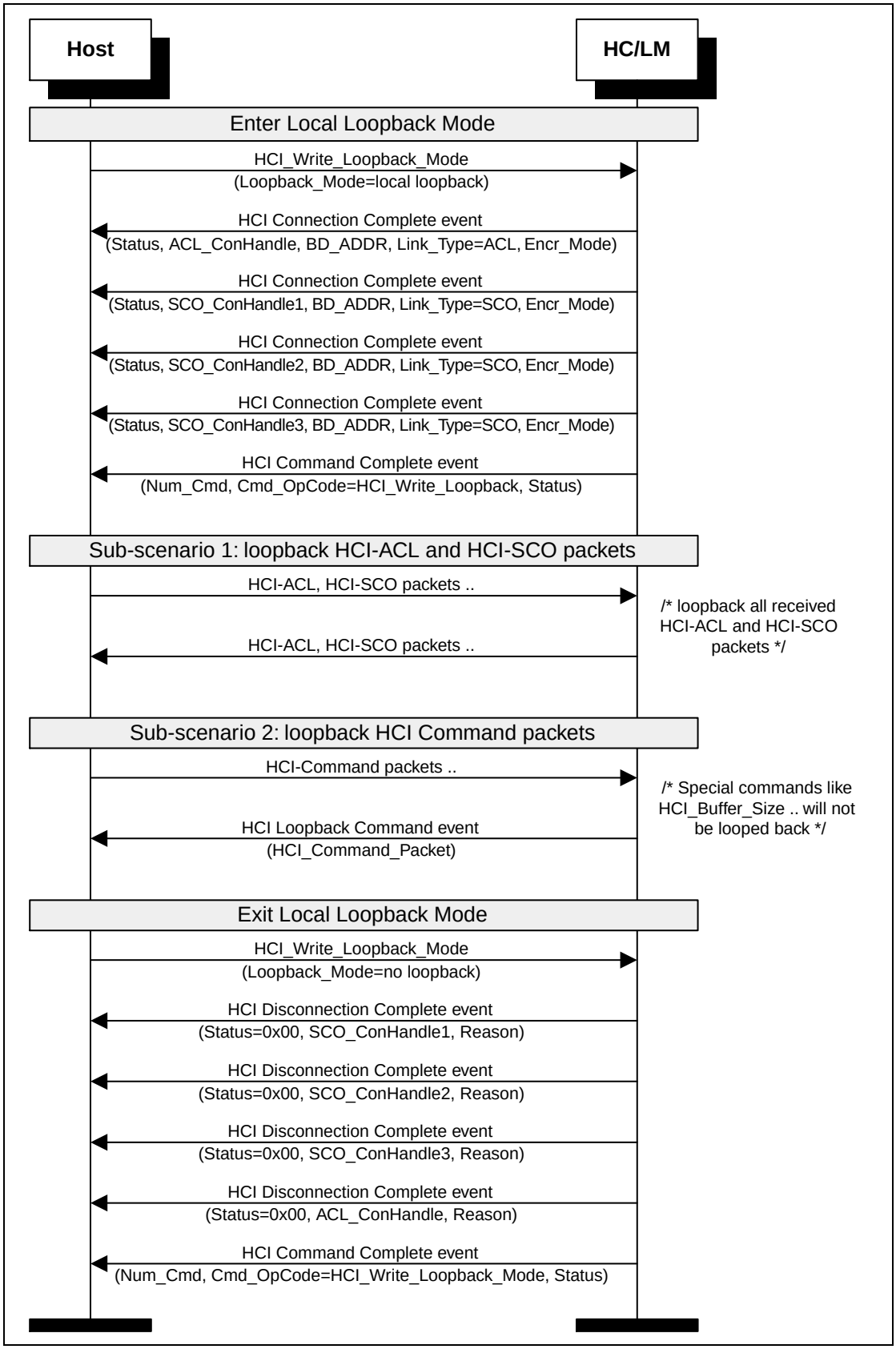


Figure 8.1: Local Loopback Mode

8.2 REMOTE LOOPBACK MODE

The remote Loopback Mode is used to loopback all received Baseband ACL and SCO Data received from a remote BT Device. During remote Loopback Mode, ACL and SCO Connection can be created. The remote Loopback Mode can be released with the command HCI_Write_Loopback_Mode (Loopback_Mode=0x00).

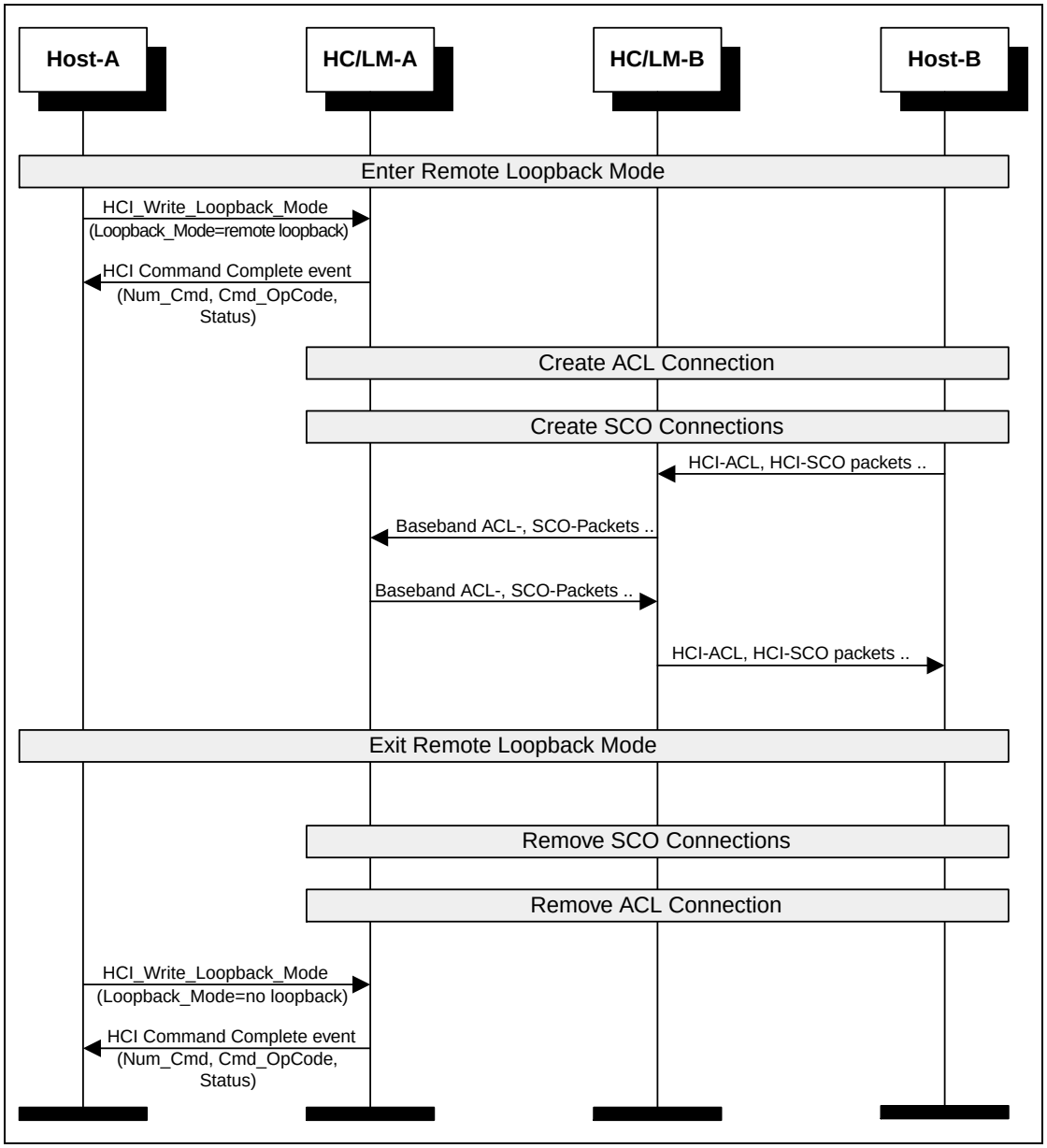


Figure 8.2: Remote Loopback Mode

9 LIST OF ACRONYMS AND ABBREVIATIONS

BT	Bluetooth
HC	Host Controller
HCI	Host Controller Interface
LAP	Lower Address Part
LC	Link Controller
LM	Link Manager
LMP	Link Manager Protocol
MSC	Message Sequence Chart
PDU	Protocol Data Unit

10 LIST OF FIGURES

Figure 2.1: Remote Name Request 1035
Figure 2.2: One-Time Inquiry 1036
Figure 2.3: Periodic Inquiry 1037
Figure 3.1: Overview of ACL Connection establishment and detachment 1038
Figure 3.2: ACL Connection Request phase 1040
Figure 3.3: ACL Connection setup with pairing 1042
Figure 3.4: ACL Connection setup with authentication 1043
Figure 3.5: Encryption and Setup complete 1044
Figure 3.6: ACL Disconnection 1045
Figure 4.1: Authentication Requested 1046
Figure 4.2: Set Connection Encryption 1048
Figure 4.3: Change Connection Link Key 1049
Figure 4.4: Master Link Key 1050
Figure 4.5: Read Remote Supported Features 1051
Figure 4.6: Read Clock Offset 1052
Figure 4.7: Read Remote Version Information 1052
Figure 4.8: QoS Setup 1053
Figure 4.9: Switch Role 1054
Figure 5.1: SCO Connection setup (activated from master) 1055
Figure 5.2: SCO Connection setup (activated from slave) 1056
Figure 5.3: SCO Disconnection 1057
Figure 6.1: Sniff Mode 1059
Figure 6.2: Hold Mode 1060
Figure 6.3: Enter Park Mode 1062
Figure 6.4: Exit Park Mode 1063
Figure 7.1: Host to HC flow control 1064
Figure 7.2: HC to Host Flow Control 1065
Figure 8.1: Local Loopback Mode 1067
Figure 8.2: Remote Loopback Mode 1068

11 LIST OF TABLES

Table 6.1: Summary of modes (Sniff, Hold, Park).....1058

12 REFERENCES

[1] “Baseband Specification” on page 33

[2] “Link Manager Protocol” on pag e183

[3] “Host Controller Interface Functional Specification” on page 535

[4] “Logical Link Control and Adaptation Protocol Specification” on page253

Alphabetical Index

Numerics

0x7E 808

A

Abort- 436
ACCESS RIGHTS ACCEPT 467
ACCESS RIGHTS REJECT 467
ACCESS RIGHTS REQUEST 467
Ack Code 805
Ack code 804
Acknowledgement Timer (T1) 412
ALERTING 458
asynchronous notifications 526
AT+CMUX 410
authentication 196

B

basic option 406, 408
Baud Rate 805
baud rate 401, 804
beacon 218
beginning delimiter 809
Bluetooth 430
BOF(0x7E) 810
Briefcase Trick 518
business card 431
byte ordering 400
byte stream 437

C

Call Control 451
CALL PROCEEDING 457
Calling Line Identity 473
checksum 809
CL INFO 472
claimant 196
clock offset 206
COBS 808, 809, 811
COBS code block 811
COBS code byte 811
commands in TS 07.10 406
Configuration distribution 465
CONNECT 458
Connect-request 434
Consistent Overhead Byte Stuffing 809
control channel 406
convergence layer 407
CRC 806
CRC-CCITT 809, 810
CTS 812
current link key 201

D

Data Link Connection Identifier 403
data throughput 401
DCE 402
default URL 527
delayed loopback 838
Delimiter 806
delimiter 0x7E 809
delimiter, 0x7E 808
direction bit 413
DISC command 414
DISC command frame 410
DISCONNECT 462
Disconnect-request 435
DNS 521
drift 207
DTE 402, 411
DTMF ACKNOWLEDGE 474
DTMF start & stop 473
DTR/DSR 417

E

EIATIA-232-E 399, 401
eliminating zeros 809
emergency call 496
emulated ports 403
encryption 203
ending delimiter 809
EOF(0x7E) 810
Error detection 804
error message packet 809, 813
Error Message Packet (0x05) 803
error packet 812
Error Recovery 807
error recovery 804, 808
error recovery procedure 809, 812
Error Type 810, 813
ETSI TS 07.10 437
external call 496

F

Fast inter member access 465

FCoff 417
FCon 417
flow control 417
Forbidden Message 519
frame types 406

G

generator polynomial 809
Get-request 436
Group Management 451

H

HCI RS232 Transport Layer 802
header ID 433
hold mode 213
Host Controller Interface 526
HTML 523
HTTP 521, 523

I

in-band tones/announcements 459
INFO ACCEPT 468
INFO SUGGEST 468
INFORMATION 457
initialisation key 197
intercom call 496
Internet Engineering Task Force (IETF) 522
interoperability 529
interrupt latency 804
IrCOMM 432
IrDA 430
IrMC 441
IrOBEX 430

J

JavaScript 523
jitter 207

L

L2CAP channel 423
latency requirements 423
link key 196
link loss notification 411, 423
link supervision 233
LISTEN REJECT 470
LISTEN REQUEST 469
LISTEN SUGGEST 469
loop back test 841
low power mode 423

M

Management Entity 526
Maximum Frame Size (N1) 412
Modem Status Command 407
multiple bearers 526
multiplexer control channel 410
Multiplexer Control commands 414

N

name request 212
negotiation packet 804, 805
Negotiation Packet (0x06) 803
negotiation phase 804
null modem 402
null modem emulation 401
number of data bit 804
number of stop bit 804

O

OBEX 430
OBEX session protocol 433
Obtain access rights 465
output power 224

P

paging scheme 232
parity type 804
park mode 218
payload heade 193
PIN 197
PN command 415
port emulation entity 421
port proxy entity 421
Protocol Mode 806
protocol mode 804
protocol mode 0x13 809
protocol mode 0x14 812
Proxy/gateway Addressing 527
Put-request 435

Q

Q.931 451
Quality of Service 227

R

register recall 473
RELEASE 462
RELEASE COMPLETE 462
reliability 423
reliable transmission 412

Response Timer for Multiplexer Control Channel (T2) 412
resynchronization 808
resynchronize 812
retransmission holding buffer 809, 813
retransmission packets 809, 812
RFCOMM 430
RFCOMM entity 403
RFCOMM multiplexer 410
RFCOMM reference model 405
RFCOMM Server Channel 421
RFCOMM server channels 403, 413
RFCOMM session 403, 410
RLS command 415
RPN command 414
RS-232 399, 401, 421
RS232 802
RS-232 control signals 402, 407
RS232 Transport Packet 803
RSSI 224
RTS 812
RTS/CTS 417, 804
RTS/CTS Mode 806

S

SABM command 410
SCO link 228
semi-permanent link key 201, 202
SEQ No with Error 809
sequence number 803
sequence number field 809, 812
Sequence Number with Error field 809, 812
serial port emulation entity 405
service call 496
Service Discovery Protocol 524, 530
service records 421
SetPath- 436
SETUP 455
SETUP ACKNOWLEDGE 457
simple error recovery scheme 812
Smart Kiosk 519
sniff mode 216
SSL 523
START DTMF 474
START DTMF REJECT 474
STOP DTMF 474
STOP DTMF ACKNOWLEDGE 474
supervision timeout 233
synchronization 809
synchronize 812

T

TCP 523
TCP port number 439
TCP/IP 430
TCS Binary 451
Tdetect 804
Tdetect Time 806
Tdetect time 812
temporary link key 201
test mode 248, 832
Tiny TP 432
transmitter test 837
TS 07.10 399
TS 07.10 multiplexer 403, 423

U

UART 804
UART Settings 805
UDP 522
Uniform Resource Locators 527
unit key 200
URL 525
User Addressing 527

V

vCalendar 431
vCard 431
verifier 196
vMessage 431
vNotes 431

W

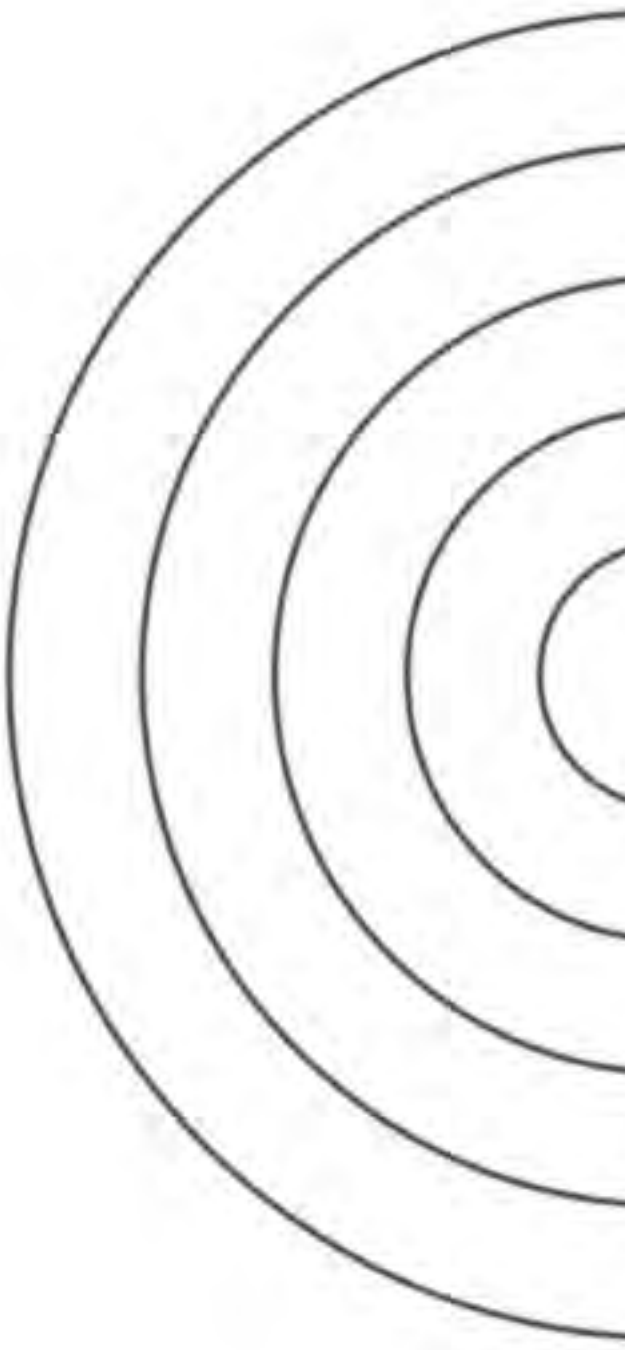
WAP Client 520
WAP Proxy/gateway 521
WAP Server 521
WDP 522
Wireless User Group 465
WSP 522
WTLS 522
WTP 522
WUG 465

X

XML 523
XON/XOFF 417

Z

zero elimination 811



https://web.archive.org/web/20011217082346/http://www.bluetooth.com/pdf/Bluetooth_11_Profiles_Book.pdf

Specification of the Bluetooth System

Wireless connections made easy

Profiles

Bluetooth™

Version 1.1
February 22 2001

Revision History

The Revision History is shown in Appendix I on page 423

Contributors

The persons who contributed to this specification are listed in Appendix II on page 433.

Web Site

This specification can also be found on the web site for Bluetooth adopters: http://www.bluetooth.com.

Disclaimer and Copyright Notice

The copyright in these specifications is owned by the Promoter Members of Bluetooth SIG, Inc. (“Bluetooth SIG”). Use of these specifications and any related intellectual property (collectively, the “Specification”), is governed by the Promoters Membership Agreement among the Promoter Members and Bluetooth SIG (the “Promoters Agreement”), certain membership agreements between Bluetooth SIG and its Adopter and Associate Members (the “Membership Agreements”) and the Bluetooth Specification Early Adopters Agreements (“1.2 Early Adopters Agreements”) among Early Adopter members of the unincorporated Bluetooth special interest group and the Promoter Members (the “Early Adopters Agreement”). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to Bluetooth SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of Bluetooth SIG or a party to an Early Adopters Agreement (each such person or party, a “Member”), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by aplicable law to Bluetooth SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WAR-

RANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the Bluetooth™ technology ("Bluetooth™ Products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of Bluetooth™ Products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Bluetooth™ Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Bluetooth™ Products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate and to adopt a process for adding new Bluetooth™ profiles after the release of the Specification.

Copyright © 1999, 2000, 2001

3Com Corporation,
Agere Systems, Inc.,
Ericsson Technology Licensing, AB,
IBM Corporation,
Intel Corporation,
Microsoft Corporation,
Motorola, Inc.,
Nokia Mobile Phones,
Toshiba Corporation

*Third-party brands and names are the property of their respective owners.

MASTER TABLE OF CONTENTS

For the Core Specification, *see Volume 1*

Part K:1

GENERIC ACCESS PROFILE

Contents 15

Foreword 19
1 Introduction 20
2 Profile overview 22
3 User interface aspects 25
4 Modes 29
5 Security aspects 33
6 Idle mode procedures 37
7 Establishment procedures 45
8 Definitions 52
9 Annex A (Normative): Timers and constants 56
10 Annex B (Informative): Information flows of related procedures 57
11 References 60

Part K:2

SERVICE DISCOVERY APPLICATION PROFILE

Contents 63

Foreword 65
1 Introduction 66
2 Profile overview 68
3 User interface aspects 72
4 Application layer 73
5 Service Discovery 79
6 L2CAP 82
7 Link Manager 86
8 Link control 88
9 References 91
10 Definitions 92
11 Appendix A (Informative): Service primitives and the Bluetooth PDUs 93

Part K:3

CORDLESS TELEPHONY PROFILE

Contents 97

1 Introduction 100
2 Profile overview 103
3 Application layer 108
4 TCS-BIN procedures 110
5 Service Discovery procedures 120
6 L2CAP procedures 121
7 LMP procedures overview 122
8 LC features 124
9 Generic Access Profile Interoperability Requirements 126
10 Annex A (Informative): Signalling flows 128
11 Timers and counters 135
12 References 136
13 List of Figures 137
14 List of Tables 138

Part K:4

INTERCOM PROFILE

Contents 141

1 Introduction 143
2 Profile Overview 145
3 Application layer 148
4 TCS Binary 149
5 SDP Interoperability Requirements 153
6 L2CAP Interoperability Requirements 154
7 Link Manager (LM) Interoperability Requirements 155
8 Link Control (LC) Interoperability Requirements 156
9 Generic Access Profile 158
10 Annex A (Informative): Signalling flows 159
11 Timers and counters 161
12 List of Figures 162
13 List of Tables 163

Part K:5

SERIAL PORT PROFILE

Contents 167

Foreword 169

1 Introduction 170

2 Profile overview 171

3 Application layer 174

4 RFCOMM Interoperability Requirements 177

5 L2CAP Interoperability Requirements 179

6 SDP Interoperability Requirements 181

7 Link Manager (LM) Interoperability Requirements 183

8 Link Control (LC) Interoperability Requirements 184

9 References 186

10 List of Figures 187

11 List of Tables 188

Part K:6

HEADSET PROFILE

Contents 191

1 Introduction 193

2 Profile Overview 196

3 Application layer 200

4 Headset Control Interoperability Requirements 201

5 Serial Port Profile 211

6 Generic Access Profile 215

7 References 216

8 List of Figures 217

9 List of Tables 218

Part K:7

DIAL-UP NETWORKING PROFILE

Contents 221

1 Introduction 223
2 Profile overview 226
3 Application layer 230
4 Dialling and Control Interoperability Requirements 231
5 Serial Port Profile Interoperability Requirements 235
6 Generic Access Profile Interoperability Requirements 238
7 References 240
8 List of Figures 241
9 List of Tables 242

Part K:8

FAX PROFILE

Contents 245

1 Introduction 246
2 Profile overview 249
3 Application layer 253
4 Dialling and Control Interoperability Requirements 254
5 Serial Port Profile 256
6 Generic Access Profile Interoperability Requirements 259
7 References 261
8 List of Figures 262
9 List of Tables 263

Part K:9

LAN ACCESS PROFILE

Contents 267

1 Introduction 269
2 Profile overview 271
3 User interface aspects 275
4 Application layer 278
5 PPP 281
6 RFCOMM 284
7 Service Discovery 285
8 L2CAP 287
9 Link Manager 288
10 Link control 290
11 Management Entity Procedures 291
12 APPENDIX A (Normative): Timers and counters 293
13 APPENDIX B (Normative): Microsoft Windows 294
14 APPENDIX C (Informative): Internet Protocol (IP) 295
15 List of Figures 297
16 List of Tables 298
17 References 299

Part K:10

GENERIC OBJECT EXCHANGE PROFILE

Contents 303

Foreword 305
1 Introduction 306
2 Profile overview 310
3 User interface aspects 312
4 Application layer 313
5 OBEX Interoperability Requirements 314
6 Serial Port Profile Interoperability Requirements 324
7 Generic Access Profile Interoperability Requirements 326
8 References 328

Part K:11

OBJECT PUSH PROFILE

Contents 331

Foreword 333
1 Introduction 334
2 Profile overview 338
3 User interface aspects 340
4 Application layer 344
5 OBEX 348
6 Service Discovery 351
7 References 353

Part K:12

FILE TRANSFER PROFILE

Contents 357

Foreword 359
1 Introduction 360
2 Profile overview 364
3 User interface aspects 367
4 Application layer 370
5 OBEX 374
6 Service Discovery 383
7 References 385

Part K:13

SYNCHRONIZATION PROFILE

Contents 389

Foreword 391
1 Introduction 392
2 Profile overview 396
3 User interface aspects 399
4 Application layer 402
5 IrMC Synchronization Requirements 404
6 OBEX 406
7 Service Discovery 408
8 References 411

Appendix I

REVISION HISTORY 415

Appendix II

CONTRIBUTORS 425

Appendix III

LIST OF ACRONYMS AND ABBREVIATIONS 433

Index

INDEX 437

Part K:1

GENERIC ACCESS PROFILE

This profile defines the generic procedures related to discovery of Bluetooth devices (idle mode procedures) and link management aspects of connecting to Bluetooth devices (connecting mode procedures). It also defines procedures related to use of different security levels. In addition, this profile includes common format requirements for parameters accessible on the user interface level.

CONTENTS

1 Introduction 20
1.1 Scope 20
1.2 Symbols and conventions 20
1.2.1 Requirement status symbols 20
1.2.2 Signalling diagram conventions 21
1.2.3 Notation for timers and counters 21
2 Profile overview 22
2.1 Profile stack 22
2.2 Configurations and roles 22
2.3 User requirements and scenarios 23
2.4 Profile fundamentals 23
2.5 Conformance 24
3 User interface aspects 25
3.1 The user interface level 25
3.2 Representation of Bluetooth parameters 25
3.2.1 Bluetooth device address (BD_ADDR) 25
3.2.1.1 Definition 25
3.2.1.2 Term on user interface level 25
3.2.1.3 Representation 25
3.2.2 Bluetooth device name (the user-friendly name) 25
3.2.2.1 Definition 25
3.2.2.2 Term on user interface level 26
3.2.2.3 Representation 26
3.2.3 Bluetooth passkey (Bluetooth PIN) 26
3.2.3.1 Definition 26
3.2.3.2 Terms at user interface level 26
3.2.3.3 Representation 26
3.2.4 Class of Device 27
3.2.4.1 Definition 27
3.2.4.2 Term on user interface level 27
3.2.4.3 Representation 27
3.3 Pairing 28

4 Modes 29

4.1 Discoverability modes 29

4.1.1 Non-discoverable mode 30

4.1.1.1 Definition 30

4.1.1.2 Term on UI-level 30

4.1.2 Limited discoverable mode 30

4.1.2.1 Definition 30

4.1.2.2 Conditions 31

4.1.2.3 Term on UI-level 31

4.1.3 General discoverable mode 31

4.1.3.1 Definition 31

4.1.3.2 Conditions 31

4.1.3.3 Term on UI-level 31

4.2 Connectability modes 31

4.2.1 Non-connectable mode 31

4.2.1.1 Definition 31

4.2.1.2 Term on UI-level 32

4.2.2 Connectable mode 32

4.2.2.1 Definition 32

4.2.2.2 Term on UI-level 32

4.3 Pairing modes 32

4.3.1 Non-pairable mode 32

4.3.1.1 Definition 32

4.3.1.2 Term on UI-level 32

4.3.2 Pairable mode 32

4.3.2.1 Definition 32

4.3.2.2 Term on UI-level 32

5 Security aspects 33

5.1 Authentication 33

5.1.1 Purpose 33

5.1.2 Term on UI level 33

5.1.3 Procedure 34

5.1.4 Conditions 34

5.2 Security modes 34

5.2.1 Security mode 1 (non-secure) 36

5.2.2 Security mode 2 (service level enforced security) 36

5.2.3 Security modes 3 (link level enforced security) 36

6 Idle mode procedures 37

6.1 General inquiry 37

6.1.1 Purpose 37

6.1.2 Term on UI level 37

6.1.3 Description 38

6.1.4 Conditions 38

6.2 Limited inquiry 38

6.2.1 Purpose 38

6.2.2 Term on UI level 39

6.2.3 Description 39

6.2.4 Conditions 39

6.3 Name discovery 40

6.3.1 Purpose 40

6.3.2 Term on UI level 40

6.3.3 Description 40

6.3.3.1 Name request 40

6.3.3.2 Name discovery 40

6.3.4 Conditions 41

6.4 Device discovery 41

6.4.1 Purpose 41

6.4.2 Term on UI level 41

6.4.3 Description 42

6.4.4 Conditions 42

6.5 Bonding 42

6.5.1 Purpose 42

6.5.2 Term on UI level 42

6.5.3 Description 43

6.5.3.1 General bonding 43

6.5.3.2 Dedicated bonding 44

6.5.4 Conditions 44

7 Establishment procedures 45
7.1 Link establishment 45
7.1.1 Purpose 45
7.1.2 Term on UI level 45
7.1.3 Description 46
7.1.3.1 B in security mode 1 or 2 46
7.1.3.2 B in security mode 3 47
7.1.4 Conditions 47
7.2 Channel establishment 48
7.2.1 Purpose 48
7.2.2 Term on UI level 48
7.2.3 Description 48
7.2.3.1 B in security mode 2 49
7.2.3.2 B in security mode 1 or 3 49
7.2.4 Conditions 49
7.3 Connection establishment 50
7.3.1 Purpose 50
7.3.2 Term on UI level 50
7.3.3 Description 50
7.3.3.1 B in security mode 2 50
7.3.3.2 B in security mode 1 or 3 51
7.3.4 Conditions 51
7.4 Establishment of additional connection 51
8 Definitions 52
8.1 General definitions 52
8.2 Connection-related definitions 52
8.3 Device-related definitions 53
8.4 Procedure-related definitions 54
8.5 Security-related definitions 54
9 Annex A (Normative): Timers and constants 56
10 Annex B (Informative): Information flows of related procedures 57
10.1 lmp-authentication 57
10.2 lmp-pairing 58
10.3 Service discovery 58
11 References 60

FOREWORD

Interoperability between devices from different manufacturers is provided for a specific service and use case, if the devices conform to a Bluetooth SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications and gives an unambiguous description of the air interface for specified service(s) and use case(s).

All defined features are process-mandatory. This means that, if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

The purpose of the Generic Access Profile is:

To introduce definitions, recommendations and common requirements related to modes and access procedures that are to be used by transport and application profiles.

To describe how devices are to behave in standby and connecting states in order to guarantee that links and channels always can be established between Bluetooth devices, and that multi-profile operation is possible. Special focus is put on discovery, link establishment and security procedures.

To state requirements on user interface aspects, mainly coding schemes and names of procedures and parameters, that are needed to guarantee a satisfactory user experience.

1.2 SYMBOLS AND CONVENTIONS

1.2.1 Requirement status symbols

In this document (especially in the profile requirements tables), the following symbols are used:

‘M’ for mandatory to support (used for capabilities that shall be used in the profile);

’O’ for optional to support (used for capabilities that can be used in the profile);

‘C’ for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

‘X’ for excluded (used for capabilities that may be supported by the unit but shall never be used in the profile);

’N/A’ for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

In this specification, the word *shall* is used for mandatory requirements, the word *should* is used to express recommendations and the word *may* is used for options.

1.2.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures :

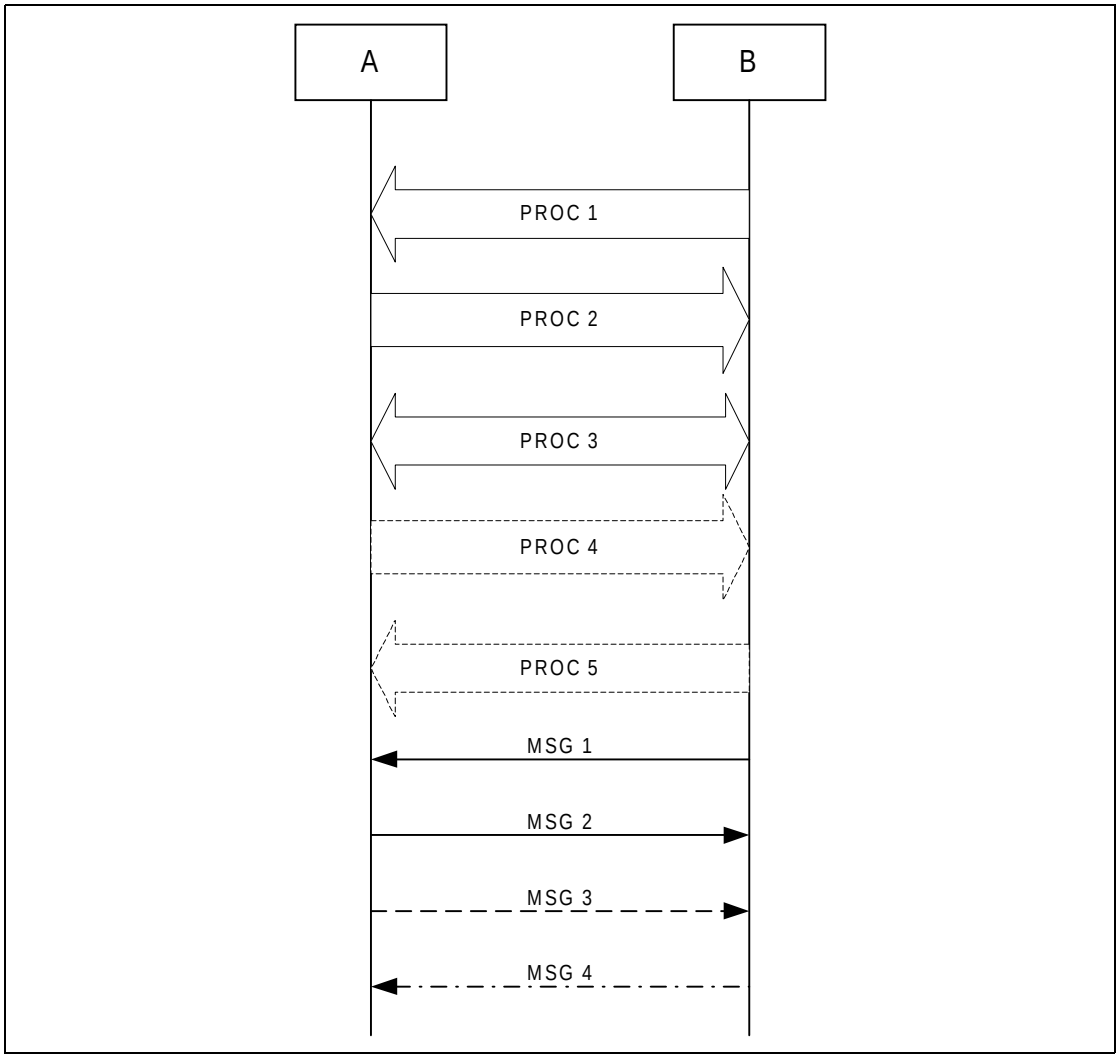


Figure 1.1: Arrows used in signalling diagrams

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A or B). Dashed arrows denote optional steps. PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates a conditional message from B to A.

1.2.3 Notation for timers and counters

Timers are introduced specific to this profile. To distinguish them from timers used in the Bluetooth protocol specifications and other profiles, these timers are named in the following format: 'T_{GAP}*(nnn)*'.

2 PROFILE OVERVIEW

2.1 PROFILE STACK

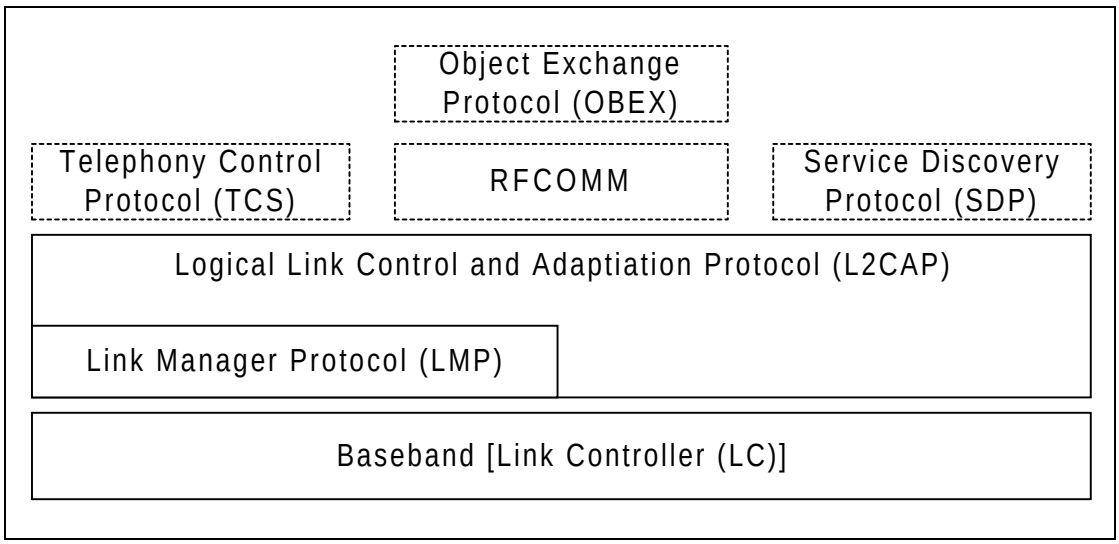


Figure 2.1: Profile stack covered by this profile.

The main purpose of this profile is to describe the use of the lower layers of the Bluetooth protocol stack (LC and LMP). To describe security related alternatives, also higher layers (L2CAP, RFCOMM and OBEX) are included.

2.2 CONFIGURATIONS AND ROLES

For the descriptions in this profile of the roles that the two devices involved in a Bluetooth communication can take, the generic notation of the A-party (the *paging device* in case of link establishment, or *initiator* in case of another procedure on an established link) and the B-party (*paged device* or *acceptor*) is used. The A-party is the one that, for a given procedure, initiates the establishment of the physical link or initiates a transaction on an existing link.

This profile handles the procedures between two devices related to discovery and connecting (link and connection establishment) for the case where none of the two devices has any link established as well as the case where (at least) one device has a link established (possibly to a third device) before starting the described procedure.

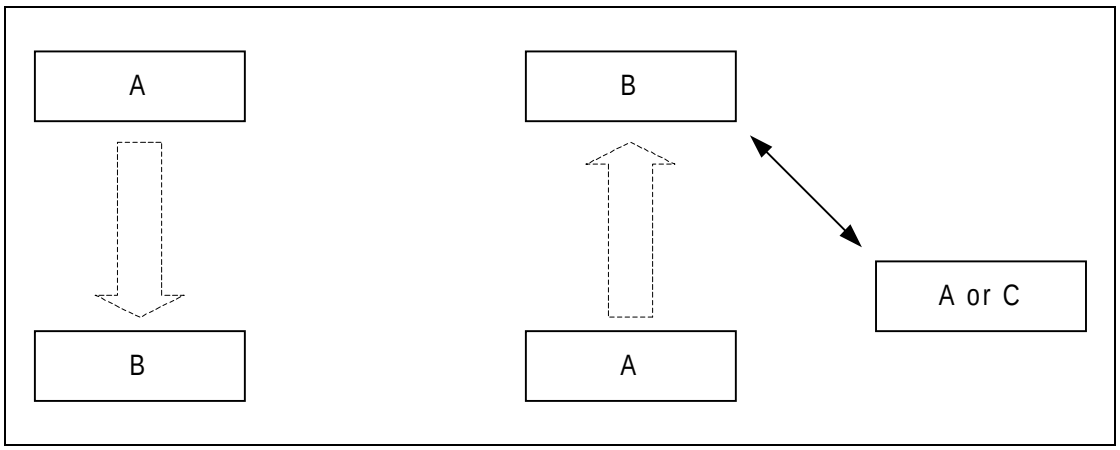


Figure 2.2: This profile covers procedures initiated by one device (A) towards another device (B) that may or may not have an existing Bluetooth link active.

The initiator and the acceptor generally operate the generic procedures according to this profile or another profile referring to this profile. If the acceptor operates according to several profiles simultaneously, this profile describes generic mechanisms for how to handle this.

2.3 USER REQUIREMENTS AND SCENARIOS

The Bluetooth user should in principle be able to connect a Bluetooth device to any other Bluetooth device. Even if the two connected devices don't share any common application, it should be possible for the user to find this out using basic Bluetooth capabilities. When the two devices do share the same application but are from different manufacturers, the ability to connect them should not be blocked just because manufacturers choose to call basic Bluetooth capabilities by different names on the user interface level or implement basic procedures to be executed in different orders.

2.4 PROFILE FUNDAMENTALS

This profile states the requirements on names, values and coding schemes used for names of parameters and procedures experienced on the user interface level.

This profile defines modes of operation that are not service- or profile-specific, but that are generic and can be used by profiles referring to this profile, and by devices implementing multiple profiles.

This profile defines the general procedures that can be used for discovering identities, names and basic capabilities of other Bluetooth devices that are in a mode where they can be discoverable. Only procedures where no channel or connection establishment is used are specified.

This profile defines the general procedure for how to create bonds (i.e. dedicated exchange of link keys) between Bluetooth devices.

This profile describes the general procedures that can be used for establishing connections to other Bluetooth devices that are in mode that allows them to accept connections and service requests.

2.5 CONFORMANCE

Bluetooth devices that do not conform to any other Bluetooth profile shall conform to this profile to ensure basic interoperability and co-existence.

Bluetooth devices that conform to another Bluetooth profile may use adaptations of the generic procedures as specified by that other profile. They shall, however, be compatible with devices compliant to this profile at least on the level of the supported generic procedures.

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 USER INTERFACE ASPECTS

3.1 THE USER INTERFACE LEVEL

In the context of this specification, the user interface level refers to places (such as displays, dialog boxes, manuals, packaging, advertising, etc.) where users of Bluetooth devices encounters names, values and numerical representation of Bluetooth terminology and parameters.

This profile specifies the generic terms that should be used on the user interface level.

3.2 REPRESENTATION OF BLUETOOTH PARAMETERS

3.2.1 Bluetooth device address (BD_ADDR)

3.2.1.1 Definition

BD_ADDR is the unique address of a Bluetooth device as defined in [1]. It is received from a remote device during the device discovery procedure.

3.2.1.2 Term on user interface level

When the Bluetooth address is referred to on UI level, the term ’Bluetooth Device Address’ should be used.

3.2.1.3 Representation

On BB level the BD_ADDR is represented as 48 bits [1].

On the UI level the Bluetooth address shall be represented as 12 hexadecimal characters, possibly divided into sub-parts separated by’:’.
(E.g., ’000C3E3A4B69’ or ’00:0C:3E:3A:4B:69’.) At UI level, any number shall have the MSB -> LSB (from left to right) ’natural’ ordering (e.g., the number ’16’ shall be shown as ’0x10’).

3.2.2 Bluetooth device name (the user-friendly name)

3.2.2.1 Definition

The Bluetooth device name is the user-friendly name that a Bluetooth device presents itself with. It is a character string returned in LMP_name_res as response to a LMP_name_req.

3.2.2.2 Term on user interface level

When the Bluetooth device name is referred to on UI level, the term 'Bluetooth Device Name' should be used.

3.2.2.3 Representation

The Bluetooth device name can be up to 248 bytes maximum according to [2]. It shall be encodedaccording to UTF-8 (i.e. name entered on UI level may be down to 82 characters outside the Unicode range 0x00-0x7F are used).

A device can not expect that a general remote device is able to handle more than the first 40 characters of the Bluetooth device name. If a remote device has limited display capabilities, it may use only the first 20 characters.

3.2.3 Bluetooth passkey (Bluetooth PIN)

3.2.3.1 Definition

The Bluetooth PIN is used to authenticate two Bluetooth devices (that have not previously exchanged link keys) to each other and create a trusted relationship between them. The PIN is used in the pairing procedure (see Section 10.2) to generate the initial link key that is used for further authentication.

The PIN may be entered on UI level but may also be stored in the device; e.g. in the case of a device without sufficient MMI for entering and displaying digits.

3.2.3.2 Terms at user interface level

When the Bluetooth PIN is referred to on UI level, the term 'Bluetooth Passkey' should be used.

3.2.3.3 Representation

The Bluetooth PIN has different representations on different levels. PINBB is used on baseband level, and PINUI is used on user interface level.

PINBB is the PIN used by [1] for calculating the initialization key during the pairing procedure. PINUI is the character representation of the PIN that is entered on UI level. The transformation from PINUI to PINBB shall be according to UTF-8 and decimal digits shall be within the Unicode range 0x00 - 0x7F.

According to [1], PINBB can be 128 bits (16 bytes). I.e. if a device supports entry of characters outside the Unicode range 0x00 - 0x7F, the maximum number of characters in the PINUI may be less than 16.

Examples:

User-entered code	Corresponding PIN_{BB}[0..length-1] (value as a sequence of octets in hexadecimal notation)
’0123’	length = 4, value = 0x30 0x31 0x32 0x33
’Ärlich’	length = 7, value = 0xC3 0x84 0x72 0x6C 0x69 0x63 0x68

All Bluetooth devices that support the bonding procedure and support PIN handling on UI level shall support UI level handling of PINs consisting of decimal digits. In addition, devices may support UI level handling of PINs consisting of general characters.

If a device has a fixed PIN (i.e. PIN is stored in the device and cannot be entered on UI level during pairing), the PIN shall be defined using decimal digits. A device that is expected to pair with a remote device that has restricted UI capabilities should ensure that the PIN can be entered on UI level as decimal digits.

3.2.4 Class of Device

3.2.4.1 Definition

Class of device is a parameter received during the device discovery procedure, indicating the type of device and which types of service that are supported.

3.2.4.2 Term on user interface level

The information within the Class of Device parameter should be referred to as ’Bluetooth Device Class’ (i.e. the major and minor device class fields) and ’Bluetooth Service Type’ (i.e. the service class field). The terms for the defined Bluetooth Device Types and Bluetooth Service Types are defined in [11].

When using a mix of information found in the Bluetooth Device Class and the Bluetooth Service Type, the term ’Bluetooth Device Type’ should be used.

3.2.4.3 Representation

The Class of device is a bit field and is defined in [11]. The UI-level representation of the information in the Class of device is implementation specific.

3.3 PAIRING

Two procedures are defined that make use of the pairing procedure defined on LMP level (LMP-pairing, see Section 10.2). Either the user initiates the bonding procedure and enters the passkey with the explicit purpose of creating a bond (and maybe also a secure relationship) between two Bluetooth devices, or the user is requested to enter the passkey during the establishment procedure since the devices did not share a common link key beforehand. In the first case, the user is said to perform ’bonding (with entering of passkey)’ and in the second case the user is said to ’authenticate using the passkey’.

4 MODES

	Procedure	Ref.	Support
1	Discoverability modes	4.1	
	Non-discoverable mode		C1
	Limited discoverable mode		O
	General discoverable mode		O
2	Connectability modes	4.1.3.3	
	Non-connectable mode		O
	Connectable mode		M
3	Pairing modes	4.2.2.2	
	Non-pairable mode		O
	Pairable mode		C2
C1: If limited discoverable mode is supported, non-discoverable mode is mandatory, otherwise optional.			
C2: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise optional.			

Table 4.1: Conformance requirements related to modes defined in this section

4.1 DISCOVERABILITY MODES

With respect to inquiry, a Bluetooth device shall be either in non-discoverable mode or in a discoverable mode. (The device shall be in one, and only one, discoverability mode at a time.) The two discoverable modes defined here are called limited discoverable mode and general discoverable mode. Inquiry is defined in [1].

When a Bluetooth device is in non-discoverable mode it does not respond to inquiry.

A Bluetooth device is said to be made discoverable, or set into a discoverable mode, when it is in limited discoverable mode or in general discoverable mode. Even when a Bluetooth device is made discoverable it may be unable to respond to inquiry due to other baseband activity [1]. A Bluetooth device that does not respond to inquiry for any of these two reasons is called a silent device.

After being made discoverable, the Bluetooth device shall be discoverable for at least $T_{GAP}(103)$.

4.1.1 Non-discoverable mode

4.1.1.1 Definition

When a Bluetooth device is in non-discoverable mode, it shall never enter the INQUIRY_RESPONSE state.

4.1.1.2 Term on UI-level

Bluetooth device is ’non-discoverable’ or in ’non-discoverable mode’.

4.1.2 Limited discoverable mode

4.1.2.1 Definition

The limited discoverable mode should be used by devices that need to be discoverable only for a limited period of time, during temporary conditions or for a specific event. The purpose is to respond to a device that makes a limited inquiry (inquiry using the LIAC).

A Bluetooth device should not be in limited discoverable mode for more than $T_{GAP}(104)$. The scanning for the limited inquiry access code can be done either in parallel or in sequence with the scanning of the general inquiry access code. When in limited discoverable mode, one of the following options shall be used.

4.1.2.1.1 Parallel scanning

When a Bluetooth device is in limited discoverable mode, it shall enter the INQUIRY_SCAN state at least once in $T_{GAP}(102)$ and scan for the GIAC and the LIAC for at least $T_{GAP}(101)$.

4.1.2.1.2 Sequential scanning

When a Bluetooth device is in limited discoverable mode, it shall enter the INQUIRY_SCAN state at least once in $T_{GAP}(102)$ and scan for the GIAC for at least $T_{GAP}(101)$ and enter the INQUIRY_SCAN state more often than once in $T_{GAP}(102)$ and scan for the LIAC for at least $T_{GAP}(101)$.

If an inquiry message is received when in limited discoverable mode, the entry into the INQUIRY_RESPONSE state takes precedence over the next entries into INQUIRY_SCAN state until the inquiry response is completed.

4.1.2.2 Conditions

When a device is in limited discoverable mode it shall set bit no 13 in the Major Service Class part of the Class of Device/Service field [11].

4.1.2.3 Term on UI-level

Bluetooth device is ’discoverable’ or in ’discoverable mode’.

4.1.3 General discoverable mode

4.1.3.1 Definition

The general discoverable mode shall be used by devices that need to be discoverable continuously or for no specific condition. The purpose is to respond to a device that makes a general inquiry (inquiry using the GIAC).

4.1.3.2 Conditions

When a Bluetooth device is in general discoverable mode, it shall enter the INQUIRY_SCAN state more often than once in $T_{GAP}(102)$ and scan for the GIAC for at least $T_{GAP}(101)$.

A device in general discoverable mode shall not respond to a LIAC inquiry.

4.1.3.3 Term on UI-level

Bluetooth device is ’discoverable’ or in ’discoverable mode’.

4.2 CONNECTABILITY MODES

With respect to paging, a Bluetooth device shall be either in non-connectable mode or in connectable mode. Paging is defined in [1].

When a Bluetooth device is in non-connectable mode it does not respond to paging. When a Bluetooth device is in connectable mode it responds to paging.

4.2.1 Non-connectable mode

4.2.1.1 Definition

When a Bluetooth device is in non-connectable mode it shall never enter the PAGE_SCAN state.

4.2.1.2 Term on UI-level

Bluetooth device is ’non-connectable’ or in ’non-connectable mode’.

4.2.2 Connectable mode

4.2.2.1 Definition

When a Bluetooth device is in connectable mode it shall periodically enter the PAGE_SCAN state.

4.2.2.2 Term on UI-level

Bluetooth device is ’connectable’ or in ’connectable mode’.

4.3 PAIRING MODES

With respect to pairing, a Bluetooth device shall be either in non-pairable mode or in pairable mode. In pairable mode the Bluetooth device accepts paring – i.e. creation of bonds – initiated by the remote device, and in non-pairable mode it does not. Pairing is defined in [1] and [2].

4.3.1 Non-pairable mode

4.3.1.1 Definition

When a Bluetooth device is in non-pairable mode it shall respond to a received LMP_in_rand with LMP_not_accepted with the reason *pairing not allowed*.

4.3.1.2 Term on UI-level

Bluetooth device is ’non-bondable’ or in ’non-bondable mode’ or “does not accept bonding”.

4.3.2 Pairable mode

4.3.2.1 Definition

When a Bluetooth device is in pairable mode it shall respond to a received LMP_in_rand with LMP_accepted (or with LMP_in_rand if it has a fixed PIN).

4.3.2.2 Term on UI-level

Bluetooth device is ’bondable’ or in ’bondable mode’ or “accepts bonding”.

5 SECURITY ASPECTS

	Procedure	Ref.	Support
1	Authentication	5.1	C1
2	Security modes	5.2	
	Security mode 1		O
	Security mode 2		C2
	Security mode 3		C2
C1: If security mode 1 is the only security mode that is supported, support for authentication is optional, otherwise mandatory. (Note: support for LMP-authentication and LMP-pairing is mandatory according [2] independent of which security mode that is used.)			
C2: If security mode 1 is not the only security mode that is supported, then support for at least one of security mode 2 or security mode 3 is mandatory.			

Table 5.1: Conformance requirements related to the generic authentication procedure and the security modes defined in this section

5.1 AUTHENTICATION

5.1.1 Purpose

The generic authentication procedure describes how the LMP-authentication and LMP-pairing procedures are used when authentication is initiated by one Bluetooth device towards another, depending on if a link key exists or not and if pairing is allowed or not.

5.1.2 Term on UI level

'Bluetooth authentication'.

5.1.3 Procedure

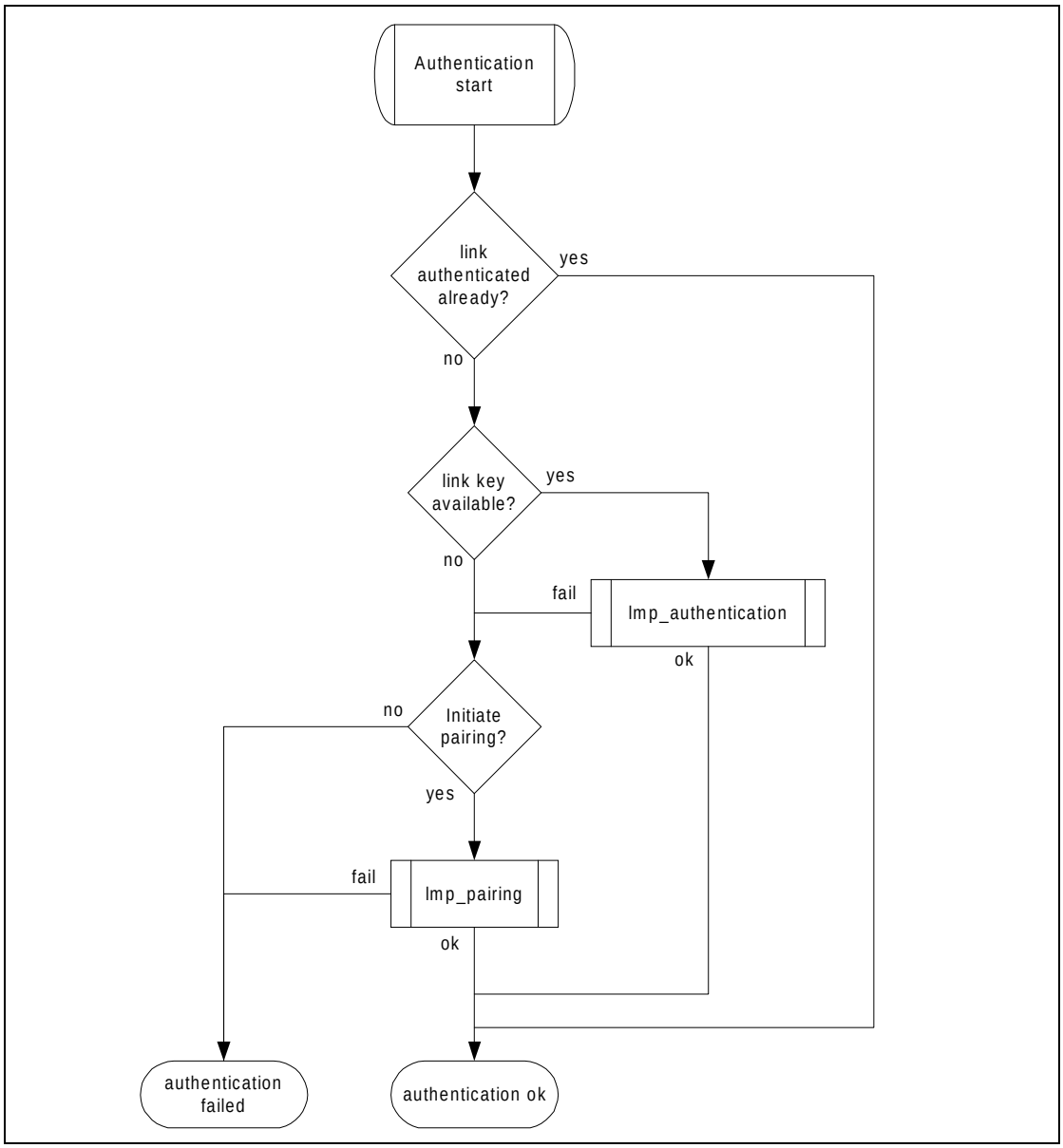


Figure 5.1: Definition of the generic authentication procedure.

5.1.4 Conditions

The device that initiates authentication has to be in security mode 2 or in security mode 3.

5.2 SECURITY MODES

The following flow chart describes where in the channel establishment procedures initiation of authentication takes place, depending on which security mode the Bluetooth device is in.

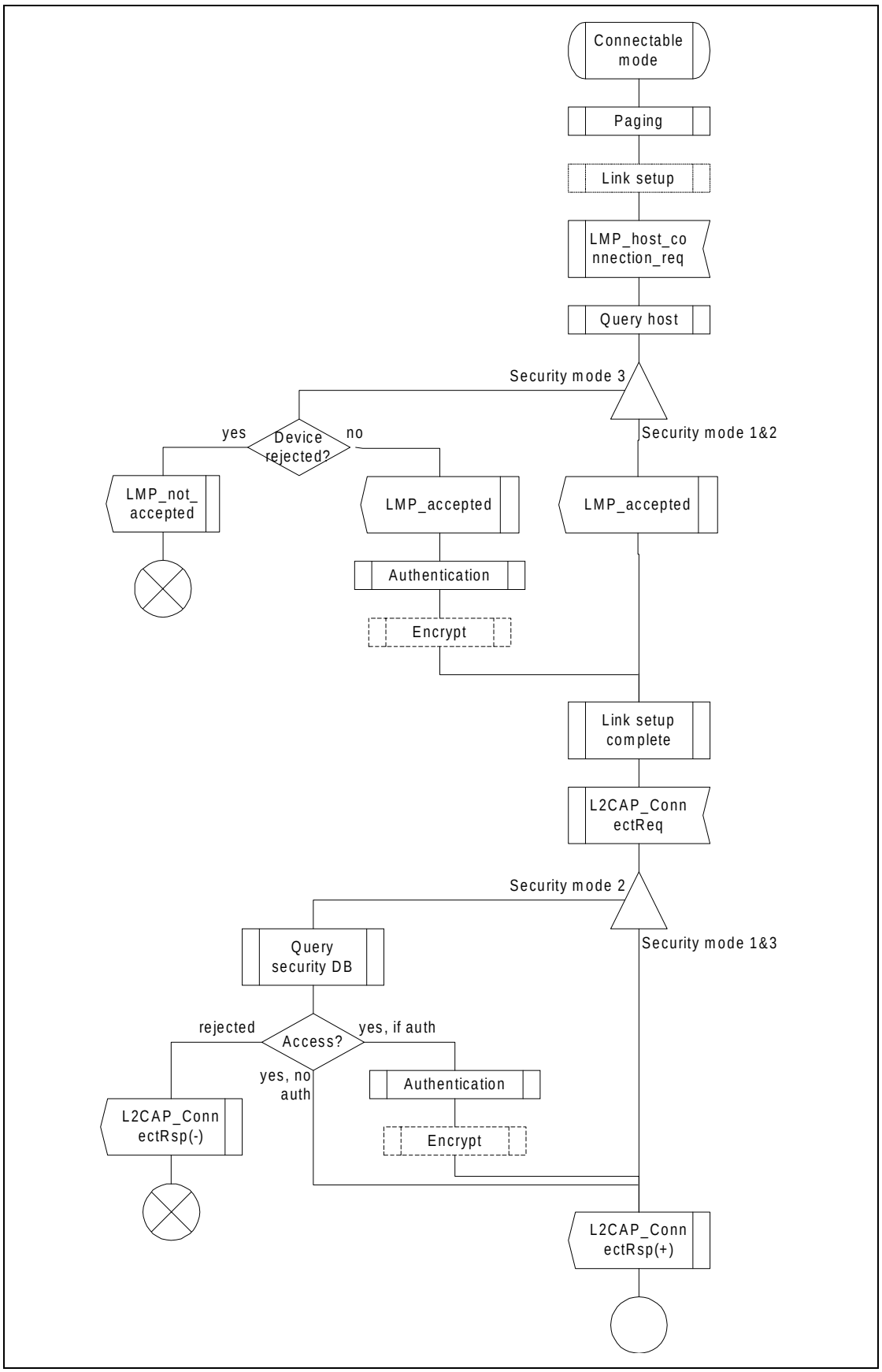


Figure 5.2: Illustration of channel establishment using different security modes.

When authentication is initiated towards a Bluetooth device, it shall act according to [2] and the current pairing mode, independent of which security mode it is in.

5.2.1 Security mode 1 (non-secure)

When a Bluetooth device is in security mode 1 it shall never initiate any security procedure (i.e., it shall never send LMP_au_rand, LMP_in_rand or LMP_encryption_mode_req).

5.2.2 Security mode 2 (service level enforced security)

When a Bluetooth device is in security mode 2 it shall not initiate any security procedure before a channel establishment request (L2CAP_ConnectReq) has been received or a channel establishment procedure has been initiated by itself. (The behavior of a device in security mode 2 is further described in [10].) Whether a security procedure is initiated or not depends on the security requirements of the requested channel or service.

A Bluetooth device in security mode 2 should classify the security requirements of its services using at least the following attributes:

- Authorization required;
- Authentication required;
- Encryption required.

Note: Security mode 1 can be considered (at least from a remote device point of view) as a special case of security mode 2 where no service has registered any security requirements.

5.2.3 Security modes 3 (link level enforced security)

When a Bluetooth device is in security mode 3 it shall initiate security procedures before it sends LMP_link_setup_complete. (The behavior of a device in security mode 3 is as described in [2].)

A Bluetooth device in security mode 3 may reject the host connection request (respond with LMP_not_accepted to the LMP_host_connection_req) based on settings in the host (e.g. only communication with pre-paired devices allowed).

6 IDLE MODE PROCEDURES

The inquiry and discovery procedures described here are applicable only to the device that initiates them (A). The requirements on the behavior of B is according to the modes specified in Section 4 and to [2].

	Procedure	Ref.	Support
1	General inquiry	6.1	C1
2	Limited inquiry	6.2	C1
3	Name discovery	6.3	O
4	Device discovery	6.4	O
5	Bonding	6.5	O
C1: If initiation of bonding is supported, support for at least one inquiry procedure is mandatory, otherwise optional. (Note: support for LMP-pairing is mandatory [2].)			

6.1 GENERAL INQUIRY

6.1.1 Purpose

The purpose of the general inquiry procedure is to provide the initiator with the Bluetooth device address, clock, Class of Device and used page scan mode of general discoverable devices (i.e. devices that are in range with regard to the initiator and are set to scan for inquiry messages with the General Inquiry Access Code). Also devices in limited discoverable mode will be discovered using general inquiry.

The general inquiry should be used by devices that need to discover devices that are made discoverable continuously or for no specific condition.

6.1.2 Term on UI level

'Bluetooth Device Inquiry'.

6.1.3 Description

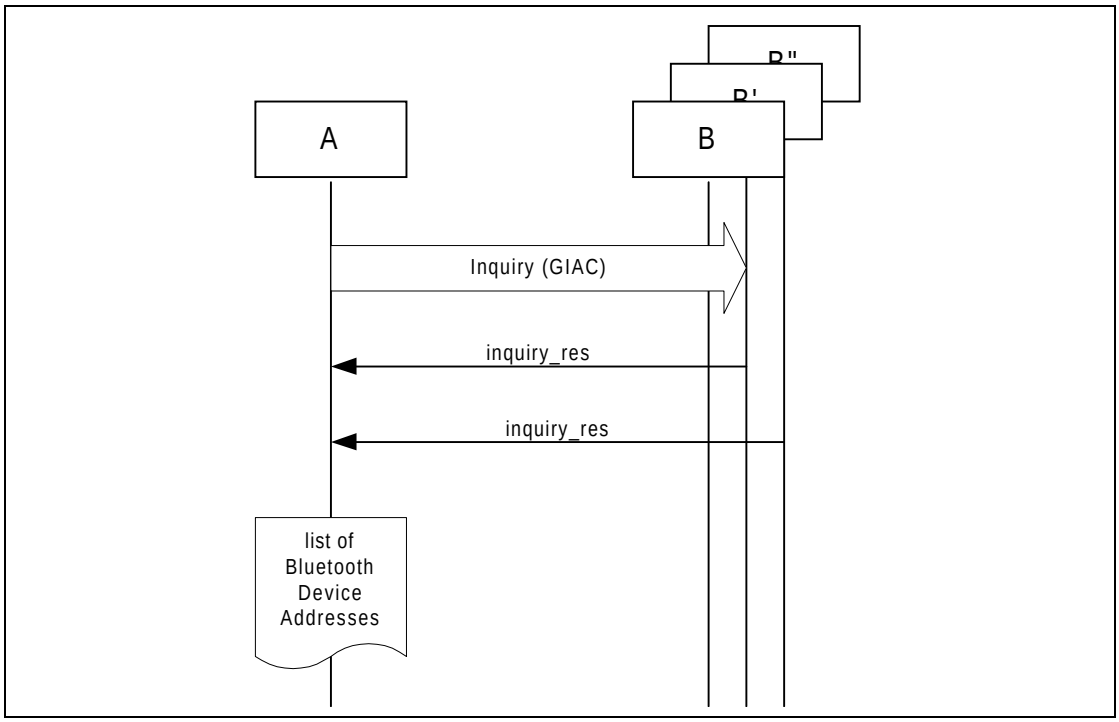


Figure 6.1: General inquiry ,where B is a device in non-discoverable mode, B´ is a device in limited discoverable mode and B" is a device in general discoverable mode. (Note that all discoverable devices are discovered using general inquiry, independent of which discoverable mode they are in.)

6.1.4 Conditions

When general inquiry is initiated by a Bluetooth device, it shall be in the INQUIRY state for at least $T_{GAP}(100)$ and perform inquiry using the GIAC.

In order to receive inquiry response, the remote devices in range have to be made discoverable (limited or general).

6.2 LIMITED INQUIRY

6.2.1 Purpose

The purpose of the limited inquiry procedure is to provide the initiator with the Bluetooth device address, clock, Class of Device and used page scan mode of limited discoverable devices. The latter devices are devices that are in range with regard to the initiator, and may be set to scan for inquiry messages with the Limited Inquiry Access Code, in addition to scanning for inquiry messages with the General Inquiry Access Code.

The limited inquiry should be used by devices that need to discover devices that are made discoverable only for a limited period of time, during temporary conditions or for a specific event. Since it is not guaranteed that the

discoverable device scans for the LIAC, the initiating device may choose any inquiry procedure (general or limited). Even if the remote device that is to be discovered is expected to be made limited discoverable (e.g. when a dedicated bonding is to be performed), the limited inquiry should be done in sequence with a general inquiry in such a way that both inquiries are completed within the time the remote device is limited discoverable, i.e. at least $T_{GAP}(103)$.

6.2.2 Term on UI level

'Bluetooth Device Inquiry'.

6.2.3 Description

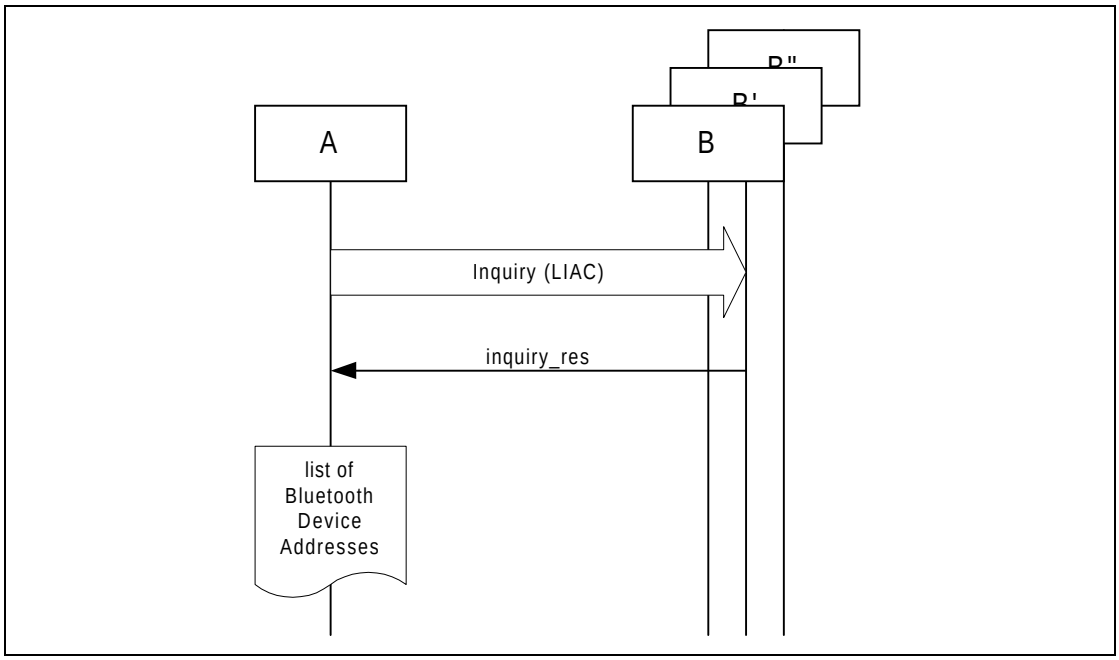


Figure 6.2: Limited inquiry where B is a device in non-discoverable mode, B' is a device in limited discoverable mode and B" is a device in general discoverable mode. (Note that only limited discoverable devices can be discovered using limited inquiry.)

6.2.4 Conditions

When limited inquiry is initiated by a Bluetooth device, it shall be in the INQUIRY state for at least $T_{GAP}(100)$ and perform inquiry using the LIAC.

In order to receive inquiry response, the remote devices in range has to be made limited discoverable.

6.3 NAME DISCOVERY

6.3.1 Purpose

The purpose of name discovery is to provide the initiator with the Bluetooth Device Name of connectable devices (i.e. devices in range that will respond to paging).

6.3.2 Term on UI level

'Bluetooth Device Name Discovery'.

6.3.3 Description

6.3.3.1 Name request

Name request is the procedure for retrieving the Bluetooth Device Name from a connectable Bluetooth device. It is not necessary to perform the full link establishment procedure (see Section 7.1) in order to just to get the name of another device.

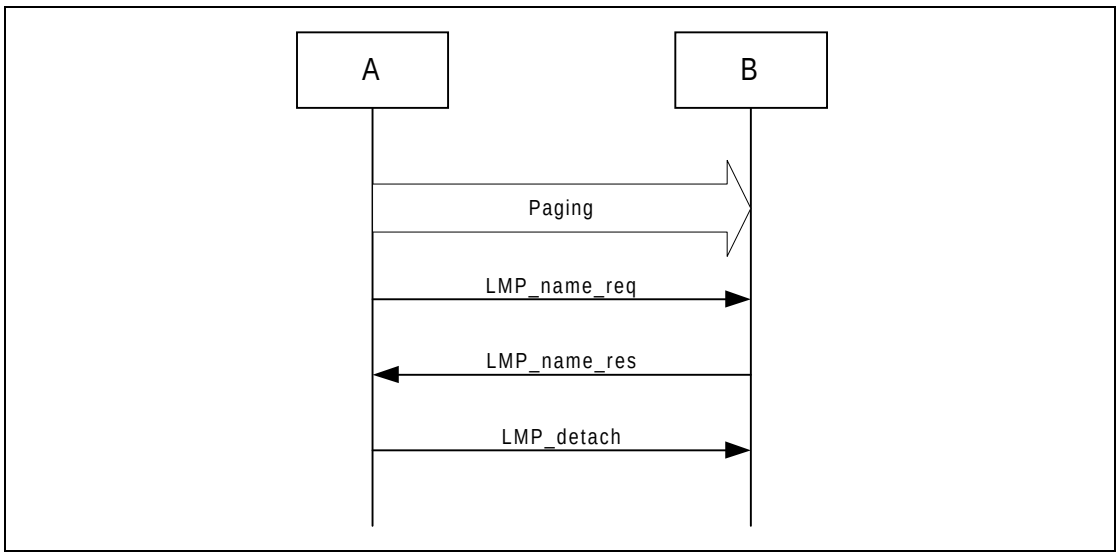


Figure 6.3: Name request procedure.

6.3.3.2 Name discovery

Name discovery is the procedure for retrieving the Bluetooth Device Name from connectable Bluetooth devices by performing name request towards known devices (i.e. Bluetooth devices for which the Bluetooth Device Addresses are available).

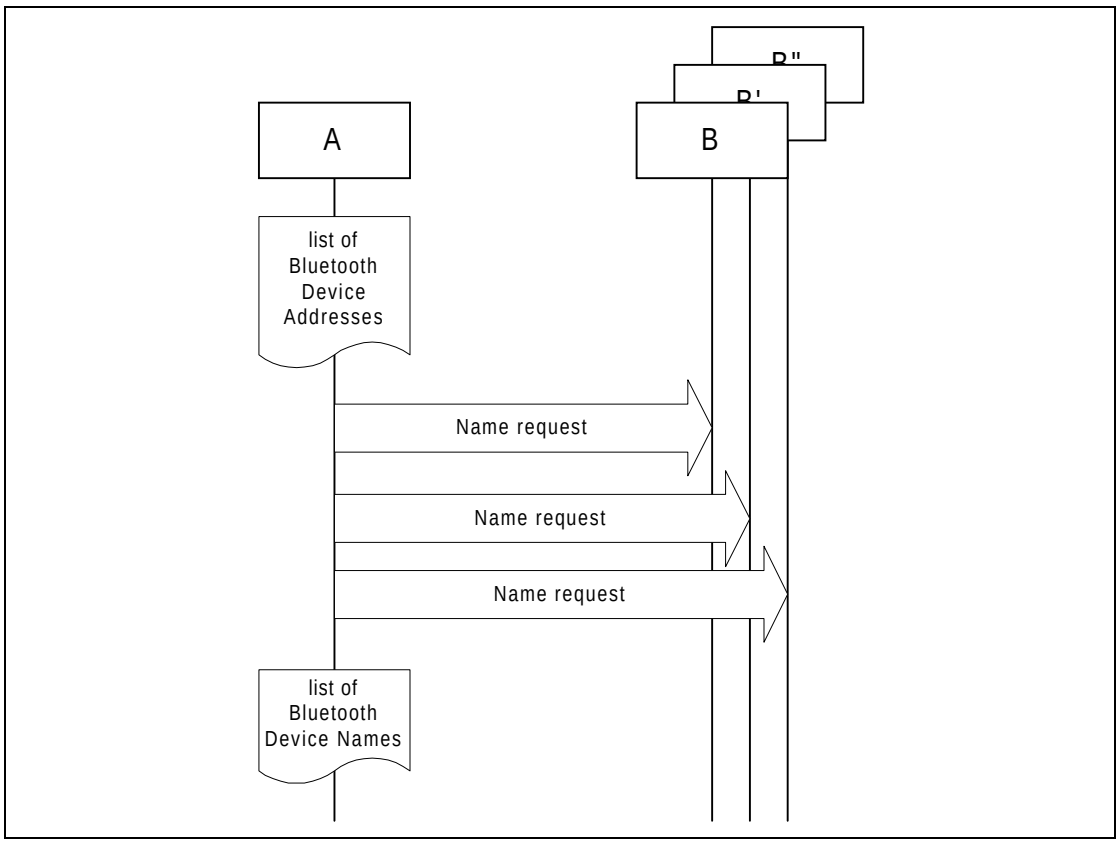


Figure 6.4: Name discovery procedure.

6.3.4 Conditions

In the name request procedure, the initiator will use the Device Access Code of the remote device as retrieved immediately beforehand – normally through an inquiry procedure.

6.4 DEVICE DISCOVERY

6.4.1 Purpose

The purpose of device discovery is to provide the initiator with the Bluetooth Address, clock, Class of Device, used page scan mode and Bluetooth device name of discoverable devices.

6.4.2 Term on UI level

'Bluetooth Device Discovery'.

6.4.3 Description

During the device discovery procedure, first an inquiry (either general or limited) is performed, and then name discovery is done towards some or all of the devices that responded to the inquiry.

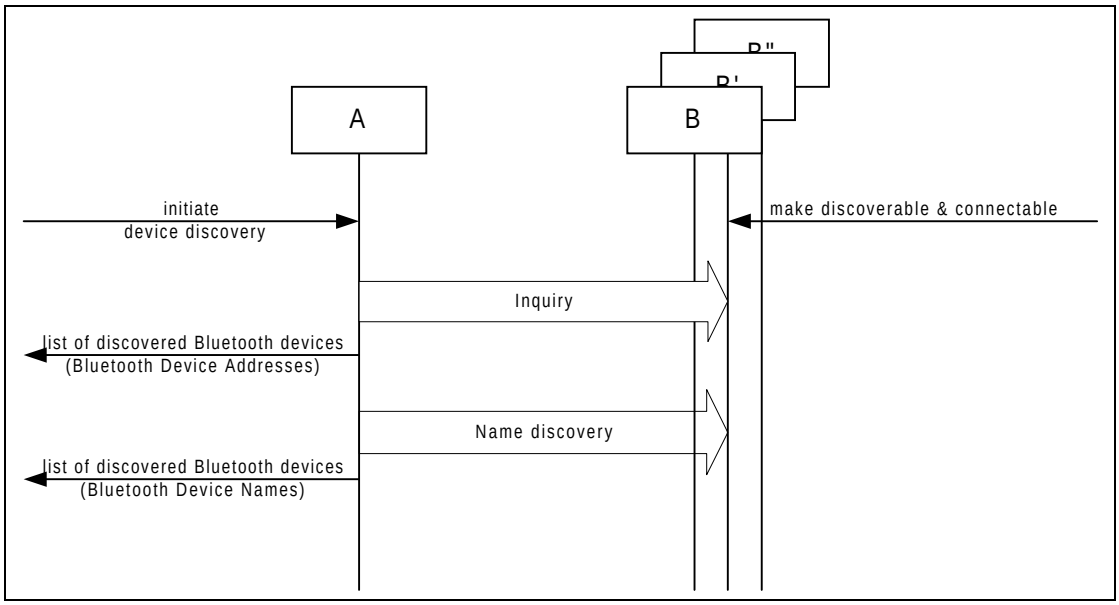


Figure 6.5: Device discovery procedure.

6.4.4 Conditions

Conditions for both inquiry (general or limited) and name discovery must be fulfilled (i.e. devices discovered during device discovery must be both discoverable and connectable).

6.5 BONDING

6.5.1 Purpose

The purpose of bonding is to create a relation between two Bluetooth devices based on a common link key (a bond). The link key is created and exchanged (pairing) during the bonding procedure and is expected to be stored by both Bluetooth devices, to be used for future authentication.

In addition to pairing, the bonding procedure can involve higher layer initialization procedures.

6.5.2 Term on UI level

'Bluetooth Bonding'

6.5.3 Description

Two aspects of the bonding procedure are described here. Dedicated bonding is what is done when the two devices are explicitly set to perform only a creation and exchange of a common link key.

General bonding is included to indicate that the framework for the dedicated bonding procedure is the same as found in the normal channel and connection establishment procedures. This means that pairing may be performed successfully if A has initiated bonding while B is in its normal connectable and security modes.

The main difference with bonding, as compared to a pairing done during link or channel establishment, is that for bonding it is the paging device (A) that must initiate the authentication.

6.5.3.1 General bonding

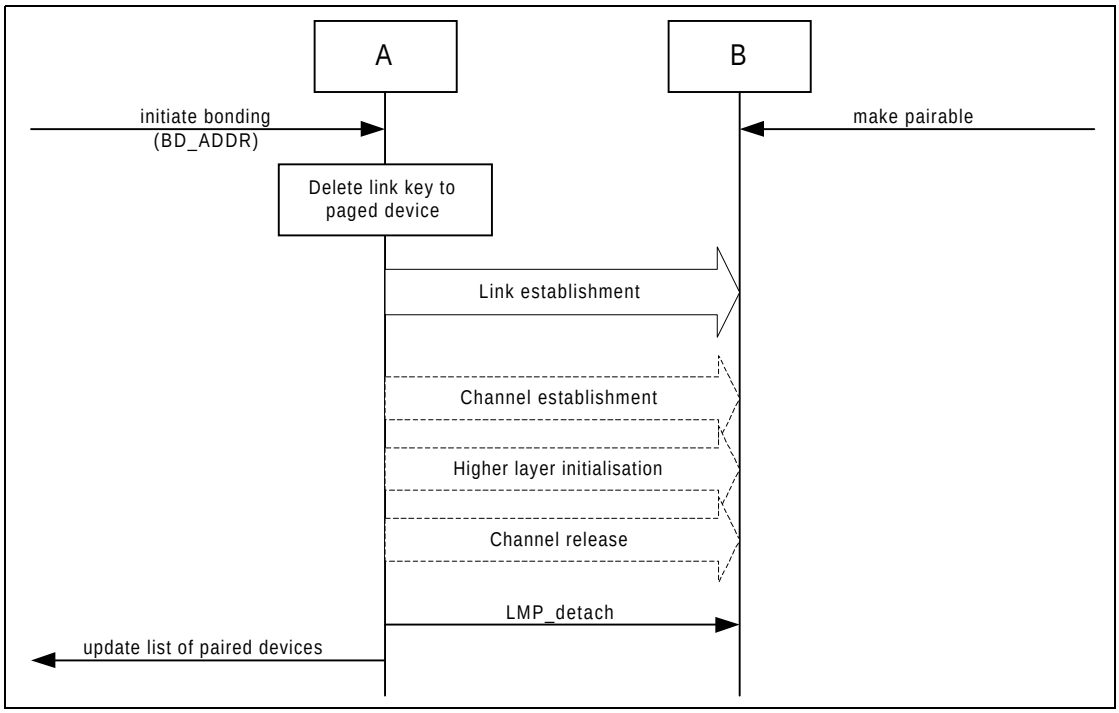


Figure 6.6: General description of bonding as being the link establishment procedure executed under specific conditions on both devices, followed by an optional higher layer initalization process.

6.5.3.2 Dedicated bonding

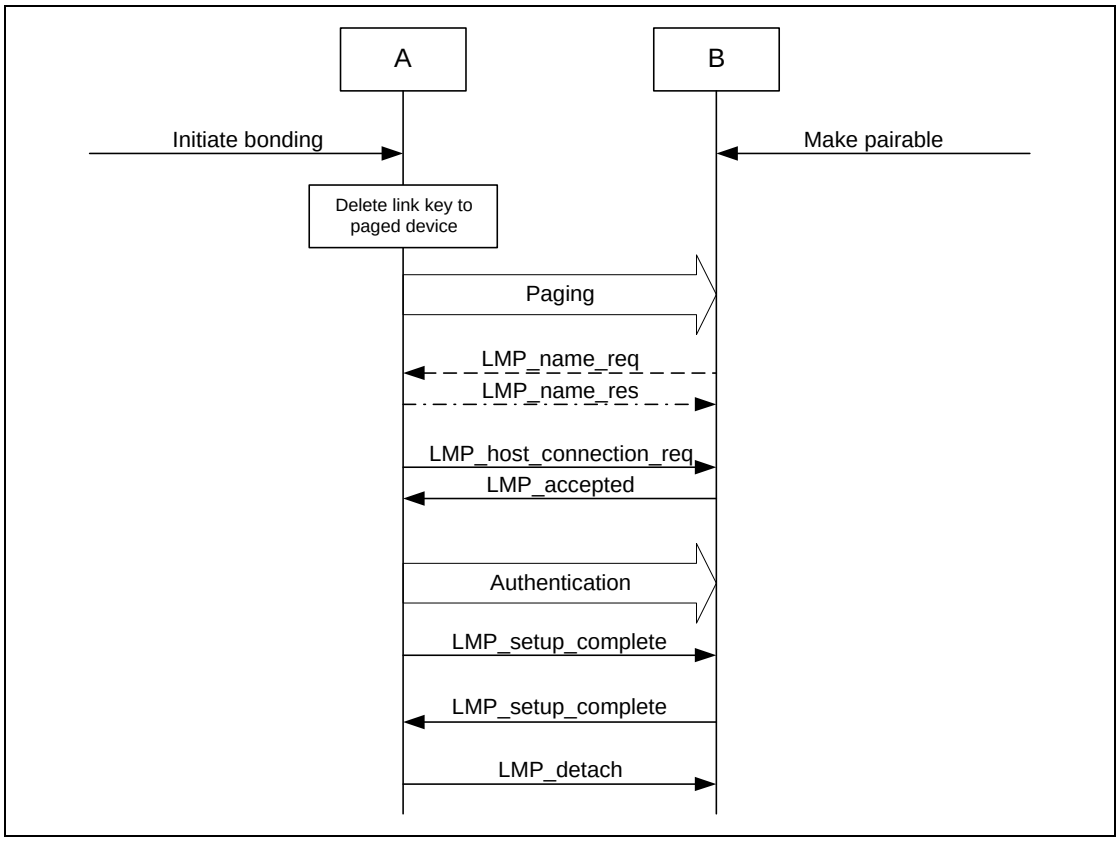


Figure 6.7: Bonding as performed when the purpose of the procedure is only to create and exchange a link key between two Bluetooth devices.

6.5.4 Conditions

Before bonding can be initiated, the initiating device (A) must know the Device Access Code of the device to pair with. This is normally done by first performing device discovery. A Bluetooth Device that can initiate bonding (A) should use limited inquiry, and a Bluetooth Device that accepts bonding (B) should support the limited discoverable mode.

Bonding is in principle the same as link establishment with the conditions:

- The paged device (B) shall be set into pairable mode. The paging device (A) is assumed to allow pairing since it has initiated the bonding procedure.
- The paging device (the initiator of the bonding procedure, A) shall initiate authentication.
- Before initiating the authentication part of the bonding procedure, the paging device should delete any link key corresponding to a previous bonding with the paged device.

7 ESTABLISHMENT PROCEDURES

	Procedure	Ref.	Support in A	Support in B
1	Link establishment	7.1	M	M
2	Channel establishment	7.2	O	M
3	Connection establishment	7.3	O	O

Table 7.1: Establishment procedures

The establishment procedures defined here do not include any discovery part. Before establishment procedures are initiated, the information provided during device discovery (in the FHS packet of the inquiry response or in the response to a name request) has to be available in the initiating device. This information is:

- The Bluetooth Device Address (BD_ADDR) from which the Device Access Code is generated;
- The system clock of the remote device;
- The page scan mode used by the remote device.

Additional information provided during device discovery that is useful for making the decision to initiate an establishment procedure is:

- The Class of device;
- The Device name.

7.1 LINK ESTABLISHMENT

7.1.1 Purpose

The purpose of the link establishment procedure is to establish a physical link (of ACL type) between two Bluetooth devices using procedures from [1] and [2].

7.1.2 Term on UI level

'Bluetooth link establishment'

7.1.3 Description

In this sub-section, the paging device (A) is in security mode 3. The paging device cannot during link establishment distinguish if the paged device (B) is in security mode 1 or 2.

7.1.3.1 B in security mode 1 or 2

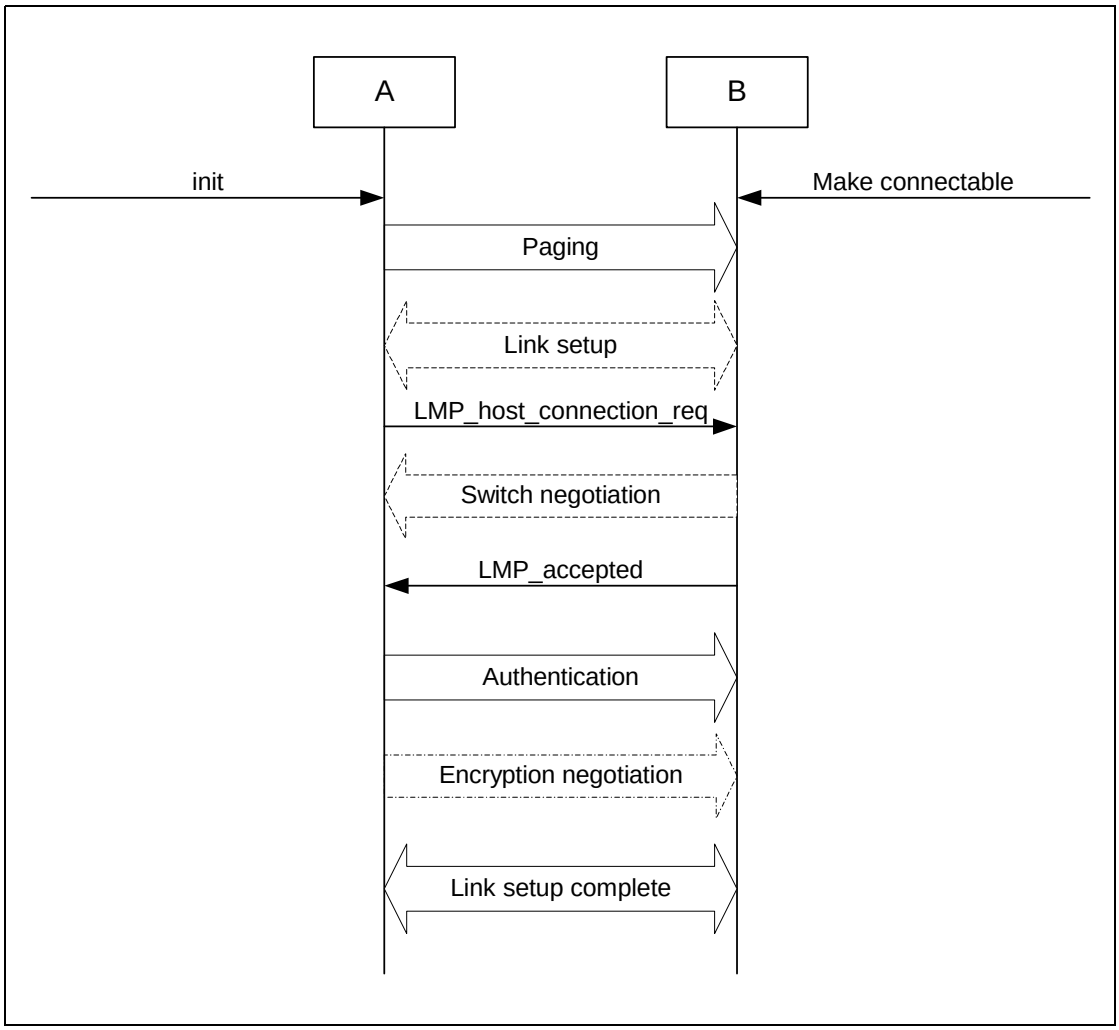


Figure 7.1: Link establishment procedure when the paging device (A) is in security mode 3 and the paged device (B) is in security mode 1 or 2.

7.1.3.2 B in security mode 3

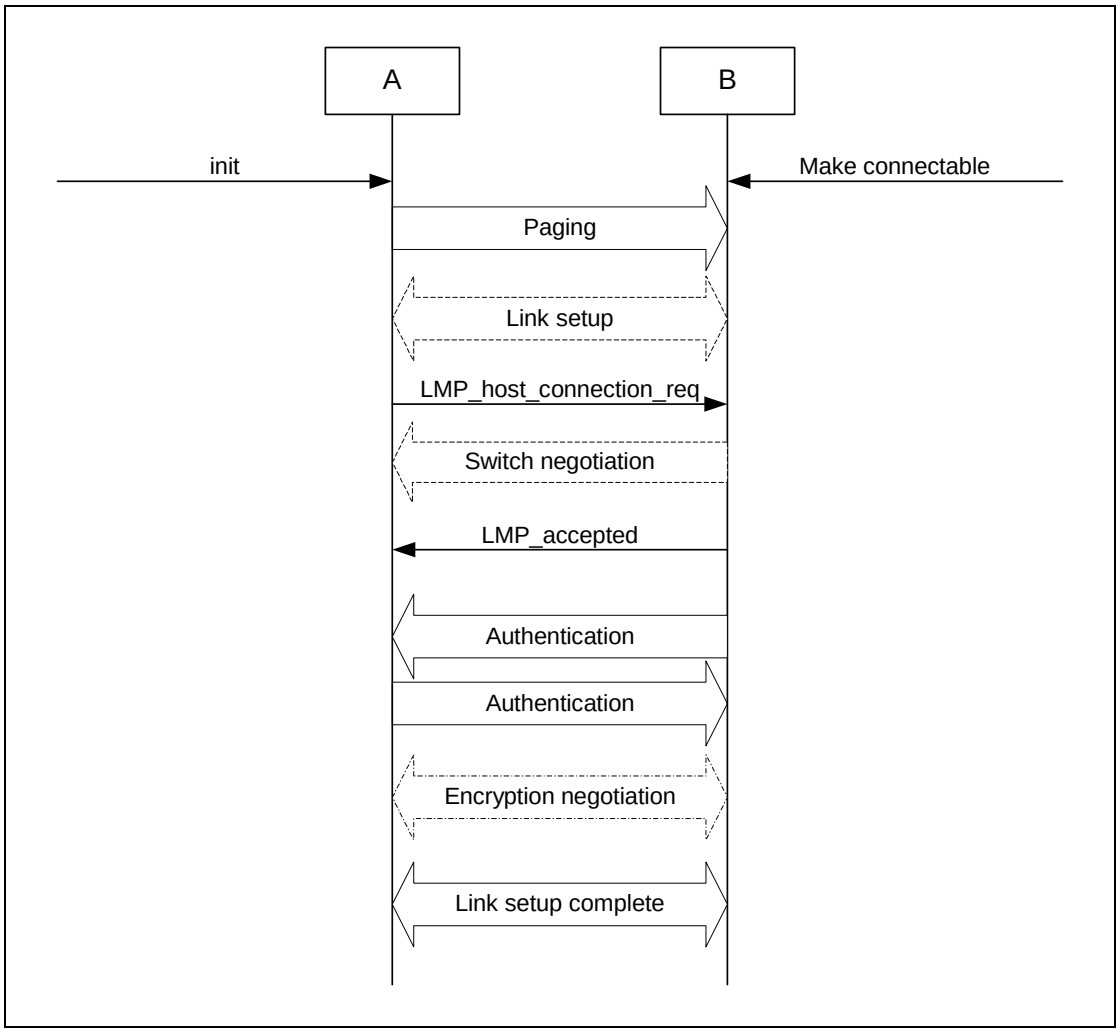


Figure 7.2: Link establishment procedure when both the paging device (A) and the paged device (B) are in security mode 3.

7.1.4 Conditions

The paging procedure shall be according to [1] and the paging device should use the Device access code and page mode received through a previous inquiry. When paging is completed, a physical link between the two Bluetooth devices is established.

If role switching is needed (normally it is the paged device that has an interest in changing the master/slave roles) it should be done as early as possible after the physical link is established. If the paging device does not accept the switch, the paged device has to consider whether to keep the physical link or not.

Both devices may perform link setup (using LMP procedures that require no interaction with the host on the remote side). Optional LMP features can be used after having confirmed (using LMP_feature_req) that the other device supports the feature.

When the paging device needs to go beyond the link setup phase, it issues a request to be connected to the host of the remote device. If the paged device is in security mode 3, this is the trigger for initiating authentication.

The paging device shall send LMP_host_connection_req during link establishment (i.e. before channel establishment) and may initiate authentication only after having sent LMP_host_connection_request.

After an authentication has been performed, any of the devices can initiate encryption.

Further link configuration may take place after the LMP_host_connection_req. When both devices are satisfied, they send LMP_setup_complete.

Link establishment is completed when both devices have sent LMP_setup_complete.

7.2 CHANNEL ESTABLISHMENT

7.2.1 Purpose

The purpose of the channel establishment procedure is to establish a Bluetooth channel (a logical link) between two Bluetooth devices using [3].

7.2.2 Term on UI level

'Bluetooth channel establishment'.

7.2.3 Description

In this sub-section, the initiator (A) is in security mode 3. During channel establishment, the initiator cannot distinguish if the acceptor (B) is in security mode 1 or 3.

7.2.3.1 B in security mode 2

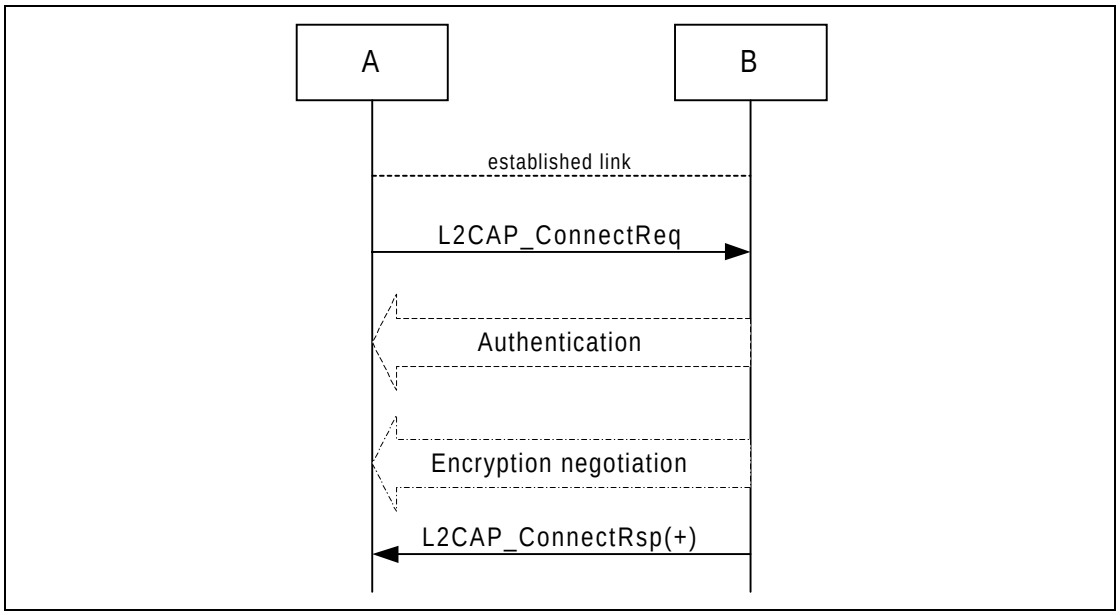


Figure 7.3: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

7.2.3.2 B in security mode 1 or 3

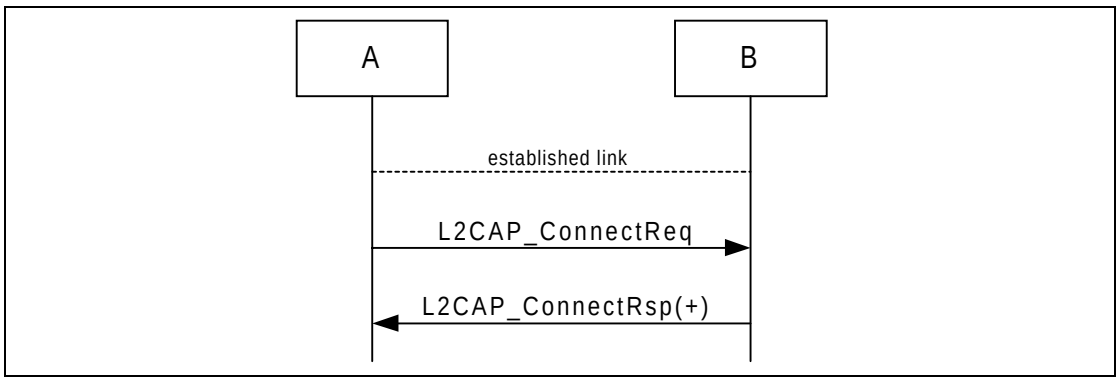


Figure 7.4: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

7.2.4 Conditions

Channel establishment starts after link establishment is completed when the initiator sends a channel establishment request (L2CAP_ConnectReq).

Depending on security mode, security procedures may take place after the channel establishment has been initiated.

Channel establishment is completed when the acceptor responds to the channel establishment request (with a positive L2CAP_ConnectRsp).

7.3 CONNECTION ESTABLISHMENT

7.3.1 Purpose

The purpose of the connection establishment procedure is to establish a connection between applications on two Bluetooth devices.

7.3.2 Term on UI level

'Bluetooth connection establishment'

7.3.3 Description

In this sub-section, the initiator (A) is in security mode 3. During connection establishment, the initiator cannot distinguish if the acceptor (B) is in security mode 1 or 3.

7.3.3.1 B in security mode 2

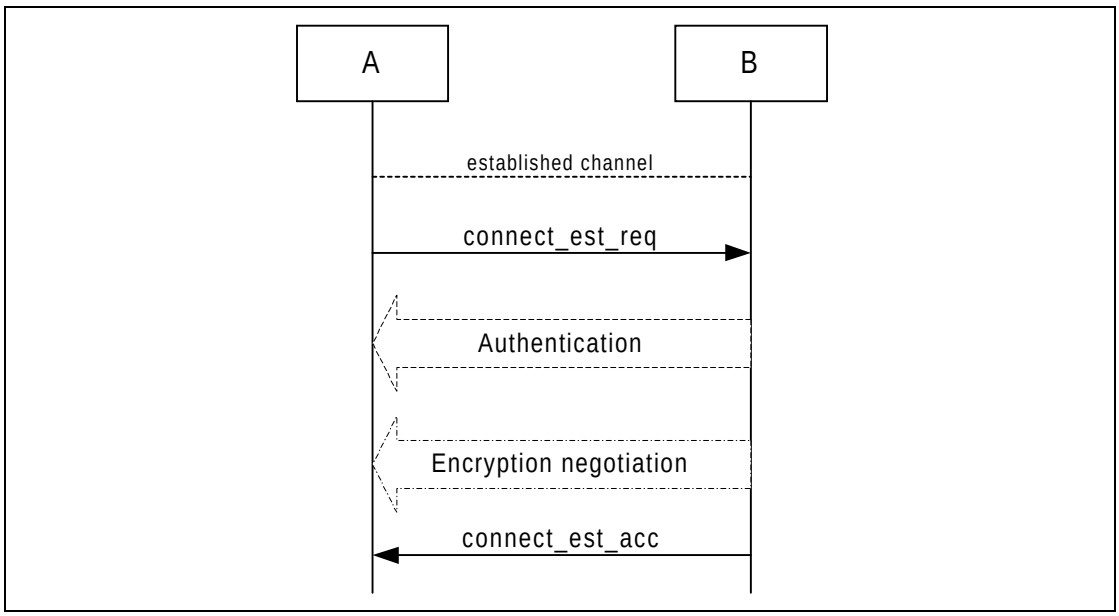


Figure 7.5: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

7.3.3.2 B in security mode 1 or 3

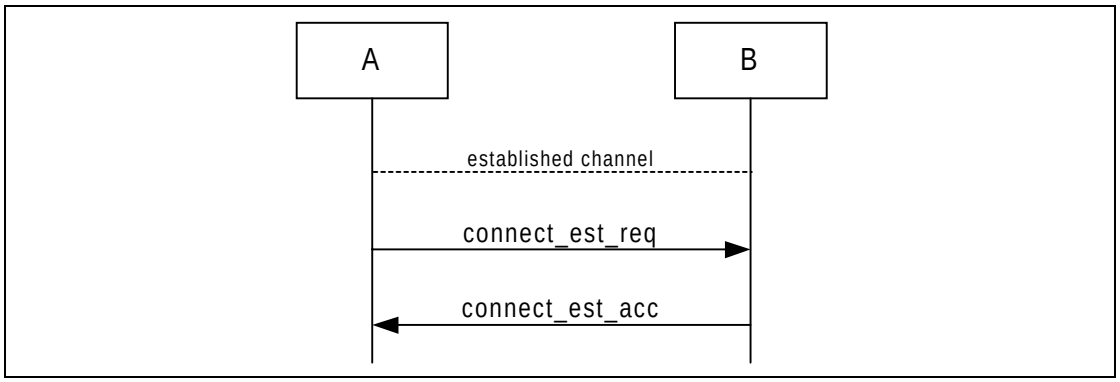


Figure 7.6: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

7.3.4 Conditions

Connection establishment starts after channel establishment is completed, when the initiator sends a connection establishment request (’connect_est_req’ is application protocol-dependent). This request may be a TCS SETUP message [5] in the case of a Bluetooth telephony application Cordless Telephony Profile, or initialization of RFCOMM and establishment of DLC [4] in the case of a serial port-based application Serial Port Profile (although neither TCS or RFCOMM use the term ’connection’ for this).

Connection establishment is completed when the acceptor accepts the connection establishment request (’connect_est_acc’ is application protocol dependent).

7.4 ESTABLISHMENT OF ADDITIONAL CONNECTION

When a Bluetooth device has established one connection with another Bluetooth device, it may be available for establishment of:

- A second connection on the same channel, and/or
- A second channel on the same link, and/or
- A second physical link.

If the new establishment procedure is to be towards the same device, the security part of the establishment depends on the security modes used. If the new establishment procedure is to be towards a new remote device, the device should behave according to active modes independent of the fact that it already has another physical link established (unless allowed co-incident radio and baseband events have to be handled).

8 DEFINITIONS

In the following, terms written with capital letters refer to states.

8.1 GENERAL DEFINITIONS

Mode A set of directives that defines how a device will respond to certain events.

Idle As seen from a remote device, a Bluetooth device is idle, or is in idle mode, when there is no link established between them.

Bond A relation between two Bluetooth devices defined by creating, exchanging and storing a common link key. The bond is created through the bonding or LMP-pairing procedures.

8.2 CONNECTION-RELATED DEFINITIONS

Physical channel A synchronized Bluetooth baseband-compliant RF hopping sequence.

Piconet A set of Bluetooth devices sharing the same physical channel defined by the master parameters (clock and BD_ADDR).

Physical link A Baseband-level connection[1] between two devices established using paging. A physical link comprises a sequence of transmission slots on a physical channel alternating between master and slave transmission slots.

ACL link An asynchronous (packet-switched) connection[1] between two devices created on LMP level. Traffic on an ACL link uses ACL packets to be transmitted.

SCO link A synchronous (circuit-switched) connection[1] for reserved bandwidth communications; e.g. voice between two devices, created on the LMP level by reserving slots periodically on a physical channel. Traffic on an SCO link uses SCO packets to be transmitted. SCO links can be established only after an ACL link has first been established.

Link Shorthand for an ACL link.

PAGE A baseband state where a device transmits page trains, and processes any eventual responses to the page trains.

PAGE_SCAN A baseband state where a device listens for page trains.

1. The term ’connection’ used here is not identical to the definition below. It is used in the absence of a more concise term.

Page The transmission by a device of page trains containing the Device Access Code of the device to which the physical link is requested.

Page scan The listening by a device for page trains containing its own Device Access Code.

Channel A logical connection on L2CAP level between two devices serving a single application or higher layer protocol.

Connection A connection between two peer applications or higher layer protocols mapped onto a channel.

Connecting A phase in the communication between devices when a connection between them is being established. (Connecting phase follows after the link establishment phase is completed.)

Connect (to service) The establishment of a connection to a service. If not already done, this includes establishment of a physical link, link and channel as well.

8.3 DEVICE-RELATED DEFINITIONS

Discoverable device A Bluetooth device in range that will respond to an inquiry (normally in addition to responding to page).

Silent device A Bluetooth device appears as silent to a remote device if it does not respond to inquiries made by the remote device. A device may be silent due to being non-discoverable or due to baseband congestion while being discoverable.

Connectable device A Bluetooth device in range that will respond to a page.

Trusted device A paired device that is explicitly marked as trusted.

Paired device A Bluetooth device with which a link key has been exchanged (either before connection establishment was requested or during connecting phase).

Pre-paired device A Bluetooth device with which a link key was exchanged, and the link key is stored, before link establishment.

Un-paired device A Bluetooth device for which there was no exchanged link key available before connection establishment was request.

Known device A Bluetooth device for which at least the BD_ADDR is stored.

Un-known device A Bluetooth device for which no information (BD_ADDR, link key or other) is stored.

Authenticated device A Bluetooth device whose identity has been verified during the lifetime of the current link, based on the authentication procedure.

8.4 PROCEDURE-RELATED DEFINITIONS

Paging A procedure for establishing a physical link of ACL type on baseband level, consisting of a page action of the initiator and a page scan action of the responding device.

Link establishment A procedure for establishing a link on LMP level. A link is established when both devices have agreed that LMP setup is completed.

Channel establishment A procedure for establishing a channel on L2CAP level.

Connection establishment A procedure for creating a connection mapped onto a channel.

Creation of a trusted relationship A procedure where the remote device is marked as a trusted device. This includes storing a common link key for future authentication and pairing (if the link key is not available).

Creation of a secure connection. A procedure of establishing a connection, including authentication and encryption.

Device discovery A procedure for retrieving the Bluetooth device address, clock, class-of-device field and used page scan mode from discoverable devices.

Name discovery A procedure for retrieving the user-friendly name (the Bluetooth device name) of a connectable device.

Service discovery Procedures for querying and browsing for services offered by or through another Bluetooth device.

8.5 SECURITY-RELATED DEFINITIONS

Authentication A generic procedure based on LMP-authentication if a link key exists or on LMP-pairing if no link key exists.

LMP-authentication An LMP level procedure for verifying the identity of a remote device. The procedure is based on a challenge-response mechanism using a random number, a secret key and the BD_ADDR of the non-initiating device. The secret key used can be a previously exchanged link key.

Authorization A procedure where a user of a Bluetooth device grants a specific (remote) Bluetooth device access to a specific service. Authorization implies that the identity of the remote device can be verified through authentication.

Authorize The act of granting a specific Bluetooth device access to a specific service. It may be based upon user confirmation, or given the existence of a trusted relationship.

LMP-pairing A procedure that authenticates two devices, based on a PIN, and subsequently creates a common link key that can be used as a basis for a trusted relationship or a (single) secure connection. The procedure consists of the steps: creation of an initialization key (based on a random number and a PIN), creation and exchange of a common link key and LMP-authentication based on the common link key.

Bonding A dedicated procedure for performing the first authentication, where a common link key is created and stored for future use.

Trusting The marking of a paired device as trusted. Trust marking can be done by the user, or automatically by the device (e.g. when in pairable mode) after a successful pairing.

9 ANNEX A (NORMATIVE): TIMERS AND CONSTANTS

The following timers are required by this profile.

Timer name	Recommended value	Description	Comment
$T_{GAP}(100)$	10.24 s	Normal time span that a Bluetooth device performs inquiry.	Used during inquiry and device discovery.
$T_{GAP}(101)$	10.625 ms	Minimum time in INQUIRY_SCAN.	A discoverable Bluetooth device enters INQUIRY_SCAN for at least $T_{GAP}(101)$ every $T_{GAP}(102)$.
$T_{GAP}(102)$	2.56 s	Maximum time between repeated INQUIRY_SCAN enterings.	Maximum value of the inquiry scan interval, $T_{inquiry\ scan}$.
$T_{GAP}(103)$	30.72 s	A Bluetooth device shall not be in a discoverable mode less than $T_{GAP}(103)$.	Minimum time to be discoverable.
$T_{GAP}(104)$	1 min	A Bluetooth device should not be in limited discover-able mode more than $T_{GAP}(104)$.	Recommended upper limit.

Table 9.1: Defined GAP timers

10 ANNEX B (INFORMATIVE): INFORMATION FLOWS OF RELATED PROCEDURES

10.1 LMP-AUTHENTICATION

The specification of authentication on link level is found in [2].

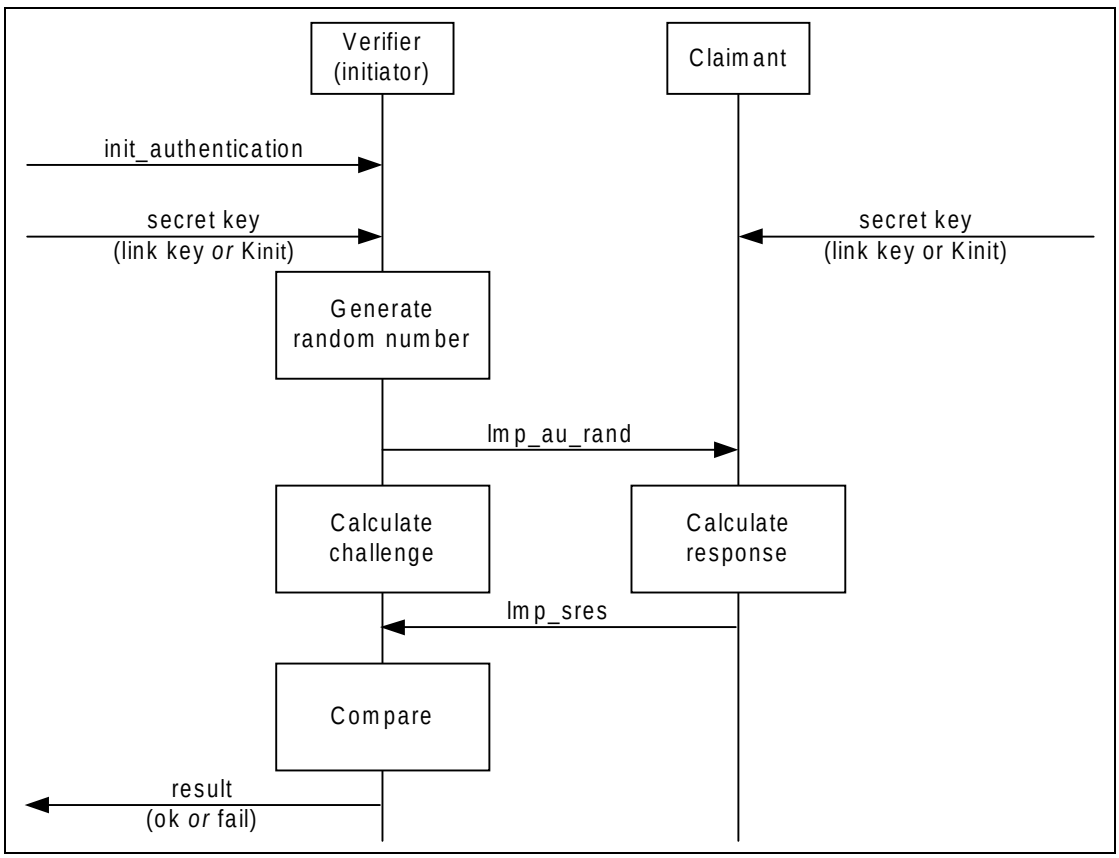


Figure 10.1: LMP-authentication as defined by [2].

The secret key used here is an already exchanged link key .

10.2 LMP-PAIRING

The specification of pairing on link level is found in [2].

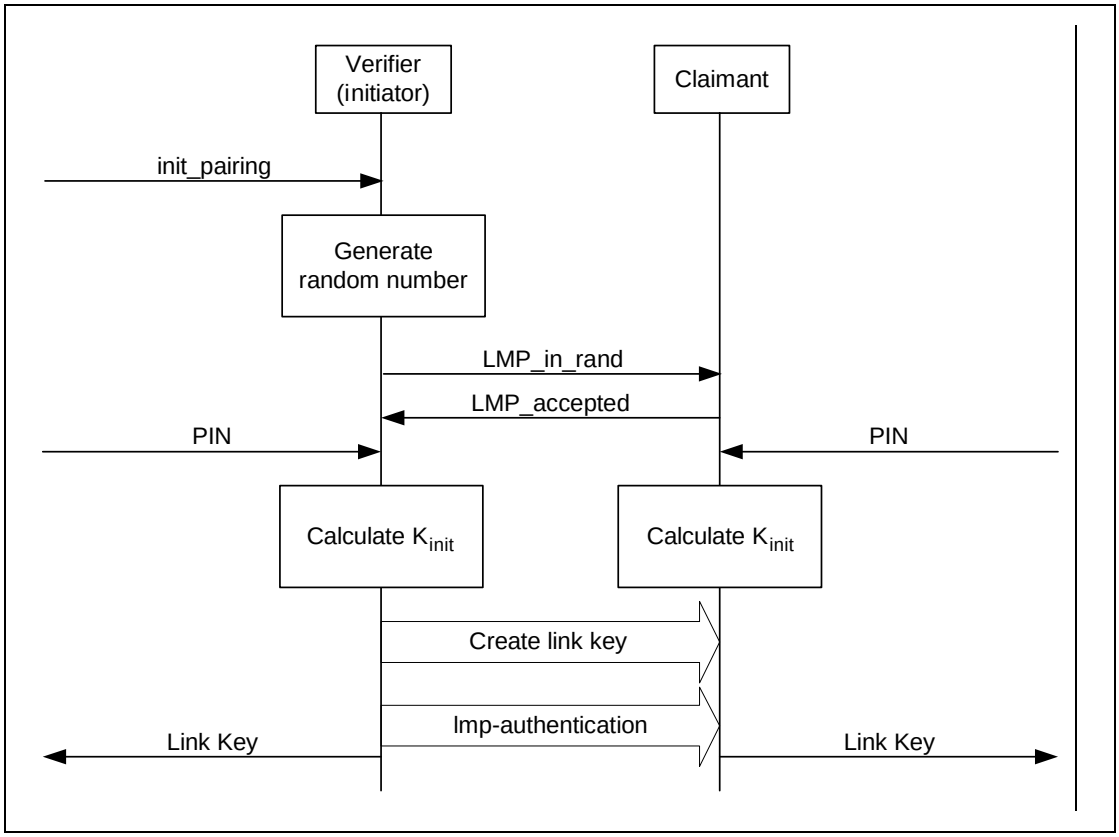


Figure 10.2: LMP-pairing as defined in [2].

The PIN used here is PN_{BB}.

The create link key procedure is described in section 3.3.4 of [2] and section 14.2.2 of [1]. In case the link key is based on a combination key, a mutual authentication takes place and shall be performed irrespective of current security mode.

10.3 SERVICE DISCOVERY

The Service Discovery Protocol [6] specifies what PDUs are used over-the-air to inquire about services and service attributes. The procedures for discovery of supported services and capabilities using the Service Discovery Protocol are described in the Service Discovery Application Profile. This is just an example.

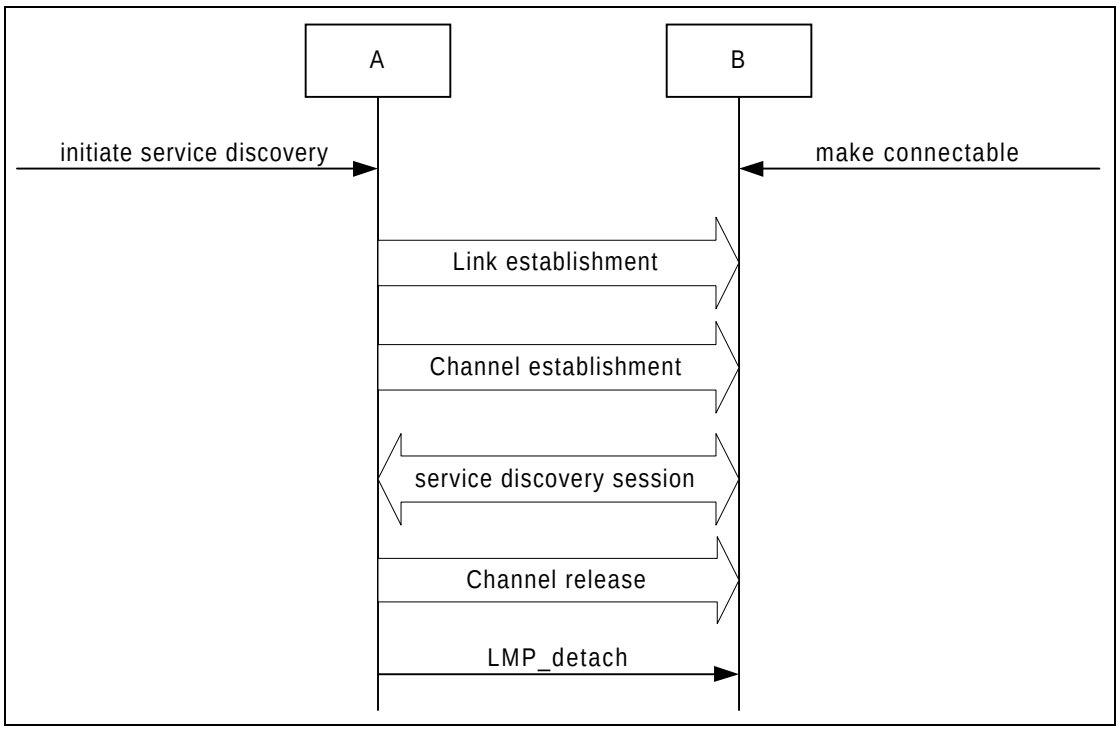


Figure 10.3: Service discovery procedure.

11 REFERENCES

[1] Bluetooth Baseband Specification

[2] Bluetooth Link Manager Protocol

[3] Bluetooth Logical Link Control and Adaptation Protocol

[4] Bluetooth RFCOMM

[5] Bluetooth Telephony Control Specification

[6] Bluetooth Service Discovery Protocol

[7] Bluetooth Service Discovery Application Profile

[8] Bluetooth Cordless Telephony Profile

[9] Bluetooth Serial Port Profile

[10] Bluetooth Security Architecture (white paper)

[11] Bluetooth Assigned Numbers
http://www.bluetooth.org/assigned-numbers.htm

Part K:2

SERVICE DISCOVERY APPLICATION PROFILE

This document defines the features and procedures for an application in a Bluetooth device to discover services registered in other Bluetooth devices and retrieve any desired available information pertinent to these services.

CONTENTS

1 Introduction 66
1.1 Scope 66
1.2 Symbols and conventions 67
2 Profile overview 68
2.1 Profile stack 68
2.2 Configurations and roles 69
2.3 User requirements and scenarios 70
2.4 Profile fundamentals 71
2.5 Conformance 71
3 User interface aspects 72
3.1 Pairing 72
3.2 Mode selection 72
4 Application layer 73
4.1 The service discovery application 73
4.2 Service primitives abstractions 75
4.3 Message sequence charts (MSCs) 77
5 Service Discovery 79
5.1 An SDP PDU exchange example 80
6 L2CAP 82
6.1 Channel types 83
6.2 Signalling 83
6.3 Configuration options 83
6.3.1 Maximum Transmission Unit (MTU) 83
6.3.2 Flush Time-out 83
6.3.3 Quality of Service 84
6.4 SDP transactions and L2CAP connection lifetime 84
7 Link Manager 86
7.1 Capability overview 86
7.2 Error behavior 87
7.3 Link policy 87
8 Link control 88
8.1 Capability overview 88
8.2 Inquiry 89
8.3 Inquiry scan 90
8.4 Paging 90
8.5 Page scan 90
8.6 Error behavior 90

9 References 91

9.1 Normative references 91

10 Definitions 92

11 Appendix A (Informative): Service primitives and the Bluetooth PDUs 93

FOREWORD

Interoperability between devices from different manufacturers is provided for a specific service and use case, if the devices conform to a Bluetooth SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications, and gives an unambiguous description of the air interface for specified service(s) and use case(s).

All defined features are process-mandatory. This means that, if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

It is expected that the number of services that can be provided over Bluetooth links will increase in an undetermined (and possibly uncontrolled) manner. Therefore, procedures need to be established to aid a user of a Bluetooth-enabled device to sort the ever-increasing variety of services that will become available to him/her. While many of the Bluetooth-enabled services that may be encountered are currently unknown, a standardized procedure can still be put into place on how to locate and identify them.

The Bluetooth protocol stack contains a Service Discovery Protocol (SDP) BT_SDP_spec:[7] that is used to locate services that are available on or via devices in the vicinity of a Bluetooth enabled device. Having located what services are available in a device, a user may then select to use one or more of them. Selecting, accessing, and using a service is outside the scope of this document. Yet, even though SDP is not directly involved in accessing services, information retrieved via SDP facilitates service access by using it to properly condition the local Bluetooth stack to access the desired service.

The service discovery profile defines the protocols and procedures that shall be used by a service discovery application on a device to locate services in other Bluetooth-enabled devices using the Bluetooth Service Discovery Protocol (SDP). With regard to this profile, the service discovery application is a specific user-initiated application. In this aspect, this profile is in contrast to other profiles where service discovery interactions between two SDP entities in two Bluetooth-enabled devices result from the need to enable a particular transport service (e.g. RFCOMM, etc.), or a particular usage scenario (e.g. file transfer, cordless telephony, LAN AP, etc.) over these two devices. Service discovery interactions of the latter kind can be found within the appropriate Bluetooth usage scenario profile documents.

The service discovery in the other profile documents has a very narrow scope; e.g. learning about the protocols and related protocol parameters needed for accessing a particular service. Nevertheless, the fundamentals of the service discovery procedures covered in this profile document, and the use of the Bluetooth protocols in support of these procedures can be replicated in other profile documents as well. The only difference is that for the other profiles these procedures are initiated by application-level actions within the applications described by the corresponding profiles, as opposed to user-level actions for this profile.

SDP provides direct support for the following set of service inquiries:

- Search for services by service class;
- Search for services by service attributes; and
- Service browsing.

The generic service discovery application considered for this profile also covers the above service inquiry scenarios.

The former two cases represent searching for known and specific services. They provide answers to user questions like: “Is service A, or is service A with characteristics B and C, available?” The latter case represents a general service search and provides answers to questions like: “What services are available?” or “What services of type A are available?”

The above service inquiry scenarios can be realized two-fold:

- By performing the service searches on a particular device that a user ‘consciously’ has already connected to, and/or
- By performing the service searches by ‘unconsciously’ connecting to devices discovered in a device's vicinity.

Both of the above approaches require that devices need first to be discovered, then linked with, and then inquired about the services they support.

1.2 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [3].

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 shows the Bluetooth protocols and supporting entities involved in this profile.

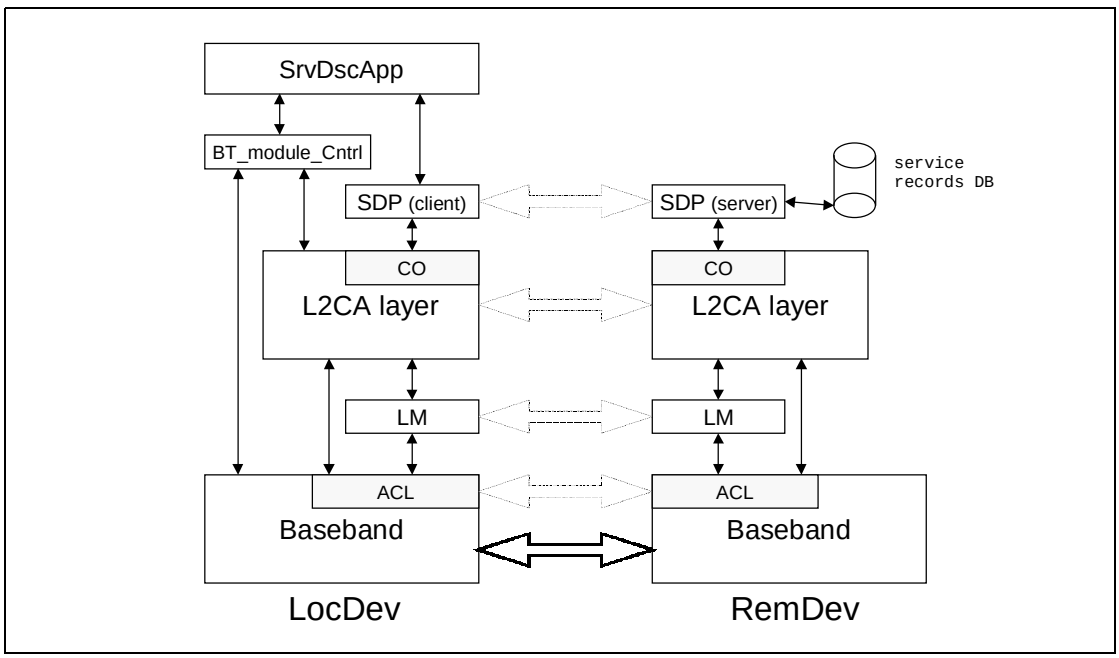


Figure 2.1: The Bluetooth protocol stack for the service discovery profile

The service discovery user application (SrvDscApp) in a local device (LocDev) interfaces with the Bluetooth SDP client to send service inquiries and receive service inquiry responses from the SDP servers of remote devices (RemDevs) BT_SDP_spec:[7]. SDP uses the connection-oriented (CO) transport service in L2CAP, which in turn uses the baseband asynchronous connectionless (ACL) links to ultimately carry the SDP PDUs over the air.

Service discovery is tightly related to discovering devices, and discovering devices is tightly related to performing inquiries and pages. Thus, the SrvDscApp interfaces with the baseband via the BT_module_Cntrl entity that instructs the Bluetooth module when to enter various search modes of operation.[1]

1. The BT_module_Cntrl may be part of a Bluetooth stack implementation (and thus be shared by many Bluetooth-aware applications) or a 'lower part' of the SrvDscApp. Since, no assumptions about any particular stack or SrvDscApp implementations are made, the BT_module_Cntrl entity represents a logical entity separate from the SrvDscApp, which may or may not be part of the SrvDscApp itself, a stack component, or any other appropriate piece of code.

The service records database (DB) shown in Figure 2.1 next to an SDP server is a logical entity that serves as a repository of service discovery-related information. The ‘physical form’ of this database is an implementation issue outside the scope of this profile.

2.2 CONFIGURATIONS AND ROLES

The following roles are defined in this profile:

- **Local device (LocDev):** A LocDev is the device that initiates the service discovery procedure. A LocDev must contain at least the *client* portion of the Bluetooth SDP architecture BT_SDP_spec:[7]. A LocDev contains the service discovery application (SrvDscApp) used by a user to initiate discoveries and display the results of these discoveries.
- **Remote Device(s) (RemDev(s)):** A RemDev is any device that participates in the service discovery process by responding to the service inquiries generated by a LocDev. A RemDev must contain at least the *server* portion of the Bluetooth SDP architecture BT_SDP_spec:[7]. A RemDev contains a service records database, which the server portion of SDP consults to create responses to service discovery requests.

The LocDev or RemDev role assigned to a device is neither permanent nor exclusive. A RemDev may also have a SrvDscApp installed into it as well as an SDP client, and a LocDev may also have an SDP server. In conjunction with which device has an SrvDscApp installed, an SDP-client installed, and an SDP-server installed, the assignment of devices to the above roles is relative to each individual SDP (and related) transaction and which device initiates the transaction. Thus, a device could be a LocDev for a particular SDP transaction, while at the very same time be a RemDev for another SDP transaction.

With respect to this profile, a device without a UI (directly or indirectly available) for entering user input and returning the results of service searches is not considered as a candidate for a LocDev. Nevertheless, even if such a device is not considered as a candidate for a LocDev, the procedures presented in the following sections can still apply if applications running in such a device need to execute a service discovery transaction.

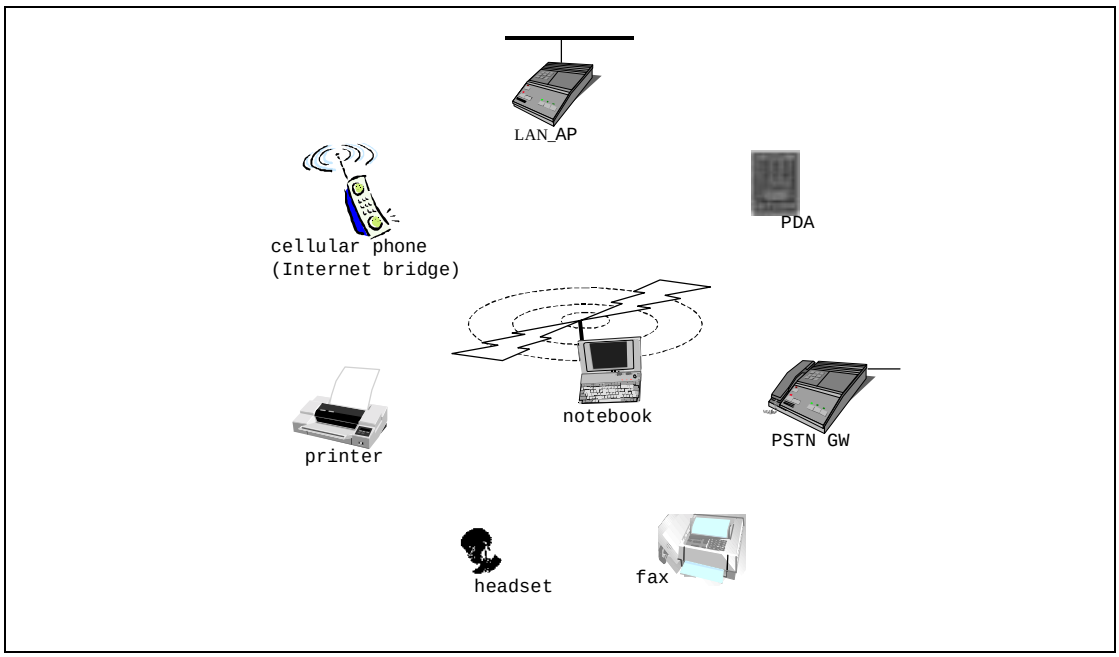


Figure 2.2: A typical service discovery scenario

The figure above shows a local device (the notebook) inquiring for services among a plethora of remote devices.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Search for services by service class,
- Search for services by service attributes, and
- Service browsing.

The first two cases represent searching for known and specific services, as part of the user question “Is service A, or is service A with characteristics B and C, available?” The latter case represents a general service search that is a response to the user question “What services are available?”

This profile implies the presence of a Bluetooth-aware, user-level application, the SrvDscApp, in a LocDev that interfaces with the SDP protocol for locating services. In this aspect, this profile is unique as compared to other profiles. It is a profile that describes an application that interfaces to a specific Bluetooth protocol to take full advantage of it for the direct benefit of an end-user.

2.4 PROFILE FUNDAMENTALS

Before any two Bluetooth-equipped devices can communicate with each other the following may be needed:

- The devices need to be powered-on and initialized. Initialization may require providing a PIN for the creation of a link key, for device authorization and data encryption.
- A Bluetooth link has to be created, which may require the discovery of the other device's BD_ADDR via an inquiry process, and the paging of the other device.

While it may be seem natural to consider a LocDev serving as a Bluetooth master and the RemDev(s) serving as Bluetooth slave(s), there is no such requirement imposed on the devices participating in this profile. Service discovery as presented in this document can be initiated by either a master or a slave device at any point for which these devices are members of the same piconet. Also, a slave in a piconet may possibly initiate service discovery in a new piconet, provided that it notifies the master of the original piconet that it will be unavailable (possibly entering the hold operational mode) for a given amount of time.[2]

The profile does not require the use of authentication and/or encryption. If any of these procedures are used by any of the devices involved, service discovery will be performed only on the subset of devices that pass the authentication and encryption security ‘roadblocks’ that may impose to each other. In other words, any security restrictions for SDP transactions are dictated by the security restrictions already in place (if any) on the Bluetooth link.

2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

2. Recall that a master of a piconet cannot initiate a new piconet. Since a piconet is ultimately identified by the BD_ADDR and the Bluetooth clock of its master, the latter piconet will be identical to and indistinguishable from the former.

3 USER INTERFACE ASPECTS

3.1 PAIRING

No particular requirements regarding pairing are imposed by this profile. Pairing may or may not be performed. Whenever a LocDev performs service discovery against as yet 'unconnected' RemDev(s), it shall be the responsibility of the SrvDscApp to allow pairing prior to connection, or to by-pass any devices that may require pairing first. This profile is focused on only performing service discovery whenever the LocDev can establish a legitimate and useful baseband link[3] with RemDev(s).

3.2 MODE SELECTION

This profile assumes that, under the guidance of the SrvDscApp, the LocDev shall be able to enter the inquiry and/or page states. It is also assumed that a RemDev with services that it wants to make available to other devices (e.g. printer, a LAN DAP, a PSTN gateway, etc.) shall be able to enter the inquiry scan and/or page scan states. For more information about the inquiry and page related states see Section 8.

Since the SrvDscApp may also perform service inquiries against already connected RemDevs, it is not mandatory according to the profile that a LocDev always be the master of a connection with a RemDev. Similarly, a RemDev may not always be the slave of a connection with a LocDev.

3. A legitimate and useful baseband link is a Bluetooth baseband link that is properly authenticated and encrypted (if so desired), whenever any of these options are activated by any of the devices participating in this profile.

4 APPLICATION LAYER

4.1 THE SERVICE DISCOVERY APPLICATION

In this subsection, the operational framework of the SrvDscApp is presented.[4] Figure 4.1 shows alternative possibilities for a SrvDscApp.

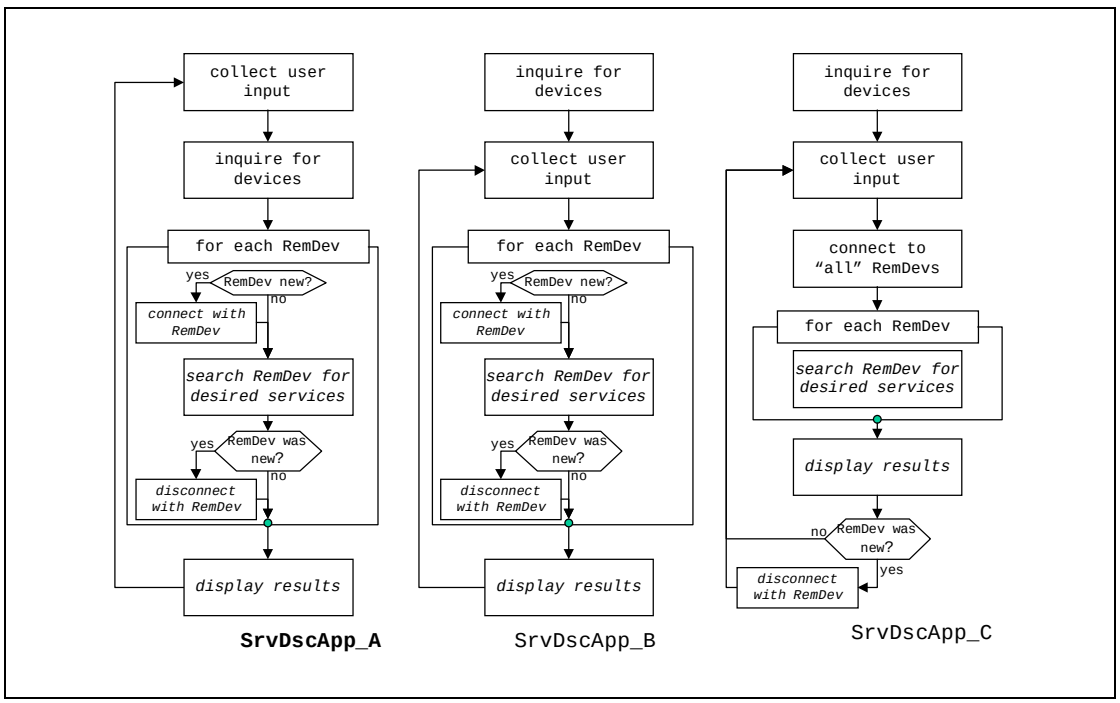


Figure 4.1: Three possible SrvDscApps

The SrvDscApp alternatives shown in Figure 4.1, which are not exhaustive by any means, achieve the same objectives but they follow different paths for achieving them. In the first alternative (SrvDscApp_A), the SrvDscApp on a LocDev inquires its user to provide information for the desired service search. Following this, the SrvDscApp searches for devices, via the Bluetooth inquiry procedure. For each device found, the LocDev will connect to it, perform any necessary link set-up, see related procedures in Generic Access Profile [3], and then inquire it for the desired services. In the second alternative (SrvDscApp_ B), the inquiry of devices is done prior to collecting user input for the service search.[5]

4. This profile does not dictate any particular implementation for a SevDisApp. It only presents the procedures needed to achieve its objectives.
5. Device inquiries may even occur by means outside the scope of a particular SrvDscApp implementation. But, since such other means are not guaranteed to exist, it is recommended that the SrvDscApp activates device inquiries too.

In the first two alternatives, page, link creation, and service discovery are done sequentially on a per RemDev basis; i.e., the LocDev does not page any new RemDev prior to completing the service search with a previous RemDev and (if necessary) disconnecting from it. In the last alternative (SrvDscApp_C), the LocDev, under the control of the SrvDscApp, will first page all RemDevs, then will create links with all of these devices (up to a maximum of 7 at a time), and then inquire all the connected devices for the desired services.

Just as an example, we focus on a SrvDscApp similar to the one represented by the SrvDscApp_A in Figure 4.1. In summary, SrvDscApp (for ease of notation, the suffix '_A' has been dropped) has the following features:

- The SrvDscApp activates Bluetooth inquiries following a user request for a service search,
- For any new RemDev found following an inquiry, the SrvDscApp will finish service discovery and terminate its link against this device prior to attempting to connect to the next RemDev,
- For any RemDev already connected, the LocDev does not disconnect following service discovery, and
- The user of the SrvDscApp has the option of a trusted and untrusted mode of operation, whereby the SrvDscApp permits connections –

    a) only with trusted RemDev, or

    b) with any of the devices above plus any newly discovered RemDevs that require nothing more beyond possibly pairing with the default all-zero PIN, or

    c) with any of the devices above, plus any additional RemDev for which the user explicitly enters a non-zero PIN.

The above options have to do with the degree of user involvement in configuring and interacting with the SrvDscApp and setting the security levels that the user is willing to accept for the service searches. When selecting options (a) or (b), then for the devices with which no legitimate connections can be established, it is assumed that the SrvDscApp ignores them without any cue to its user (however, this too is an implementation issue).

When a LocDev performs a service discovery search, it does so against three different types of RemDevs:

1. *trusted devices:* These are devices that are currently not connected with the LocDev but the LocDev device has already an established trusted relation with.
2. *unknown (new) devices:* These are untrusted devices that are currently not connected with the LocDev.
3. *connected devices:* These are devices that are already connected to the LocDev.

To discover type 1 or 2 RemDevs, the SrvDscApp needs to activate the Bluetooth inquiry and/or page processes. For type 3 RemDevs, the latter processes are needed.To perform its task, SrvDscApp needs to have access to the BD_ADDR of the devices in the vicinity of a LocDev, no matter whether these devices have been located via a Bluetooth inquiry process or are already connected to the LocDev. Thus, BT_module_Cntr in a LocDev shall maintain the list of devices in the vicinity of the LocDev and shall avail this list to the SrvDscApp.

4.2 SERVICE PRIMITIVES ABSTRACTIONS

This section briefly describes the functionality of a SrvDscApp. This functionality is presented in the form of service primitive abstractions that provide a formal framework for describing the user expectations from a SrvDscApp. It is assumed that the underlying Bluetooth stack can meet the objectives of these service primitive abstractions directly or indirectly.[6] The exact syntax and semantics of the service primitive abstractions (or simply “service primitives”) may be platform-dependent (e.g. an operating system, a hardware platform, like a PDA, a notebook computer, a cellular phone, etc.) and are beyond the scope of this profile. However, the functionality of these primitives is expected to be available to the SrvDscApp to accomplish its task.

Table 4.1 contains a minimum set of enabling service primitives to support a SrvDscApp. Low-level primitives like **openSearch**(.) or **closeSearch**(.) are not shown and are assumed to be part of the implementation of the primitives shown whenever necessary. Different implementations of the Bluetooth stack shall (at a minimum) enable the functions that these service primitives provide. For example, the **serviceSearch**(.) service primitive permits multiple identical operations to be handled at once. A stack implementation that requires an application to accomplish this function by iterating through the multiple identical operations one-at-a-time will be considered as enabling the function of this service primitive.[7] The service primitives shown next relate only to service primitives whose invocation result or relate to an over-the-air data exchange using the Bluetooth protocols. Additional service primitives can be envisioned relating to purely local operations like *service registration*, but these primitives are outside the scope of this profile.

6. These service primitive abstractions do *not* represent programming interfaces, even though they may be related to them. The word ‘directly’ is used to describe the possibility that the described function is the result of a single appropriate call of the underlying Bluetooth stack implementation. The word ‘indirectly’ is used to describe the possibility that the described function can be achieved by combining the results from multiple appropriate calls of the underlying Bluetooth stack implementation.
7. Even though the service primitives presented in this profile are assumed to act upon a local device for accessing *physically* remote devices, they are general enough to apply in cases where the ‘remote device’ characterization is only a logical concept; i.e. inquired service records and service providers are located within the same device that invokes these primitives. This general situation is outside the scope of this profile.

service primitive abstraction	resulted action
serviceBrowse (LIST(*RemDev)* LIST(*RemDevRelation*) LIST(*browseGroup*) *getRemDevName stopRule*)	a search for services (service browsing) that belong to the list of *browseGroup* services in the devices in the list of RemDevs; the search may be further qualified with a list of *RemDevRelation* parameters, whereby a user specifies the trust and connection relation of the devices to be searched; e.g. search only the devices that are in the *RemDev* list for which there is a trust relation already established; when the *getRemDevName* parameter is set to "yes," the names of the devices supporting the requested services are also returned; the search continues until the stopping rule *stopRule* is satisfied
serviceSearch (LIST(*RemDev)* LIST(*RemDevRelation*) LIST(*searchPattern*, *attributeList*) *getRemDevName stopRule*)	a search whether the devices listed in the list of RemDevs support services in the requested list of services; each service in the list must have a service search pattern that is a superset of the *searchPattern*; for each such service the values of the attributes contained in the corresponding *attributeList* are also retrieved; the search may be further qualified with a list of *RemDevRelation* parameters, whereby a user specifies the trust and connection relation of the devices to be searched (e.g. search only the devices that are in the *RemDev* list for which there is a trust relation already established); when the *getRemDevName* parameter is set to "yes," the names of the devices supporting the requested services are also returned; the search continues until the stopping rule *stopRule* is satisfied
enumerateRemDev (LIST(*classOfDevice*) *stopRule*)	a search for RemDev in the vicinity of a LocDev; RemDev searches may optionally be filtered using the list of *classOfDevice (*e.g. LAN APs); the search continues until the stopping rule *stopRule* is satisfied
terminatePrimitive (*primitiveHandle returnResults*)	a termination the actions executed as a result of invoking the services primitive identified by the *primitiveHandle*;* optionally, this service primitive may return any partially accumulated results related to the terminated service primitive

Table 4.1: Service primitives in support of SrvDscApp

*. It is assumed that each invocation of a service primitive can be identified by a *primitiveHandle,* the realization of which is implementation-dependent.

The *stopRule* parameter is used to guarantee a graceful termination of a service search. It could represent the number of search items found, or the duration of search, or both. A Bluetooth stack implementation may not expose this parameter, in which case it should provide guarantees that all searches terminate within a reasonable amount of time, for example, say, 120sec.

The **enumerateRemDev**(.) service primitive is directly related to the inquiry mode of operation for the baseband. It also relates to the collection of RemDev that a LocDev is currently connected with. This service is exported to the SrvDscApp via the BT_module_Cntr, see Figure 2.1. The interface between BT_module_Cntr and baseband is for activating Bluetooth inquiries and collecting the results of these inquiries. The interface between the BT_module_Cntrl and (an) L2CAP (implementation) is for keeping track of the RemDev that currently are connected to the LocDev.

The result of the **enumerateRemDev**(.) service primitive can be used with the **serviceSearch**(.) to search for desired services in the devices found. Once again, based on the implementation of the Bluetooth stack, this service primitive may not be provided explicitly, but its service may be provided within other service primitives; e.g. the **serviceSearch**(.).

Missing primitive parameters shall be interpreted (whenever appropriate) as a general service search on the remaining parameters. For example, if the LIST(*RemDev*) parameter is missing from the **serviceSearch**(.), it means that the search shall be performed against any device found in the vicinity of a LocDev. In this case, the first two service primitives may be combined to a single one.

The above service primitives return the requested information, whenever found. Based on the way that these service primitives are supported by a Bluetooth stack implementation, the results of a search may directly return by the corresponding calling function, or a pointer to a data structure may be returned that contains all the relevant information. Alternatively, a Bluetooth stack implementation may have altogether different means for providing the results of a search.

4.3 MESSAGE SEQUENCE CHARTS (MSCS)

This profile is concerned with three distinct Bluetooth procedures. Device discovery, device name discovery, service discovery. Note that each one of these procedures does not preclude any other; e.g. to connect to a RemDev, a LocDev may have to first discover it, and it may also ask for its name. The MSCs relating to the first two procedures (i.e., device and name discovery) are provided in section 2 of LM/HCI_MSCs:[6]. Sections 3, 4.1 and 4.2 of LM/HCI_MSCs:[6] provide the MSCs relating to the third procedure (i.e., service discovery). See also section 4 of BT_LM_spec:[4]. The first two procedures do not require host intervention, while the third does.

Figure 4.2 summarizes the key message exchange ‘phases’ encountered during the execution of this profile. Not all procedures are present at all times, and not all devices need to go through these procedures all the time. For example, if authentication is not required, the authentication phase in the figure will not be executed. If the SrvDsvApp needs to inquire for services on a specific RemDev with which the LocDev is currently connected, inquiries and pages

may not be executed. In the figure, the conditions under which particular phases are executed or not are also provided.

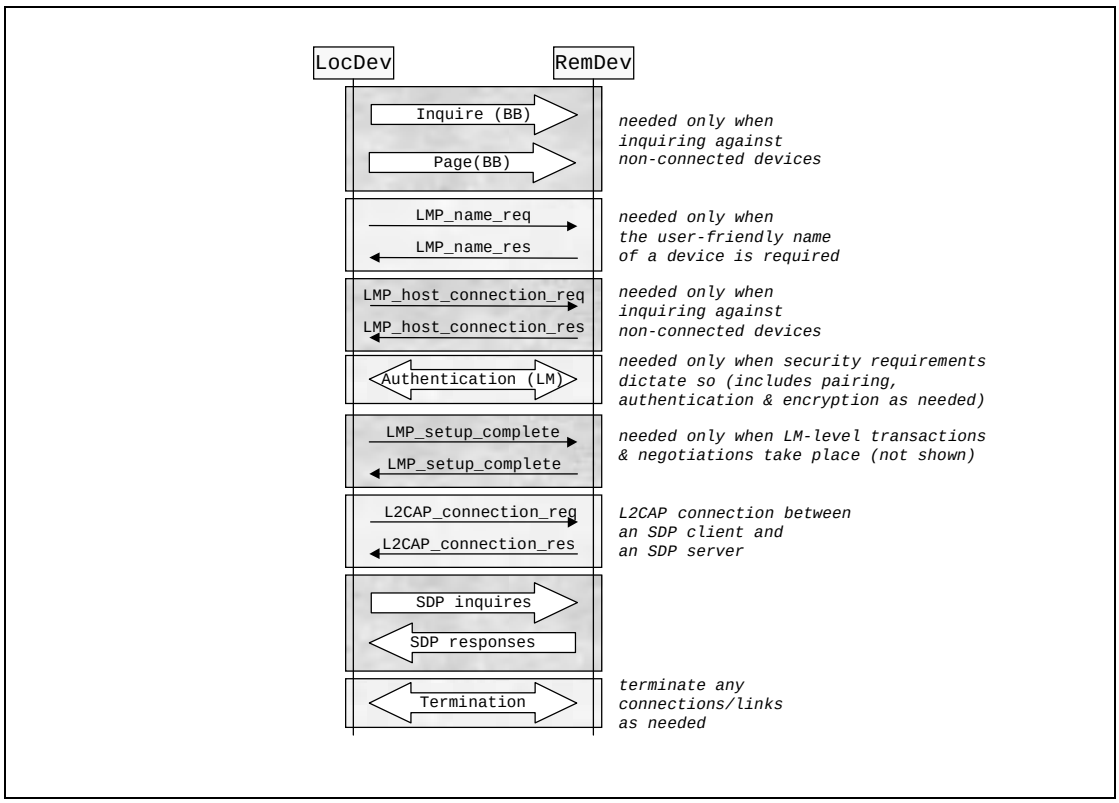


Figure 4.2: Bluetooth processes in support of this profile

In addition to the MSC in Figure 4.2, Annex A shows what Bluetooth procedures and PDUs are needed to support the service primitives presented in Section 4.2.

5 SERVICE DISCOVERY

The service discovery application does not make use of SDP as a means of accessing a service, but rather as a means of informing the user of a LocDev about the services that are available to his/her device by (and possibly via) RemDev(s). BT-aware applications running in a local device can also use the procedures described in this and the following sections to retrieve any pertinent information that will facilitate the application in accessing a desired service in a remote device.

Table 5.1 shows the SDP feature requirements in a LocDev and in a RemDev.

	SDP feature	**Support in LocDev**	**Support in RemDev**
1.	SDP client	M	O
2.	SDP server	O	M

Table 5.1: SDP feature requirements

Table 5.2 shows the SDP PDUs can be exchanged between devices following this profile.

SDP PDUs	**Ability to Send**		**Ability to Receive**	
	LocDev	**RemDev**	**LocDev**	**RemDev**
SDP_ErrorResponce	C1	M	M	C1
SDP_ServiceSearchRequest	M	C1	C1	M
SDP_ServiceSearchResponse	C1	M	M	C1
SDP_ServiceAttributeRequest	M	C1	C1	M
SDP_ServiceAttributeResponse	C1	M	M	C1
SDP_ServiceSearchAttributeRequest	M	C1	C1	M
SDP_ServiceSearchAttributeResponse	C1	M	M	C1
Comments:				
[C1]: With regard to this current profile, these PDU transmissions will not occur. Nevertheless, since a device could act as a LocDev on some occasions and as a RemDev on others, these PDU transmission may still take place between these devices.				

Table 5.2: Allowed SDP PDUs

5.1 AN SDP PDU EXCHANGE EXAMPLE

Figure 5.1 shows two examples of SDP PDU exchanges. In particular, it shows PDU exchange sequences for the inquiry and retrieval of any information pertinent to a particular Bluetooth profile.

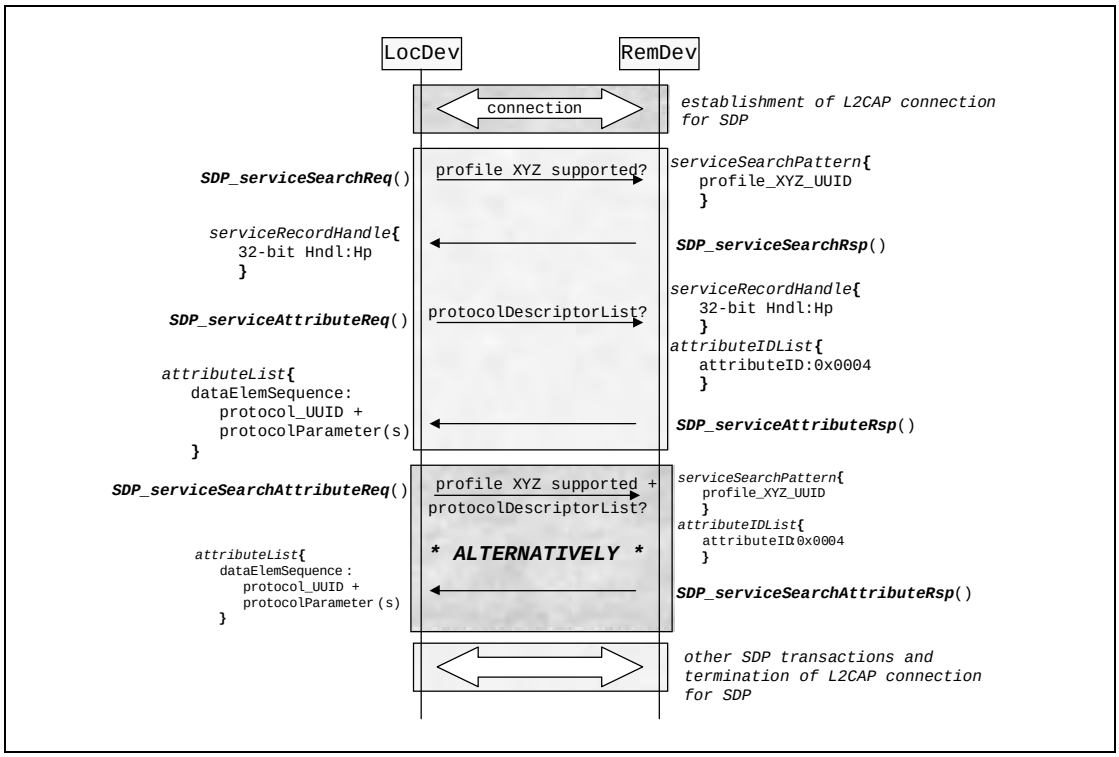


Figure 5.1: SDP PDU exchange examples for retrieving protocolDescriptorLists

For each PDU sent, the figure shows which device sends it (shown on the starting side of an arrow) and any relative information that this PDU carries (shown on the ending side of an arrow). Note that the LocDev sends request PDUs, while the RemDev sends back response PDUs.

Two alternatives are shown utilizing different SDP PDUs to ultimately retrieve the same information – the *protocolDescriptorList* attribute from devices that support a specific Bluetooth profile. With the first alternative, the desired information is derived in two steps.

- The LocDev sends an *SDP_serviceSearchReq* PDU which contains a service search pattern composed of the UUID associated with the desired profile; see section 4.3 of BT_ASN:[2]. The desired profile (profile 'XYZ') is identified by its UUID, denoted in the figure as 'profile_XYZ_UUID.' In its response PDU, the SDP server returns one or more 32-bit service record handles whose corresponding service records contain the 'profile_XYZ_UUID' UUID. In the figure, only one such handle is shown, denoted as 'prHndl'.
- The LocDev then enters prHndl in an *SDP*_serviceAttribute PDU together with one or more attribute IDs. In this example, the attribute of interest is the

protocolDescriptorList, whose attribute ID is 0x0004. The SDP server then, in its response, returns the requested protocol list.

In the event that no service record containing the desired service search pattern is found in the SDP server, the *SDP_serviceSearchResp* PDU will contain an empty *serviceRecordHandleList* and a *totalServiceRecordCount* parameter set to its minimum value; see section 4.5.2 of BT_SDP_spec:[7].

If the desired attributes do not exist in the SDP server, the *SDP_serviceAttributeResp* PDU will contain an empty *attributeList* and an *attributeListByteCount* parameter set to its minimum value, see section 4.6.2 of BT_SDP_spec:[7].

With the second alternative, the desired attributes are retrieved in one step:

- The LocDev sends an *SDP_serviceSearchAttributeReq* PDU where both the desired profile is included (service search pattern: profile_XYZ_UUID) and the desired attribute(s) is provided (attribute ID: 0x0004). In its response the SDP server will provide the requested attribute(s) from the service record(s) that matches the service search pattern.

In case no service record containing the desired service search pattern and/or the desired attribute(s) is found in the SDP server, the *SDP_serviceSearchAttributeResp* PDU will contain an empty *attributeLists* and an *attributeListsByteCount* parameter set to its minimum value, see section 4.7.2 of BT_SDP_spec:[7].

While, in the example in Figure 5.1, only very few service attributes are shown retrieved by the SDP client, additional information could and should be requested. Particularly in cases where service information is to be cached for future use, an SDP client should also request any pertinent information that can aid in assessing whether cached information has become stale.
The service attributes *serviceDatabaseState*, *serviceRecordState*, and *serviceInfoTimeToLive* have been defined for this purpose in BT_SDP_spec:[7]; see sections 5.2.4, 5.1.3 and 5.1.8 respectively.

6 L2CAP

The following text, together with the associated subclauses, defines the mandatory requirements with regard to this profile.

	L2CAP procedure		Support in LocDev	Support in RemDev
1.	Channel types			
		Connection-oriented channel	M	M
		Connectionless channel	X1	X1
2.	Signalling			
		Connection Establishment	M	C1
		Configuration	M	M
		Connection Termination	M	C2
		Echo	M	M
		Command Rejection	M	M
3.	Configuration Parameter Options			
		Maximum Transmission Unit	M	M
		Flush Time-out	M	M
		Quality of Service	O	O

Comments:

[X1]: This feature is not used in this profile, but its use by other applications running simultaneously with this profile is not excluded.

[C1]: An SDP server shall not (and cannot) initiate an L2CAP connection for SDP transactions. Nevertheless, the device that the SDP server resides in may also have an SDP client that may initiate an L2CAP connection for SDP transactions. Such action does not contradict the execution of this profile. In any case, a RemDev shall be able to process incoming requests for connection establishment.

[C2] Under normal operation, an SDP server shall not initiate the process of terminating an L2CAP connection for SDP. However, exceptional cases, such as when a RemDev shuts down during the execution on an SDP transaction, cannot be excluded. In such a case, prior to the final power-off, the RemDev may gracefully (or not!) terminate all its active L2CAP connections by sending connection termination PDUs. In any case, a RemDev shall always be able to process incoming requests for connection termination.

Table 6.1: L2CAP procedures

6.1 CHANNEL TYPES

In this profile, only connection-oriented channels shall be used. In particular, no L2CAP broadcasts are to be used for this profile.

6.2 SIGNALLING

For the purpose of retrieving SDP-related information, only a LocDev can initiate an L2CAP connection request and issue an L2CAP connection request PDU; for exceptions, see comments C1 and C2 on Table 6.1. Likewise with the corresponding L2CAP connection terminations, and the same exceptional comments C1 and C2 on Table 6.1 apply. Other than that, SDAP does not impose any additional restrictions or requirements on L2CAP signalling.

In the PSM field of the Connection Request packet, the value 0x0001 (see section 5.2 of BT_L2CAP_spec:[5]) shall be used to indicate the request for creation of an L2CAP connection for accessing the SDP layer.

6.3 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the service discovery profile.

6.3.1 Maximum Transmission Unit (MTU)

This profile does not impose any additional restrictions to MTU beyond the ones stated in section 6.1 of BT_L2CAP_spec:[5]. If no MTU negotiation takes place, the default MTU value in section 6.1 of BT_L2CAP_spec:[5] shall be used.

For efficient use of the communication resources, the MTU shall be selected as large as possible, while respecting any physical constraints imposed by the devices involved, and the need that these devices continue honoring any already agreed upon QoS contracts with other devices and/or applications. It is expected that during the lifetime of an L2CAP connection for SDP transactions (also referred to as the ‘SDP session’, see Section 6.4) between two devices, any one of these devices may become engaged in an L2CAP connection with another device and/or application. If this new connection has ‘non-default’ QoS requirements, the MTU for the aforementioned SDP session is allowed to be re-negotiated during the lifetime of this SDP session, to accommodate the QoS constraints of the new L2CAP connection.

6.3.2 Flush Time-out

The SDP transactions are carried over an L2CAP reliable channel. The flush time-out value (see section 6.2 of BT_L2CAP_spec:[5]) shall be set to its default value 0xFFFF.

6.3.3 Quality of Service

The use of Quality of Service (QoS) and QoS negotiation is optional. If QoS is to be negotiated, the default settings in section 6.4 of BT_L2CAP_spec:[5] shall be used. In particular, SDP traffic shall be treated as a best-effort service type traffic.

6.4 SDP TRANSACTIONS AND L2CAP CONNECTION LIFETIME

While, in general, SDP transactions comprise a sequence of service request-and-response PDU exchanges, SDP itself constitutes a connectionless datagram service in that no SDP-level connections are formed prior to any SDP PDU exchange. SDP delegates the creation of connections on its behalf to the L2CAP layer. It is thus the responsibility of SDP – or, more correctly, of the SDP layer – to request the L2CAP layer to 'tear down' these connections on its behalf as well.

Since SDP servers are considered stateless, 'tearing down' an L2CAP connection after a service request PDU is sent (as a true connectionless service may imply) will be detrimental to the SDP transaction. Moreover, significant performance penalty will have to be paid if, for each SDP PDU transmission, a new L2CAP connection is to be created. Thus, L2CAP connections for SDP transactions shall last more than the transmission of a single SDP PDU.

An SDP *session* between an SDP client and an SDP server represents the time interval that the client and the server have the same L2CAP connection continuously present. A *minimal* SDP transaction will represent a single exchange of an SDP request PDU transmission from an SDP client to an SDP server, and the transmission of a corresponding SDP response PDU from the SDP server back to the SDP client. With respect to this profile, under normal operational conditions, the minimum duration of an SDP session shall be the duration of a minimal SDP transaction.

An SDP session may last less than the minimum required in the event of unrecoverable (processing or link) errors in layers below SDP in the LocDev and RemDev, or in the SDP layer and the service records database in the RemDev. An SDP session may also be interrupted by user intervention that may terminate the SDP session prior to the completion of an SDP transaction.

The above minimum duration of an SDP session guarantees smooth execution of the SDP transactions. For improved performance, implementers may allow SDP sessions to last longer than the minimum duration of an SDP session. As a general implementation guideline, an SDP session shall be maintained for as long as there is a need to interact with a specific device. Since the latter time is in general unpredictable, SDP implementations may maintain timers used to time periods of SDP transaction inactivity over a specific SDP session.

SDP implementations may also rely on explicit input received from a higher layer (probably initiated from the SrvDscApp itself) to open and close an SDP session with a particular device using low level primitives; e.g. **openSearch**(.) and **closeSearch**(.). Finally, an implementation may permit users to interrupt an SDP session at any time, see the **terminatePrimitive**(.) service primitive in Section 4.2.

Normally, an SDP session shall not terminate by a RemDev. Yet, such an event can indeed occur, either having the RemDev gracefully terminating the SDP session, using the L2CAP connection termination PDU, or abnormally terminating the SDP by stopping responding to SDP requests or L2CAP signalling commands. Such an event may be an indication of an exceptional condition that SDP client/server implementers should consider addressing for the smooth execution of this profile. If a termination event initiates from a RemDev, an SDP client may want to consider clearing any information obtained by this RemDev. Such an exceptional event may imply that the SDP server has (or is about to) shut-down, in which case any service information retrieved from this server should automatically become stale.

7 LINK MANAGER

7.1 CAPABILITY OVERVIEW

In this section, the LMP layer is discussed. In the table below, all LMP features are listed. The table shows which LMP features are mandatory to support with respect to this service discovery profile, which are optional and which are excluded. The reason for excluding features is that they may degrade operation of devices in this use case. Therefore, these features shall never be activated by a unit active in this use case.

If any of the rules stated below are violated, the units shall behave as defined in Section 7.2.

Traffic generated during service discovery interactions has no particular QoS requirements. As such, no particular provision of the Bluetooth link is required to support this profile.

	LM Procedure	Support in LMP	Support in LocDev	Support in RemDev
1.	Authentication	M	C1	C1
2.	Pairing	M		
3.	Change link key	M		
4.	Change the current link key	M		
4.	Encryption	O	C1	C1
5.	Clock offset request	M		
6.	Timing accuracy information request	O		
7.	LMP version	M		
8.	Supported features	M		
9.	Switch of master slave role	O		
10.	Name request	M		
11.	Detach	M		
12.	Hold mode	O		
13.	Sniff mode	O		
14.	Park mode	O		
15.	Power control	O		

Table 7.1: LMP procedures

	LM Procedure	Support in LMP	Support in LocDev	Support in RemDev
16.	Channel quality driven DM/DH	O		
17.	Quality of service	M		
18.	SCO links	O	X1	X1
19.	Control of multi-slot packets	M		
20.	Concluding parameter negotiation	M		
21.	Host connection	M		
Comments:				
[C1] No authentication or encryption is required specifically by this profile. This profile will, however, not attempt to change the existing operational settings for these procedures. Nevertheless, when this profile is executed all by itself, the default operational settings are: - authentication: no active - encryption: no active In the latter case, a LocDev will always comply with the security requirements imposed by a RemDev. If it cannot comply, it will bypass the RemDev. [X1]: This feature is not used in this profile, but its use by other applications running simultaneously with this profile is not excluded.				

Table 7.1: LMP procedures

7.2 ERROR BEHAVIOR

If a unit tries to use a mandatory feature, and the other unit replies that it is not supported, the initiating unit shall send an LMP_detach PDU with detach reason "unsupported LMP feature."

A unit shall always be able to handle the rejection of the request for an optional feature.

7.3 LINK POLICY

There are no fixed master-slave roles for the execution of this profile.

This profile does not state any requirements on which low-power modes to use, or when to use them. It is up to the Link Manager of each device to decide and request special link features as seen appropriate.

8 LINK CONTROL

8.1 CAPABILITY OVERVIEW

The following table lists all features on the LC level

	Procedure	Support in baseband	Support in LocDev	Support in RemDev
1.	Inquiry	M	C1	
2.	Inquiry scan	M		C2
3.	Paging	M	C1	
4.	Page scan			
A	Type R0	M		C3
B	Type R1	M		C3
C	Type R2	M		C3
5.	Packet types			
A	ID packet	M		
B	NULL packet	M		
C	POLL packet	M		
D	FHS packet	M		
E	DM1 packet	M		
F	DH1 packet	M		
G	DM3 packet	O		
H	DH3 packet	O		
I	DM5 packet	O		
J	DH5 packet	O		
K	AUX packet	M	X1	X1
L	HV1 packet	M	X1	X1
M	HV2 packet	O	X1	X1
N	HV3 packet	O	X1	X1
O	DV packet	M	X1	X1
6.	Inter-piconet capabilities	O		
7.	Voice codec			

Table 8.1: LC features

	Procedure	Support in baseband	Support in LocDev	Support in RemDev
A	A-law	O	X1	X1
B	µ-law	O	X1	X1
C	CVSD	O	X1	X1
Comments:				
[C1]: This mandatory LC feature will be activated under the control of the SrvDscApp.				
[C2]: This mandatory LC feature is a settable device policy (outside the scope of this profile) that is activated whenever a device is to operate in a discoverable (public) mode.				
[C3] This mandatory LC feature is a settable device policy (outside the scope of this profile) that is activated whenever a device is to operate in a discoverable or connectable (private) mode.				
[X1]: These features are not used in this profile, but their use by other applications running simultaneously with this profile is not excluded.				

Table 8.1: LC features

For the next four subsections, it is assumed that a LocDev is to perform service searches with originally unconnected RemDevs. It thus needs to inquire for and page (or only page) these RemDevs. None of the following four subsections apply whenever a LocDev performs service searches with RemDevs to which it is already connected.

8.2 INQUIRY

Whenever instructed by the SrvDscApp, the LocDev shall advise its baseband to enter the inquiry state. Entry into this state may or may not be immediate, however, depending on QoS requirements of any already existing and ongoing connections.

The user of the SrvDscApp shall be able to set the criteria for the duration of an inquiry, see *stopRule* service primitive parameter in Section 4.2. Nevertheless, the actual residence time in the inquiry state must comply with the recommendation given in section 10.7.3 of Bluetooth Baseband Specification [1].

When inquiry is invoked in a LocDev, the general inquiry procedure shall be used using a GIAC as described in Section 6.1 of Bluetooth GAP_profile:[3].

Instead of a GIAC, an appropriate DIAC can be used to narrow down the scope of the inquiry. Since the only defined DIAC (referred to as the LIAC) does not reflect any specific device or service categories, the use of DIACs is of limited (but non-zero) benefit in this profile. In particular, the profile does not exclude (but neither does it encourage) p erforming inquiries according to the limited inquiry procedure described in Section 6.2 of GAP_profile:[3].The information contained in the Class of Device field in the FHS packet returned by the ‘inquired devices’ can be used as a filter to limit the number of devices to page and connect to for subsequent SDP transactions.

8.3 INQUIRY SCAN

Inquiry scans are device-dependent policies outside the scope of this profile. Devices that operate in a discoverable mode of operation, see Section 4.1 of GAP_profile:[3], could be discovered by inquiries sent by other devices.

To be discovered by an inquiry resulting from a SrvDscApp action, a RemDev must enter inquiry scans using the GIAC; see general discoverable mode in Section 4.1.3 of GAP_profile:[3]. A DIAC can be used instead of a GIAC. As previously mentioned, the use of DIACs are of limited (but non-zero) benefit in this profile. In particular, performing inquiry scans according to the limited discoverable procedure described in Section 6.2 of GAP_profile:[3] is not excluded, but is not encouraged either.

8.4 PAGING

Whenever the SrvDscApp needs to connect to a specific RemDev for inquiring about its service records, the LocDev will advise its baseband to enter the page state. Entry into this state may or may not be immediate, however, depending on QoS requirements of any already existing and ongoing connections.

Depending on the paging class (R0, R1, or R2) indicated by a RemDev device, the LocDev shall page accordingly. The total residence time in the page state must comply with the recommendation given in section 10.6.3 of BT_BB_spec:[1]. For the pages, the 48-bit BD_ADDR of the RemDev must be used.

8.5 PAGE SCAN

Just like inquiry scans, page scans are device-dependent policies outside the scope of this profile. Devices that operate in a connectable mode of operation, see Section 4.2.2 of GAP_profile:[3], could establish Bluetooth links with other devices from pages sent by these other devices. To establish a link with a RemDev, a LocDev must send a page that results from a SrvDscApp action using the RemDev's 48-bit BD_ADDR.

8.6 ERROR BEHAVIOR

Since most features on the LC level have to be activated by LMP procedures, errors will usually be caught at that layer. However, there are some LC procedures that are independent of the LMP layer, such as inquiry or paging. Misuse of such features is difficult or sometimes impossible to detect. There is no mechanism defined to detect or prevent such improper use.

9 REFERENCES

9.1 NORMATIVE REFERENCES

[1] Baseband specification (see Volume 1, Part B)

[2] Bluetooth Assigned Numbers
http://www.bluetooth.org/assigned-numbers.htm

[3] Generic Access Profile (see Volume 2, Part K1)

[4] Link Manager Protocol (see Volume 1, Part C)

[5] Logical Link Control and Adaptation Protocol Specification (see Volume 1, Part D)

[6] Message Sequence Charts between Host–Host Controller/Link Manager (see Volume 1, Appendix IX)

[7] Service Discovery Protocol (see Volume 1, Part E)

10 DEFINITIONS

Term	Definition
conscious	(usually referred to) a process that requires the explicit intervention of a user to be accomplished
known	(with respect to a specific device) opposite to *unknown*; a known devices is not necessarily a *paired* device
new (RemDev)	(with regard to this profile) an additional remote device (RemDev) that is discovered during a Bluetooth inquiry, and that is not already connected to local device (LocDev)
private	a mode of operation whereby a device can only be found via Bluetooth baseband pages; i.e. it only enters page scans
public	a mode of operation whereby a device can be found via Bluetooth baseband inquiries; i.e. it enters into inquiry scans. A public device also enters into page scans (contrast this with *private*)
unconscious	opposite to *conscious*
unknown	(with respect to a specific device) any other device that a specific device has no record of

11 APPENDIX A (INFORMATIVE): SERVICE PRIMITIVES AND THE BLUETOOTH PDUs

In this Annex, we relate the service primitives shown in section 4.2 with the various Bluetooth PDUs which support these primitives. The table below only shows the actions taken at the higher involved Bluetooth layer. Thus, unless specifically stated, the low-level inquiries and pages needed to discover and connect to Bluetooth devices are not discussed in detail.

service primitive	(highest layer) Bluetooth PDUs involved
serviceBrowse (LIST(*RemDev*) LIST(*RemDevRelation*) LIST(*browseGroup*) *getRemDevName* *stopRule*)	For the subset of *RemDev* that satisfy the *RemDevRelation*, this service primitive will cause the LocDev to send: an *SDP_ServiceSearchRequest* PDU and receives a corresponding response PDU, see section 4.5 in BT_SDP_spec:[7]; an *SDP_ServiceAttributeRequest* PDU and receives a corresponding response PDU, see section 4.6 in BT_SDP_spec:[7]. The first transaction above identifies the SDP servers that contain pertinent service records, while the second transaction retrieves the desired information; Alternatively, the two transactions above are combined to one: LocDev sends an *SDP_ServiceSearchAttributeRequest* PDU and receives a corresponding response PDU, see section 4.7 in BT_SDP_spec:[7] In either of the above cases, the corresponding SDP transaction may last a number of request and response PDU exchanges, due to the L2CAP MTU limitation. If the *getRemDevName* parameter is set to 'yes', then for each RemDev involved in the execution of this service primitive, the service primitive will cause a sequence of *LMP_name_request*() LM level PDUs to be sent by the LocDev.* The corresponding RemDev responds with a *LMP_name_response*() LM level PDU containing the requested user-friendly device name.
serviceSearch (LIST(*RemDev*) LIST(*RemDevRelation*) LIST(*searchPattern*, *attributeList*) *getRemDevName* *stopRule*)	same as above

Table 11.1: Bluetooth PDUs related to the service primitives in Section 4.2

service primitive	(highest layer) Bluetooth PDUs involved
enumerateRemDev (LIST(*classOfDevice*) *stopRule*)	This service primitive will cause a Bluetooth baseband *inquiry* process. The inquiry will 'indiscriminately†' find devices residing in the vicinity of the LocDev. Prior to returning the results of this inquiry the LocDev may filter them using the classOfDevice qualifier.
terminatePrimitive (*primitiveHandle* *returnResults*)	This service primitive will cause the termination of any outstanding operation caused by the invocation of the service primitive identified by the *primitiveHandle* parameter. This may cause an L2CAP connection termination request PDU to be sent from the LocDev to the RemDev, and the subsequent transmission of an L2CAP termination response PDU. It the LocDev is connecting to the RemDev only for the purposes of an SDP transaction, the baseband link will also be severed by the transmission of an LMP_detach LM level PDU.

Table 11.1: Bluetooth PDUs related to the service primitives in Section 4.2

*. If the information requested is already stored (cached) in the LocDev, this service primitive may not have to cause the described LM level PDU transaction.

†. The inquiries considered here use the GIAC. No CoD-specific DIACs have been defined. Nevertheless, the use of appropriate DIACs whenever possible is not excluded and is not outside the scope of this profile.

Part K:3

CORDLESS TELEPHONY PROFILE

This profile defines the features and procedures that are required for interoperability between different units active in the ‘3-in-1 phone’ use case. The scope of this profile includes the following layers/protocols/ profiles: Bluetooth Baseband, Link Manager Protocol, L2CAP, Service Discovery Protocol, Telephony Control Protocol Specification (TCS-Binary) and the General Access Profile.

CONTENTS

1 Introduction 100
1.1 Scope 100
1.2 Profile Dependencies 100
1.3 Symbols and conventions 101
1.3.1 Requirement status symbols 101
1.3.2 Signalling diagram conventions 102
1.3.3 Notation for timers and counters 102
2 Profile overview 103
2.1 Profile stack 103
2.2 Configurations and roles 104
2.3 User requirements and scenarios 105
2.4 Profile fundamentals 106
2.5 Feature definitions 106
2.6 Conformance 107
3 Application layer 108
4 TCS-BIN procedures 110
4.1 Connection Management 110
4.1.1 Connecting to a GW 110
4.1.2 Connecting to another TL 110
4.2 Call Control procedures 111
4.2.1 Sides 111
4.2.2 Call class 111
4.2.3 Call request 111
4.2.4 Overlap sending 111
4.2.5 Call proceeding 111
4.2.6 Call confirmation 111
4.2.7 Call connection 112
4.2.8 Non-selected user clearing 112
4.2.9 In-band tones and announcements 112
4.2.10 Failure of call establishment 112
4.2.11 Call clearing 112
4.2.12 Call information 113
4.3 Supplementary services 113
4.3.1 DTMF signalling 113
4.3.2 Calling line identity 113
4.3.3 Register recall 113

4.4 Group Management procedures 114
4.4.1 Obtain Access Rights 114
4.4.2 Configuration distribution 114
4.4.2.1 Link loss detection by GW 114
4.4.3 Periodic key update 115
4.4.4 Fast inter-member access 115
4.5 Connectionless procedures 115
4.6 TCS-BIN Message overview 116
4.7 Information Element overview 117
4.7.1 Bearer capability 118
4.7.2 Called party number 118
4.7.3 Calling party number 118
4.7.4 Cause 119
4.8 Link loss 119
5 Service Discovery procedures 120
6 L2CAP procedures 121
6.1 Channel types 121
6.2 Configuration options 121
6.2.1 Maximum Transmission unit 121
6.2.2 Flush timeout option 121
6.2.3 Quality of Service 121
7 LMP procedures overview 122
7.1 Master-slave switch 123
7.2 Link policy 123
7.3 Encryption key size 123
8 LC features 124
8.1 Inquiry scan 125
8.2 Inter-piconet capabilities 125
9 General Access Profile Interoperability Requirements 126
9.1 Modes 126
9.2 Security aspects 126
9.3 Idle mode procedures 127
9.3.1 Bonding 127

10 Annex A (Informative): Signalling flows 128
10.1 Outgoing external call without post-dialling....128
10.2 Outgoing external call with post-dialling....129
10.3 Incoming external call, SETUP delivered on connectionless channel 130
10.4 Incoming external call, SETUP delivered on connection-oriented channel 130
10.5 Call Clearing....131
10.6 DTMF signalling 131
10.7 DTMF signalling failure 131
10.8 Access rights request....132
10.9 Configuration distribution 132
10.10 Periodic key update 133
10.11 Fast inter-member access....133
11 Timers and counters....135
12 References....136
13 List of Figures....137
14 List of Tables 138

1 INTRODUCTION

1.1 SCOPE

The Cordless Telephony profile defines the protocols and procedures that shall be used by devices implementing the use case called ‘3-in-1 phone’.

The ‘3-in-1 phone’ is a solution for providing an extra mode of operation to cellular phones, using Bluetooth as a short-range bearer for accessing fixed network telephony services via a base station. However, the 3-in-1 phone use case can also be applied generally for wireless telephony in a residential or small office environment, for example for cordless-only telephony or cordless telephony services in a PC – hence the profile name ‘Cordless Telephony’.

This use case includes making calls via the base station, making direct intercom calls between two terminals, and accessing supplementary services provided by the external network.

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure. A profile has dependencies on the profile(s) in which it is contained – directly and indirectly. As indicated in the figure, the Cordless Telephony profile is dependent only upon the Generic access profile. The terminology, user interface and security aspects, modes and procedures as defined in the Generic access profile are applicable to this profile, unless explicitly stated otherwise.

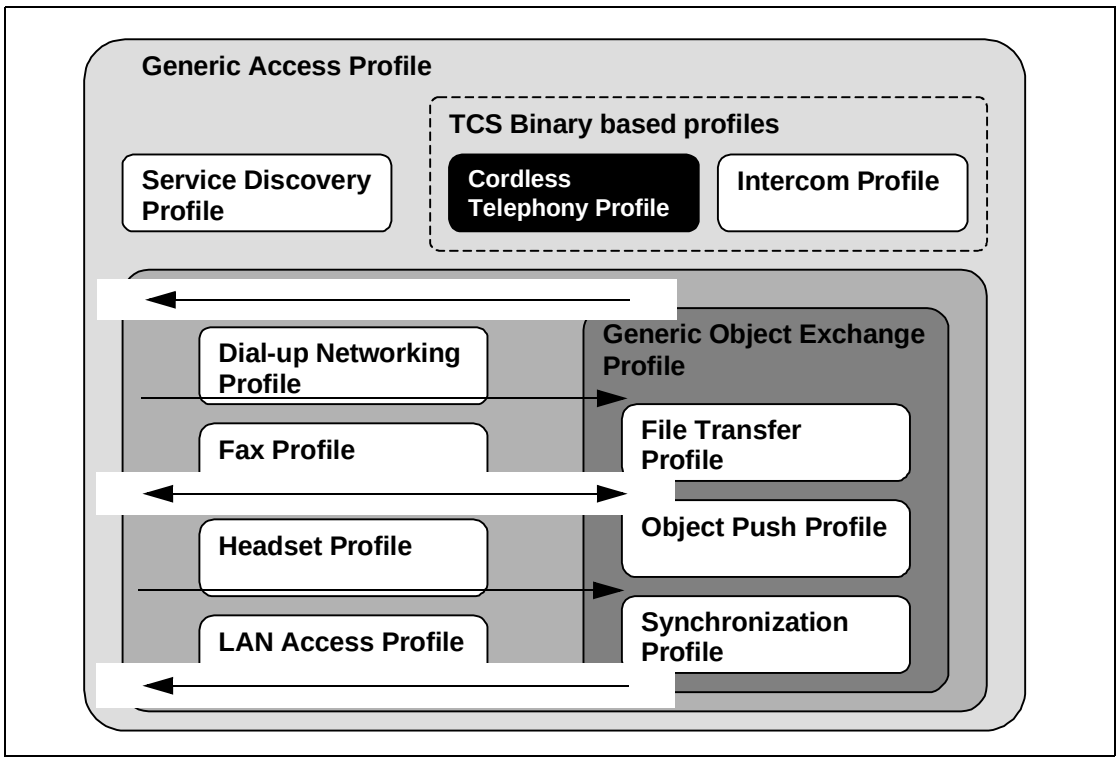


Figure 1.1: Bluetooth Profiles

1.3

1.3.1 Requirement status symbols

In this document, the following symbols are used:

'M' for mandatory to support (used for capabilities that shall be used in the profile);

'O' for optional to support (used for capabilities that can be used in the profile);

'C' for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

'X' for excluded (used for capabilities that may be supported by the unit, but which shall never be used in the profile);

'N/A' for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:

A		B
	PROC1	
	PROC2	
	PROC3	
	(PROC4)	
	(PROC5)	
	MSG1	
	MSG2	
	(MSG3)	
	(MSG4)	

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A and B). PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates an optional message from B to A.

1.3.3 Notation for timers and counters

Timers and counters may be introduced specific to this profile. To distinguish them from timers (counters) used in the Bluetooth protocol specifications and other profiles, these timers (counters) are named in the following format: 'T_{CTP-}*nnn*' ('N_{CTP}*nnn*').

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 below shows the protocols as used within this profile:

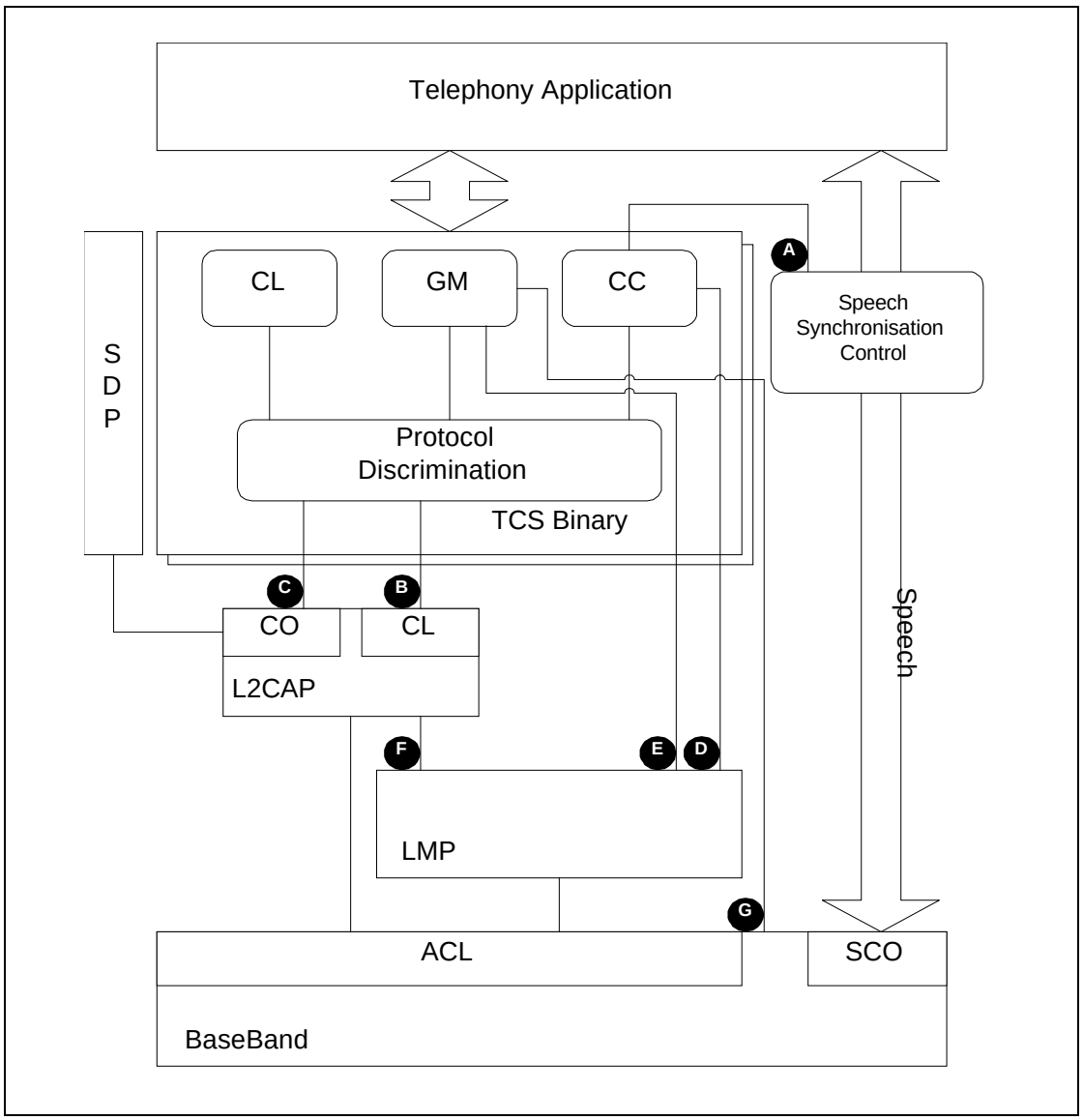


Figure 2.1: Protocol model

This profile will define the requirements for each of the layers in the model above for the Cordless Telephony profile.

In the profile, the interfaces in Figure 2.1 above are used for the following purposes:

A) The Call Control entity uses this interface to the speech synchronization control to connect and disconnect the internal speech paths.

B) This interface is used by the GW to send and by the TL to receive broadcast TCS-Binary messages.

C) This interface is used to deliver all TCS messages that are sent on a connection-oriented (point-to-point) L2CAP channel.

D) This interface is used by the Call Control entity to control the Link Manager directly for the purpose of establishing and releasing SCO links.

E) This interface is used by the Group Management to control Link Manager functions when initializing and for key handling purposes.

F) This interface is not within the scope of this profile.

G) This interface is used by the Group Management entity to control the LC/ Baseband directly to enable inquiry, paging, inquiry scan and page scan.

2.2 CONFIGURATIONS AND ROLES

The following two roles are defined for this profile:

Gateway (GW) – The GW acts as a terminal endpoint from the external network point of view and handles all interworking towards that network. The GW is the central point with respect to external calls, which means that it handles all call set-up requests to/from the external network. Examples of devices that can act as a gateway include a PSTN home base station, an ISDN home base station, a GSM gateway, a satellite gateway and an H.323 gateway.

With respect to this profile, the gateway may have the functionality to support multiple terminals being active at once, or be of a simple kind where only one terminal may be active. The simple gateway will not support multiple ringing terminals, multiple active calls or services involving more than one terminal simultaneously.

Terminal (TL) – The TL is the wireless user terminal, which may for example be a cordless telephone, a dual-mode cellular/cordless phone or a PC. Note that the scope of this profile with respect to a dual-mode cellular/cordless phone acting as TL is only the cordless mode.

The Cordless Telephony profile supports a topology of one gateway (GW) and a small number (≤7) of terminals (TLs)[1]. Figure 2.2 below shows an example of the considered architecture:

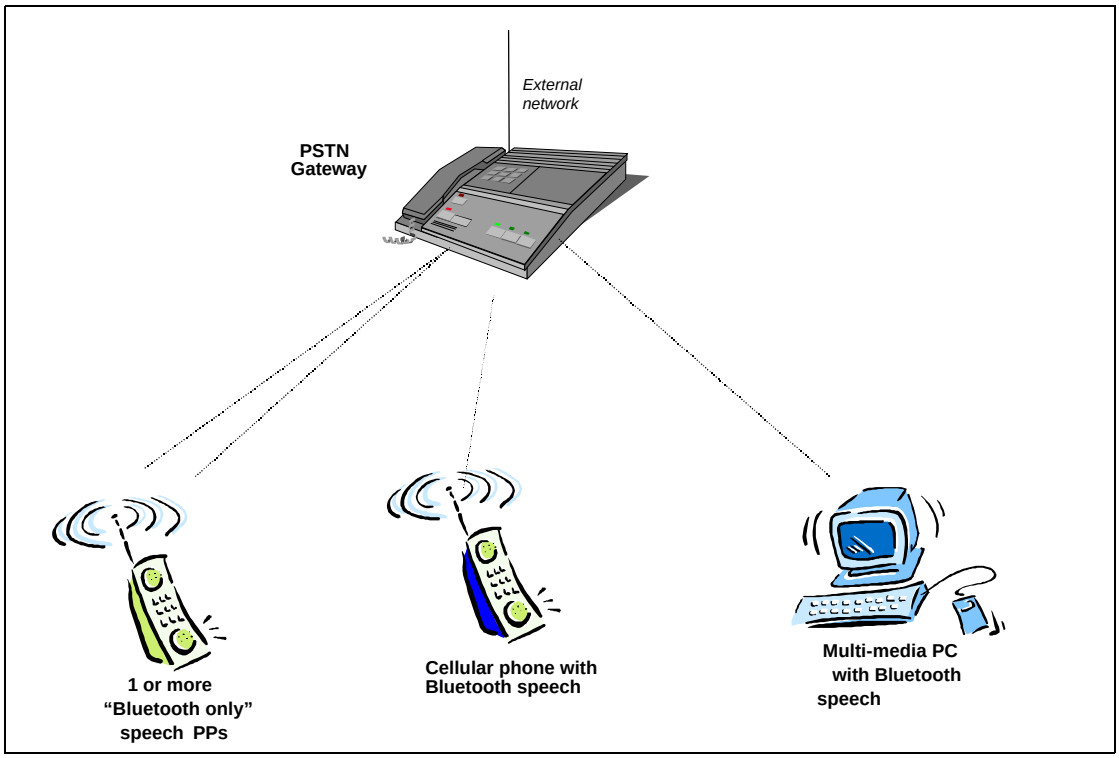


Figure 2.2: System configuration example

2.3 USER REQUIREMENTS AND SCENARIOS

The following scenarios are covered by this profile:

1. Connecting to the gateway so that incoming calls can be routed to the TL and outgoing calls can be originated.
2. Making a call from a TL to a user on the network that the gateway is connected to.
3. Receiving a call from the network that the gateway is connected to.
4. Making direct calls between two terminals.
5. Using supplementary services provided by the external network by means of DTMF signalling and register recall (hook flash).

1. Optionally, more terminals may be supported.

2.4 PROFILE FUNDAMENTALS

The GW is normally the master of the piconet in the Cordless Telephony profile. As master, the GW will control the power mode of the TLs and may broadcast information to the TLs.

A TL that is out of range of a GW searches for it by periodically trying to page it. A GW shall devote as much of its free capacity as possible (considering power limitations and ongoing signalling) to page scanning in order to allow roaming TLs that enter the range of the GW to find it as quickly as possible. This scheme minimizes 'air pollution' and gives reasonable access time when coming into range of the GW. When a TL has successfully paged a GW, a master-slave switch shall be performed since the GW shall be the master. A connection-oriented L2CAP channel and, possibly, a L2CAP connectionless channel are established to be used for all TCS signalling during that Cordless Telephony session.

A TL that is within range of a GW shall normally be in park mode when it is not engaged in calls. This mode is power-efficient, allows for reasonable call set-up times and allows broadcasting to the attached TLs.

Upon arrival of an incoming call, or when a TL wants to make an outgoing call, the GW shall be put in active mode. The L2CAP channels (see above) are used for all TCS control signalling. Voice is transported using SCO links.

For security purposes, authentication of TLs and GW is used, and all user data is encrypted. To facilitate secure communication between cordless units, the WUG concept (see TCS Binary, Section 3) is used. The GW always acts as WUG master.

2.5 FEATURE DEFINITIONS

Calling line identification presentation (CLIP) – The ability to provide the calling party number to the called party before accepting the call.

Call information – The ability to provide additional information during the active phase of a call.

Connection Management – The ability to accept and (TLs only) request connections for the purposes of TCS-Bin procedures.

DTMF signalling – The ability, in external calls, to send a DTMF signal over the external network to the other party.

Incoming external call – A call originating from the external network connected to the GW.

Initialization – The infrequent process whereby a TL receives access rights to a certain GW.

Intercom call – A call originating from a TL towards another TL.

Multi-terminal support –

1. In the GW, the ability to handle multiple active terminals being registered at the same time[2]
2. In the TL, the support for a Wireless User Group (WUG)

On hook – The ability to indicate the action of going on-hook (e.g. to terminate a call), and release of all radio resources related to that call.

Outgoing external call – A call originated by a TL towards the external network connected to the GW.

Post-dialling – The ability to send dialling information after the outgoing call request set-up message is sent.

Register recall – The ability of the TL to request 'register recall', and of the GW to transmit the request to the local network. Register recall means to seize a register (with dial tone) to permit input of further digits or other actions. In some markets, this is referred to as 'hook flash'.

2.6 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated as mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

Note that the Intercom Profile is used for intercom calls. This means that a TL claiming conformance to the Cordless Telephony profile must conform to Intercom Profile.

2. Note that a GW may support multiple active terminals but not a Wireless User Group (WUG).

3 APPLICATION LAYER

The following text, together with the associated sub-clauses, defines the feature requirements with regard to this profile.

Table 3.1 shows the feature requirements made by this profile.

Item no.	Feature	Support in TL	Support in GW
1.	Connection Management	M	M
2.	Outgoing external call	M	M
3.	Incoming external call	M	M
4.	Intercom call	M	N/A
5.	On hook	M	M
6.	Post-dialling	O	O
7.	Multi-terminal support	O	O
8.	Call information	O	O
9.	Calling line identification presentation (CLIP)	M	O
10.	DTMF signalling	M	M
11.	Register recall	M	M

Table 3.1: Application layer features

Table 3.2 maps each feature to the procedures used for that feature, and shows if the procedure is optional, mandatory or conditional for that feature. The procedures are described in the referenced section.

Feature	Procedure	Ref.	Support in TL	Support in GW
1. Connection Management	Connecting to a GW	4.1.1	M	M
	Connecting to a TL	4.1.2	M	N/A
2. Outgoing external call	Call request	4.2.3	M	M
	Overlap sending	4.2.4	C2	C2
	Call proceeding	4.2.5	C2	C2
	Call confirmation	4.2.6	M	O
	Call connection	4.2.7	M	M
	In-band tones and announcements	4.2.9	M	O

Table 3.2: Application layer feature to procedure mapping

Feature	Procedure	Ref.	Support in TL	Support in GW
3. Incoming external call	Call request	4.2.3	M	M
	Call confirmation	4.2.6	M	M
	Call connection	4.2.7	M	M
	Non-selected user clearing	4.2.8	M	M
	In-band tones and announcements	4.2.9	M	O
4. Intercom call	NOTE 1			
5. On hook	Call clearing	4.2.11	M	M
6. Post-dialling	Overlap sending	4.2.4	M	M
	Call proceeding	4.2.5	M	M
7. Multi-terminal support	Obtain access rights	4.4.1	M	O
	Configuration distri-bution	4.4.1	M	O
	Fast inter-member access	4.4.4	M	O
	Periodic key update	4.4.3	M	O
8. Call information	Call information	4.2.12	M	M
9. Calling line identification presentation (CLIP)	Calling line identity	4.3.2	M	M
10. DTMF signalling	DTMF signalling	4.3.1	M	M
11. Register recall	Register recall	4.3.3	M	M
C2: IF feature 6 THEN M else N/A				

Table 3.2: Application layer feature to procedure mapping

Note 1: For intercom calls, the intercom profile is used. Before initiating the intercom call, the TL which is initiating the call may optionally use the fast inter-member access procedure to speed up the call set-up.

4 TCS-BIN PROCEDURES

The following text together with the associated sub-clauses defines mandatory requirements with regard to this profile.

When describing TCS-BIN procedures, this section provides additional information concerning lower layer handling. The normative reference for TCS-BIN procedures is TCS Binary.

Annex A contains signalling flows that illustrate the procedures in this section.

4.1 CONNECTION MANAGEMENT

4.1.1 Connecting to a GW

When a TL connects to the GW, the link is configured and the L2CAP connection that is used for further signalling during that TCS-BIN session is set up and configured. The TL which is connecting is responsible for setting up the connection-oriented L2CAP channel.

Only trusted TLs are allowed to connect to the GW.

Note that, in order to avoid the paging delay at call set-up and to enable broadcasted messages, the TL establishes a L2CAP connection to the GW when it comes into range, and not before every call. This L2CAP connection remains until the radio link is lost or the TL stops being active in this profile. This means that the L2CAP connections used may be idle (i.e. not used to transfer data) for long periods of time.

A GW supporting feature 7, ‘Multi-terminal support’, uses a connectionless L2CAP channel for TCS-BIN broadcasted messages. A TL is added to the connectionless group when it connects to the GW.

4.1.2 Connecting to another TL

In the case of an intercom call, the TL which initiates the call establishes a direct link to the other TL. See the Intercom Profile for a description of these procedures.

If the TL has the capability to participate in two piconets at the same time, the TL may remain a member of the GW piconet and participate in signalling towards the GW during the intercom call.

If the TL does not have the capability to participate in two piconets at the same time, it must detach from the GW while the intercom call is active. After the intercom call is finished, the TL must re-establish the connection to the GW.

4.2 CALL CONTROL PROCEDURES

4.2.1 Sides

This section describes which sides shall be assumed for the purpose of reading TCS Binary.

In an outgoing external call, the TL is the outgoing side and the GW is the incoming side. In an incoming external call, the TL which terminates the call is the incoming side and the GW is the outgoing side.

Refer to the Intercom Profile for the sides assumed in intercom calls.

4.2.2 Call class

This section describes the usage of call classes in the Cordless Telephony profile.

An *external call* is a call between a TL and a third party connected via an external network (PSTN, ISDN, GSM or other). The call class used in SETUP messages for external calls (outgoing and incoming) is ‘external call’.

An *intercom call* is a call between two TLs, which may be setup with GW support if the two TLs are members of the same WUG. Refer to Intercom Profile for call class usage in intercom calls.

4.2.3 Call request

This procedure shall be performed as defined in TCS Binary.

4.2.4 Overlap sending

This procedure shall be performed as defined in TCS Binary.

4.2.5 Call proceeding

This procedure shall be performed as defined in TCS Binary.

4.2.6 Call confirmation

This procedure shall be performed as defined in TCS Binary.

If the call is an incoming external call, and the SETUP message was delivered on a connection-oriented channel, the incoming side must acknowledge the SETUP message by performing the call confirmation procedure.

4.2.7 Call connection

This procedure shall be performed as defined in TCS Binary. The following text defines the mandatory requirements with regard to this profile.

If the bearer capability for this call is ‘Synchronous Connection-Oriented’, the SCO link establishment sub-procedure (see LMP, Section 3.21) shall be initiated before sending a CONNECT.

If the bearer capability for this call is ‘Synchronous Connection-Oriented’, the audio path shall be connected to by a unit when it receives a CONNECT or CONNECT ACKNOWLEDGE.

4.2.8 Non-selected user clearing

This procedure shall be performed as defined in TCS Binary. Additionally, the text in 4.2.11 defines the mandatory requirements with regard to this profile concerning call clearing.

4.2.9 In-band tones and announcements

This procedure shall be performed as defined in TCS Binary. The following text defines the mandatory requirements with regard to this profile.

Only the GW may provide in-band tones and announcements. The SCO link establishment sub-procedure (see Link Manager Protocol, Section 3.21) is initiated before sending a Progress Indicator information element #8, “In-band information or appropriate pattern is now available”.

The audio path shall be connected to by a TL when it receives a Progress Indicator information element #8, “In-band information or appropriate pattern is now available”.

4.2.10 Failure of call establishment

This procedure shall be performed as defined in TCS Binary. Additionally, the text in 4.2.11 defines the mandatory requirements with regard to this profile concerning call clearing.

4.2.11 Call clearing

All call clearing and call collision procedures as defined in TCS Binary shall be supported by both GW and TL. For a specification of the complete behavior, see TCS Binary. This section describes how the lower layers are used to release circuit switched (SCO) connections.

A unit shall release the SCO link by invoking the appropriate LMP sub-procedure (see Link Manager Protocol, Section 3.21) when a unit has received a RELEASE message.

A unit shall release the SCO link (if not already released) by invoking the appropriate LMP sub-procedure (see Link Manager Protocol, Section 3.21) when it has received a RELEASE COMPLETE message.

4.2.12 Call information

This procedure shall be performed as defined in TCS Binary.

4.3 SUPPLEMENTARY SERVICES

Supplementary services can be either internal services within the WUG, or external services provided by the network the GW is connected to.

The exact set of external supplementary services is not defined in this profile and is dependent on the network the GW is connected to. This profile provides the means for accessing them; for example through the use of DTMF signalling and register recall.

The required support for internal services and DTMF signalling is defined in the following sub-clauses.

4.3.1 DTMF signalling

The capability to request DTMF signalling towards the external network is mandatory for the TL. The capability to accept DTMF signalling requests is mandatory for the GW.

Depending on the network the GW is connected to, it shall translate the DTMF messages to the appropriate in-band or out-of-band signalling. If the network has no DTMF signalling capability, or if the GW for some reason is unable to perform DTMF signalling towards the external network, the GW shall reject the request for DTMF signalling as described below. In the START DTMF REJECT message, the GW shall use Cause #29, “Facilities rejected”.

4.3.2 Calling line identity

This procedure shall be performed as defined in TCS Binary.

It is recommended that all GWs that are connected to networks that provide calling line identity have the capability to provide this information to the user.

4.3.3 Register recall

This procedure shall be performed as defined in TCS Binary.

4.4 GROUP MANAGEMENT PROCEDURES

4.4.1 Obtain Access Rights

This procedure shall be performed as defined in TCS Binary.

A TL which wants to become member of a WUG may initiate this procedure towards a GW. The GW may accept or reject the request depending, for example, on configuration, or if the user has physical access to the base.

A GW which accepts the access rights request shall add the TL to the WUG and initiate the Configuration distribution procedure.

4.4.2 Configuration distribution

This procedure shall be performed as defined in TCS Binary.

Because of the security implications of this procedure, a TL is not forced to store the key information received during this procedure. In addition, GW may always reject the ACCESS RIGHTS REQUEST from a TL because of implementation-dependent reasons. For example, the user may be required to press a button on the GW before being granted access to the group.

Note that for intercom calls, two TLs that are members of the WUG do not need to perform the initialization procedure described in the Intercom profile (see Intercom Profile) if they use the keys distributed in the Configuration distribution procedure.

A TL which stores link keys during the Configuration Distribution procedure shall never overwrite existing link keys to other WUG members. Only if there was previously no link key to a specific device shall the key obtained during the Configuration Distribution procedure be used.

In addition to the link-loss handling described in Section 4.8, Section 4.4.2.1 applies for this procedure.

4.4.2.1 Link loss detection by GW

If the GW detects loss of link before receiving the INFO ACCEPT message, it shall consider the WUG update to be terminated unsuccessfully and consider the TL detached. If the GW detects loss of link after receiving the INFO ACCEPT message, it shall consider the WUG update to be terminated successfully.

4.4.3 Periodic key update

The K_{master} to be used during a GW-TL connection is issued to the TL when connecting to a GW. The K_{master} is intended to be a key valid for a single session only, but since the GW piconet is operational all the time, this would mean that the same K_{master} would always be used. In order to increase the security level, the K_{master} is changed periodically.

Timer $T_{CTP}400$ determines the interval between key changes. When $T_{CTP}400$ expires, the GW tries to do a periodic key update on all TLs. However, some TLs may be out of range or powered off, or the procedure may fail for some other reason. The new key in these cases is given to the TL when it attaches the next time. After there has been an attempt to update all TLs, $T_{CTP}400$ is reset.

The periodic key update for one TL is performed as follows. First, if the TL was parked, it is unparked. Then, the new link key is issued. After this, the new link key is activated by turning encryption off and back on. Finally, the TL may be parked.

If any of the sub-procedures fails, further sub-procedures will not be performed on that TL. The GW shall proceed with updating the next TL.

4.4.4 Fast inter-member access

The Fast inter-member access procedure is used when two TLs that are members of the same WUG need to establish a piconet of their own. This may be needed when an intercom call shall be established. Refer to TCS Binary for a definition of the procedure.

The TL_T may detach from the GW after having sent the LISTEN ACCEPT message by terminating the L2CAP channel to the GW and sending a LMP_detach.

The TL_I may detach from the GW after having received the LISTEN ACCEPT message by terminating the L2CAP channel to the GW and sending a LMP_detach.

4.5 CONNECTIONLESS PROCEDURES

TCS-BIN Connectionless (CL) messaging is not within the scope of the Cordless Telephony profile.

4.6 TCS-BIN MESSAGE OVERVIEW

This section defines the required TCS-BIN messages in the Cordless Telephony profile.

Message	Ability to Send		Ability to Receive	
	TL	GW	TL	GW
Access rights accept	N/A	O	C1	N/A
Access rights reject	N/A	O	C1	N/A
Access rights request	C1	N/A	N/A	O
Alerting	M	O	M	M
Call Proceeding	C2	C2	M	M
Connect	M	M	M	M
Connect Acknowledge	M	M	M	M
Disconnect	M	M	M	M
Info suggest	N/A	O	C1	N/A
Info accept	C1	N/A	N/A	O
Information	M	O	O	M
Listen request	C1	N/A	N/A	O
Listen suggest	N/A	O	C1	N/A
Listen accept	C1	O	C1	O
Listen reject	C1	O	C1	O
Progress	N/A	O	M	N/A
Release	M	M	M	M
Release Complete	M	M	M	M
Setup	M	M	M	M
Setup Acknowledge	N/A	O	O	N/A
Start DTMF	M	N/A	N/A	M
Start DTMF Acknowledge	N/A	M	M	N/A
Start DTMF Reject	N/A	M	M	N/A
Stop DTMF	M	N/A	N/A	M
Stop DTMF Acknowledge	N/A	M	M	N/A
C1: IF feature 7 THEN M else N/A C2: IF feature 6 THEN M else N/A				

Table 4.1: TCS-BIN messages

4.7 INFORMATION ELEMENT OVERVIEW

This section together with the associated sub-clauses defines the required information elements used in TCS-BIN messages in the Cordless Telephony profile.

Information Element	Ability to Send		Ability to Receive	
	TL	**GW**	**TL**	**GW**
Message type	M	M	M	M
Audio control	N/A	N/A	N/A	N/A
Bearer capability	M	M	M	M
Call class	M	M	M	M
Called party number	M	O	O	M
Calling party number	O	C2	M	O
Cause	M	M	M	M
Clock offset	C1	O	C1	O
Company-specific	O	O	O	O
Configuration data	N/A	O	C1	N/A
Destination CID	N/A	N/A	N/A	N/A
Keypad facility	M	N/A	N/A	M
Progress indicator	N/A	O	M	N/A
SCO handle	M	M	M	M
Sending complete	M	N/A	N/A	M
Signal	N/A	M	M	N/A
C1: IF feature 7 THEN M else N/A C2: IF feature 9 THEN M else N/A				

Table 4.2: TCS-BIN information elements

The following subsections define restrictions that apply to the contents of the TCS-BIN information elements in the CordlessTelephony profile. Note that in the tables, only fields where restrictions apply are shown. If a field is not shown in a table, it means that all values defined in TCS Binary for that field are allowed.

For those information elements not listed below, no restrictions apply.

4.7.1 Bearer capability

The following restrictions apply to the contents of the Bearer capability information element:

Field	Values allowed
Link type	SCO, None

Table 4.3: Restrictions to contents of Bearer capability information element

4.7.2 Called party number

Maximum information element length is 27 octets, thus allowing a maximum of 24 number digits.

4.7.3 Calling party number

Maximum information element length is 28 octets, thus allowing a maximum of 24 number digits.

4.7.4 Cause

The following restrictions apply to the contents of the Cause information element:

Field	Values allowed
Cause value	#1 – “Unassigned (unallocated number)” #3 – “No route to destination” #16 – “Normal call clearing” #17 – “User busy” #18 – “No user responding” #19 – “No answer from user (user alerted)” #21 – “Call rejected by user” #22 – “Number changed” #26 – “Non-selected user clearing” #28 – “Invalid number format (incomplete number” #29 – “Facilities rejected” #34 – “No circuit/channel available” #41 – “Temporary failure” #44 – “Requested circuit/channel not available” #58 – “Bearer capability not presently available” #65 – “Bearer capability not implemented” #69 – “Requested facility not implemented” #102 – “Recovery on timer expiry”

Table 4.4: Restrictions to contents of Cause information element

4.8 LINK LOSS

If a unit in a CC state other than *Null* detects loss of link, it shall immediately go to the *Null* state. Release procedures shall in this case not be performed.

A unit in any GM state which detects loss of link shall consider itself to be in the null state. Any ongoing GM procedure shall immediately be aborted and considered to be terminated unsuccessfully.

5 SERVICE DISCOVERY PROCEDURES

Table 5.1 below lists all entries in the SDP database of the GW defined by this profile. The 'Status' column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonic's used in the 'Value' column, and the codes assigned to the attribute identifiers, can be found in the Bluetooth Assigned Numbers (http://www.bluetooth.org/assigned-numbers.htm).

Item	Definition:	Type:	Value:	Status	Default
Service Class ID List				M	
Service Class #0		UUID	Cordless Telephony	M	
Service Class #1		UUID	Generic Telephony	O	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	TCS-BIN-CORDLESS	M	
Service Name	Displayable Text name	String	Service-provider defined	O	'Cordless Telephony'
External Network		UInt8	0x01=PSTN 0x02=ISDN 0x03=GSM 0x04=CDMA 0x05=Analogue cellular 0x06=Packet-switched 0x07=Other	O	
BluetoothProfileDescriptorList				M	
Profile #0		UUID	Cordless Telephony	M	
Parameter for Profile #0	Version	UInt16	0x0100*	O	0x100

Table 5.1: SDP entry for GW service

*. Indicating version 1.0

6 L2CAP PROCEDURES

The following text, together with the associated sub-clauses, define the mandatory requirements with regard to this profile.

6.1 CHANNEL TYPES

In this profile, both connection-oriented channels and connectionless channels are used.

Connectionless channels are used to broadcast information from the GW to the TLs. Only the GW shall use connectionless channels for sending. Refer to the Bluetooth Security Architecture White paper for information on the security implications of using L2CAP connectionless traffic.

In this profile, only the TL may initiate the establishment of connection-oriented channels. When connecting to the GW, the TL shall use the value 0x0007 (TCS-BIN-CORDLESS) in the PSM field of the Connection Request packet. For PSM usage in intercom calls, see Intercom Profile.

6.2 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the Cordless Telephony Profile.

6.2.1 Maximum Transmission unit

The minimum MTU that a L2CAP implementation used for this profile should support is 171 octets. This means that the maximum number of TLs supported by this profile is 7.

6.2.2 Flush timeout option

The flush timeout value used for both the GW and the TL shall be the default value of 0xFFFF.

6.2.3 Quality of Service

Negotiation of Quality of Service is optional.

7 LMP PROCEDURES OVERVIEW

In this section the LMP layer is discussed. In the table below, all LMP features are listed. In the table it is shown what LMP features are mandatory to support with respect to the Cordless Telephony profile, which are optional and which are excluded. The reason for excluding features is that they may degrade operation of devices in this profile. Therefore, these features shall never be activated by a unit active in this profile.

	Procedure	Support in LMP	Support in TL	Support in GW
1.	Authentication	M		
2.	Pairing	M		
3.	Change link key	M		
4.	Change the current link key	M		
5.	Encryption	O	M	M
6.	Clock offset request	M		
7.	Slot offset information	O		
8.	Timing accuracy information request	O		
9.	LMP version	M		
10.	Supported features	M		
11.	Switch of master slave role	O	M	C1
12.	Name request	M		
13.	Detach	M		
14.	Hold mode	O		
15.	Sniff mode	O		
16.	Park mode	O	M	C1
17.	Power control	O		
18.	Channel-quality driven DM/DH	O		
19.	Quality of service	M		
20.	SCO links	O	M	M
21.	Control of multi-slot packets	O		
22.	Paging scheme	O		
23.	Link supervision	M		
24.	Connection establishment	M		
C1: IF feature 7 THEN M else N/A				

Table 7.1: LMP procedures

7.1 MASTER-SLAVE SWITCH

A GW supporting feature 7, ‘Multi-terminal support’, must always be the master of the piconet. Such a GW will request a master-slave switch when a TL connects. If the TL rejects the request, the GW may detach it. Thus, a TL which does not accept master-slave switch requests can not be guaranteed service by all GWs.

7.2 LINK POLICY

The GW shall be as conservative as possible when deciding what power mode to put the TLs in. This means that when a TL is not engaged in signalling, the GW shall put it in a low-power mode. The recommended low-power mode to use is the park mode.

The low-power mode parameters shall be chosen such that the TL can always return to the active state within 300 ms.

When the gateway does not support the Park mode (as indicated in the LMP_features message), it's up to the terminal how to maintain the link when not engaged in calls: either keep it in active mode, or release the link. After a link release, link re-establishment shall be initiated upon request, by either side. In this case, both gateway and terminal shall be in page scan mode when no active link is established.

If the GW can save power during a call, it may use the sniff mode. A TL may request to be put in the sniff mode.

7.3 ENCRYPTION KEY SIZE

In order to support encrypted broadcast messages, all devices in the profile must support an encryption key size of 5 octets.

8 LC FEATURES

The following table lists all features on the LC level.

	Procedure	Support in TL	Support in GW
1.	Inquiry		X
2.	Inquiry scan	X	
3.	Paging		C2
4.	Page scan		
A	Type R0		
B	Type R1		
C	Type R2		
5.	Packet types		
A	ID packet		
B	NULL packet		
C	POLL packet		
D	FHS packet		
E	DM1 packet		
F	DH1 packet		
G	DM3 packet		
H	DH3 packet		
I	DM5 packet		
J	DH5 packet		
K	AUX packet	X	X
L	HV1 packet		
M	HV2 packet		
N	HV3 packet	M	M
O	DV packet	X	X
6.	Inter-piconet capabilities	O	C1
7.	Voice codec		
A	A-law		
B	µ-law		
C	CVSD	M	M
C1: IF feature 7 THEN M else O			

Table 8.1: LC features

8.1 INQUIRY SCAN

A device which is active in the GW role of the CordlessTelephony profile shall, in the Class of Device field:

1. Set the ‘Telephony’ bit in the Service Class field
2. Indicate ‘Phone’ as Major Device class

This may be used by an inquiring device to filter the inquiry responses.

8.2 INTER-PICONET CAPABILITIES

Inter-piconet capability is the capability, as master, to keep the synchronization of a piconet while page scanning in free slots and allowing for new members to join the piconet. While a new unit is joining the piconet (until the master-slave switch has been performed), operation may temporarily be degraded for the other members.

A GW which supports feature 7, ‘Multiple terminal support’, shall have inter-piconet capabilities. The TL may have inter-piconet capabilities.

9 GENERIC ACCESS PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the Generic Access Profile.

This section defines the support requirements with regards to procedures and capabilities defined in Generic Access Profile.

9.1 MODES

The table shows the support status for Modes within this profile.

	Procedure	Support in TL	Support in GW
1	Discoverability modes		
	Non-discoverable mode	N/A	M
	Limited discoverable mode	N/A	O
	General discoverable mode	N/A	M
2	Connectability modes		
	Non-connectable mode	N/A	X
	Connectable mode	N/A	M
3	Pairing modes		
	Non-pairable mode	M	M
	Pairable mode	O	M

Table 9.1: Modes

9.2 SECURITY ASPECTS

The table shows the support status for Security aspects within this profile.

	Procedure	Support in TL	Support in GW
1	Authentication	M	M
2	Security modes		
	Security mode 1	X	X
	Security mode 2	C1	C1
	Security mode 3	C1	C1
C1: Support for at least one of the security modes 2 and 3 is mandatory.			

Table 9.2: Security aspects

9.3 IDLE MODE PROCEDURES

The table shows the support status for Idle mode procedures within this profile.

	Procedure	Support in TL	Support in GW
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	O	N/A
4	Device discovery	O	N/A
5	Bonding	M	M

Table 9.3: Idle mode procedures

9.3.1 Bonding

It is mandatory for the TL to support initiation of bonding, and for the GW to accept bonding.

10 ANNEX A (INFORMATIVE): SIGNALLING FLOWS

This annex contains signalling diagrams that are used to clarify the interworking between units. This annex is informative only. The diagrams do not represent all possible signalling flows as defined by this profile.

10.1 OUTGOING EXTERNAL CALL WITHOUT POST-DIALLING

The following sequence shows the successful case when the TL does not use overlap sending:

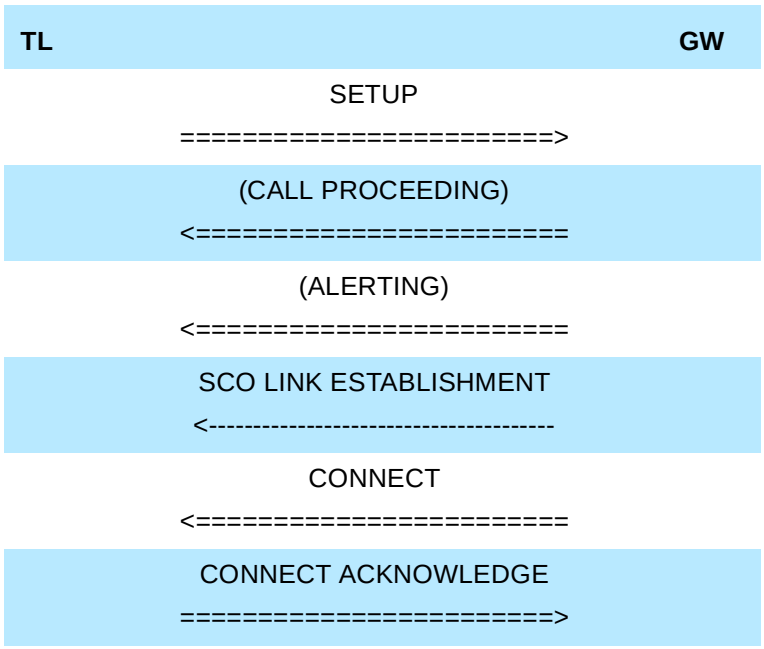


Figure 10.1: TL-originated call when overlap sending is not used

10.2 OUTGOING EXTERNAL CALL WITH POST-DIALLING

The following sequence shows the successful case when post-dialling is used.

TL		GW
	SETUP =======================>	
	SETUP ACKNOWLEDGE <=======================	
This message is repeated until the GW has enough dialling information	INFORMATION =======================>	
	(CALL PROCEEDING) <=======================	When the GW has sufficient information to complete the call, CALL PROCEEDING, ALERTING or CONNECT is sent.
	(ALERTING) <=======================	
	SCO LINK ESTABLISHMENT <---------------------------------------	
	CONNECT <=======================	
	CONNECT ACKNOWLEDGE =======================>	

Figure 10.2: Outgoing external call with post-dialling

10.3 INCOMING EXTERNAL CALL, SETUP DELIVERED ON CONNECTIONLESS CHANNEL

The figure below shows the allowed signalling flow in the successful case:

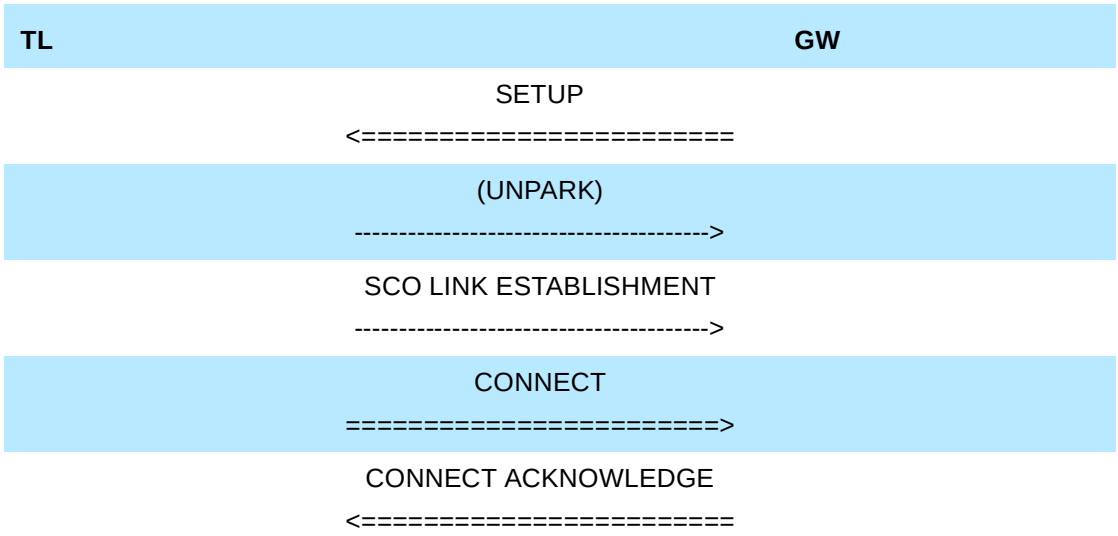


Figure 10.3: Incoming external call, SETUP delivered on connectionless channel

10.4 INCOMING EXTERNAL CALL, SETUP DELIVERED ON CONNECTION-ORIENTED CHANNEL

The figure below shows the allowed signalling flow in the successful case:

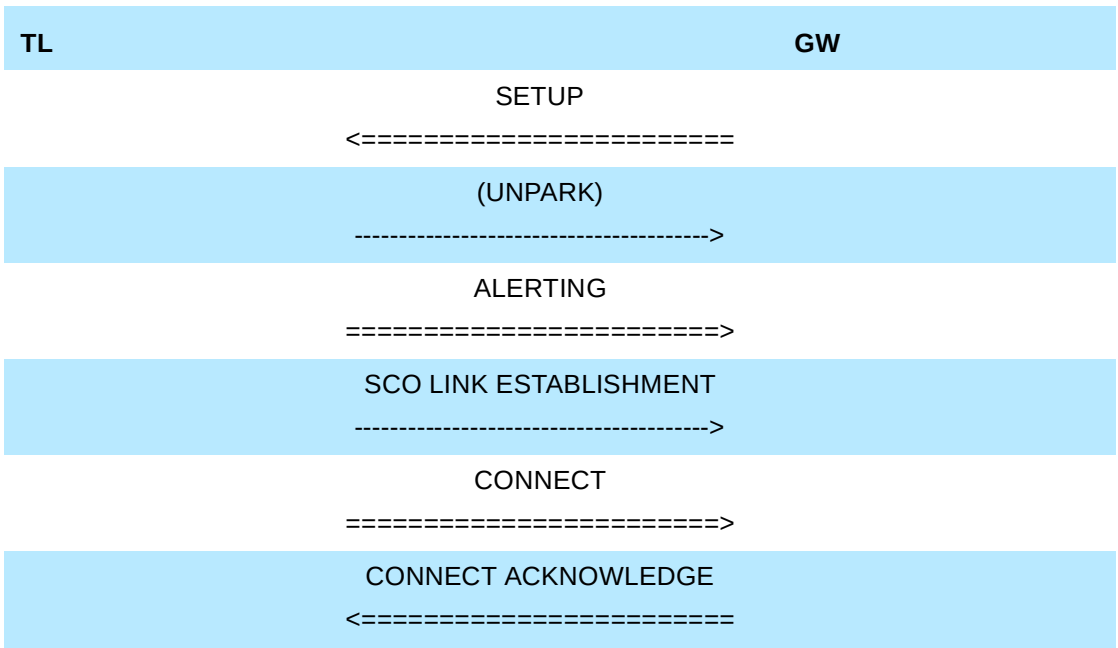


Figure 10.4: Incoming external call, SETUP delivered on connection-oriented channel

10.5 CALL CLEARING

The figure below shows the allowed signalling flow in the successful case when the TL initiates call clearing:

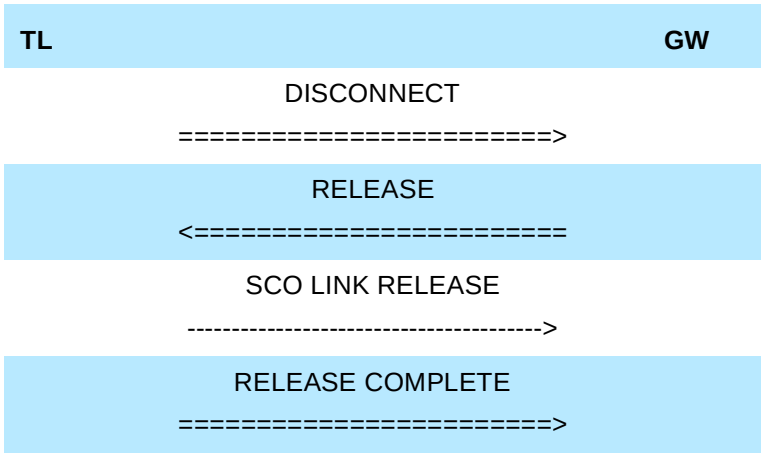


Figure 10.5: Call Clearing signalling flow, successful case

10.6 DTMF SIGNALLING

The figure below shows the allowed signalling flow in the successful case:

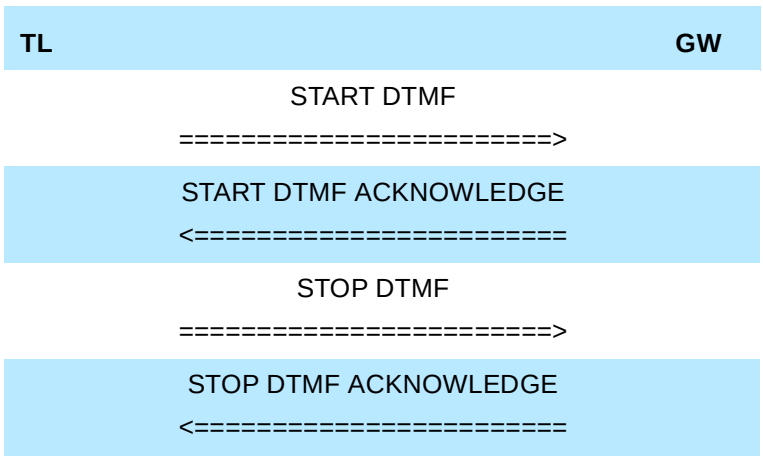


Figure 10.6: DTMF signalling, successful case

10.7 DTMF SIGNALLING FAILURE

The figure below shows the allowed signalling flow in the unsuccessful case:

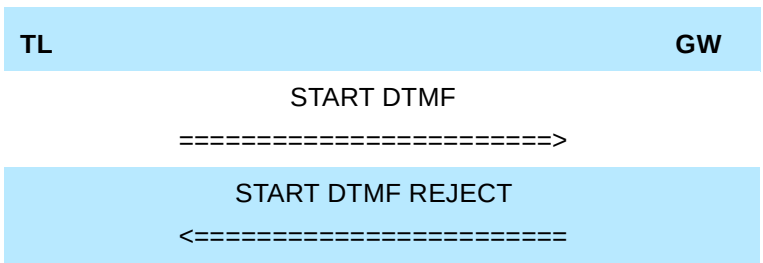


Figure 10.7: DTMF signalling, unsuccessful case

10.8 ACCESS RIGHTS REQUEST

The figure below shows the allowed signalling flow in the successful case:

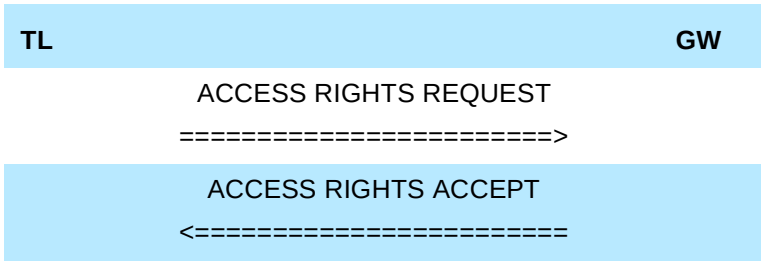


Figure 10.8: Signalling diagram for Access Rights Request

10.9 CONFIGURATION DISTRIBUTION

The figure below shows the allowed signalling flow in the successful case:

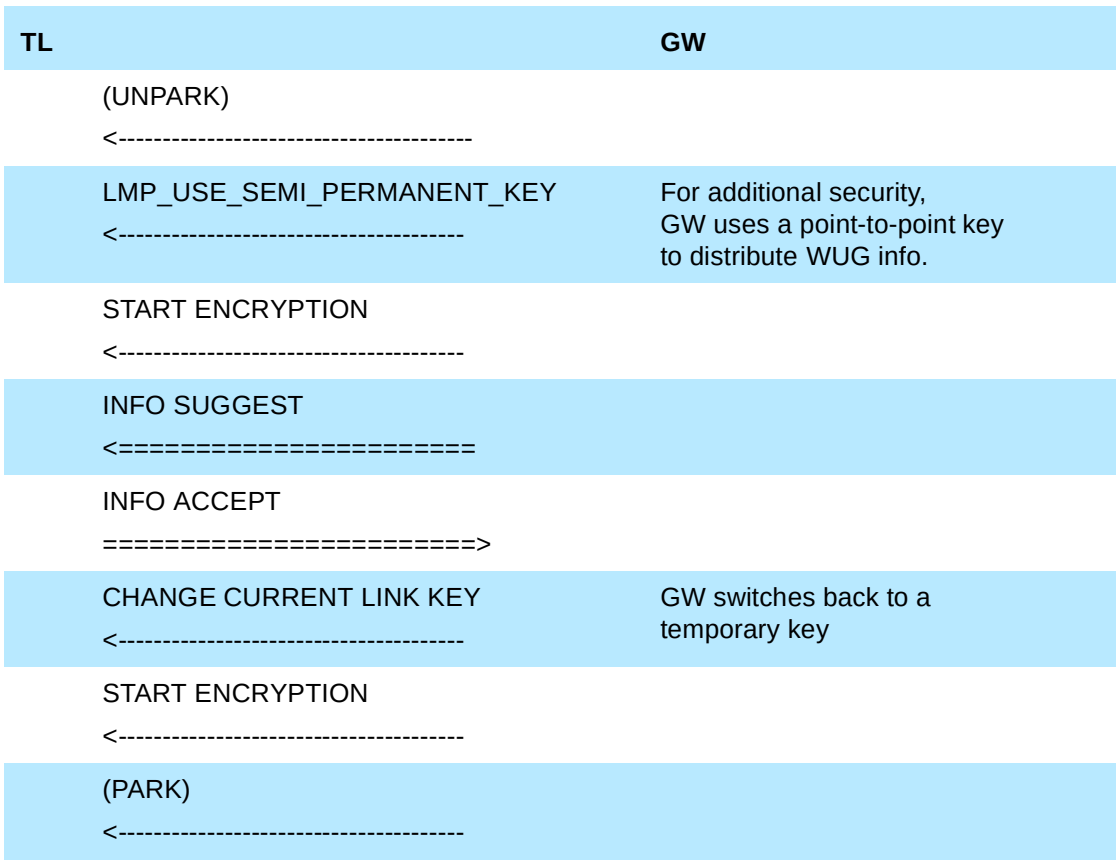


Figure 10.9: Signalling diagram for Configuration distribution

10.10 PERIODIC KEY UPDATE

The figure below shows the allowed signalling flow in the successful case:

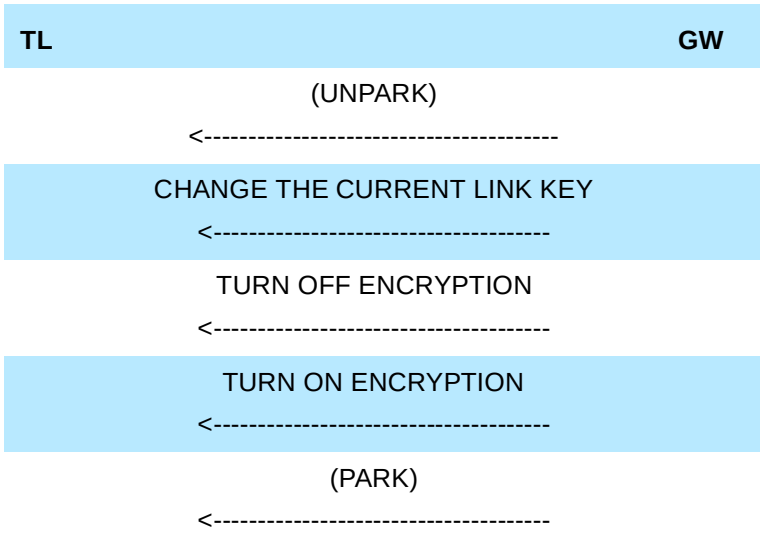


Figure 10.10: Signalling diagram for periodic key update

10.11 FAST INTER-MEMBER ACCESS

The figure below shows the allowed signalling flow in the successful case:

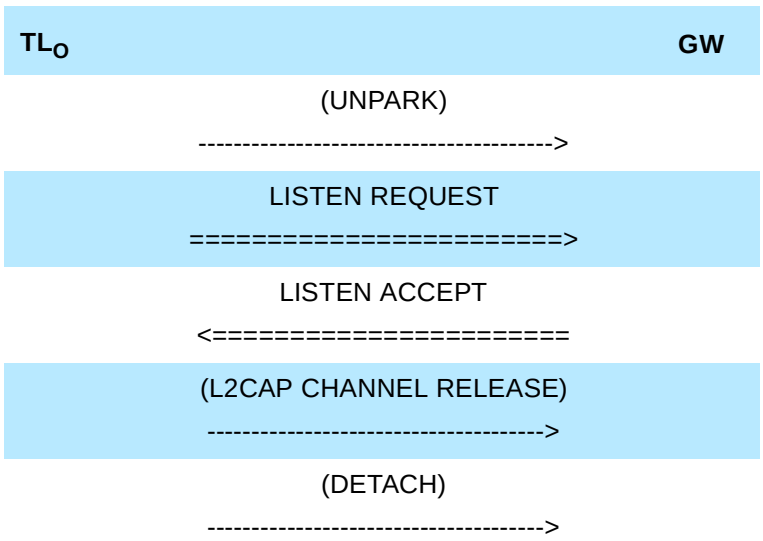


Figure 10.11: Signalling diagram for Fast inter-member access, originating side

The figure below shows the valid sub-procedure sequence between the TL_T and GW:

TL_T	**GW**
	(UNPARK) <----------
	LISTEN SUGGEST <==========
(as soon as possible after sending the LISTEN ACCEPT, the TL_T starts page scanning)	LISTEN ACCEPT ==========>
	(L2CAP CHANNEL RELEASE) ---------->
	(DETACH) ---------->

Figure 10.12: Signalling diagram for Fast inter-member access, terminating side

11 TIMERS AND COUNTERS

Timer name	Proposed value	Description	Comment
$T_{CTP}400$	1 week	Time between periodic key updates, depending on the required security level	

Table 11.1: Defined timers

12 REFERENCES

[1] Bluetooth Baseband Specification

[2] Bluetooth Link Manager Protocol

[3] Bluetooth Logical Link Control and Adaptation Protocol Specification

[4] Bluetooth Telephony Control Protocol Specification

[5] Bluetooth Service Discovery Protocol

[6] Bluetooth Intercom Profile

[7] Bluetooth Assigned Numbers
http://www.bluetooth.org/assigned-numbers.htm

[8] Thomas Müller, Security Architecture Whitepaper, version 0.5

13 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 101
Figure 2.1: Protocol model 103
Figure 2.2: System configuration example 105
Figure 10.1: TL-originated call when overlap sending is not used 128
Figure 10.2: Outgoing external call with post-dialling 129
Figure 10.3: Incoming external call, SETUP delivered on connectionless channel 130
Figure 10.4: Incoming external call, SETUP delivered on connection-oriented channel 130
Figure 10.5: Call Clearing signalling flow, successful case 131
Figure 10.6: DTMF signalling, successful case 131
Figure 10.7: DTMF signalling, unsuccessful case 131
Figure 10.8: Signalling diagram for Access Rights Request 132
Figure 10.9: Signalling diagram for Configuration distribution 132
Figure 10.10: Signalling diagram for periodic key update 133
Figure 10.11: Signalling diagram for Fast inter-member access, originating side 133
Figure 10.12: Signalling diagram for Fast inter-member access, terminating side 134

14 LIST OF TABLES

Table 3.1: Application layer features 108
Table 3.2: Application layer feature to procedure mapping 108
Table 4.1: TCS-BIN messages 116
Table 4.2: TCS-BIN information elements 117
Table 4.3: Restrictions to contents of Bearer capability information element 118
Table 4.4: Restrictions to contents of Cause information element 119
Table 5.1: SDP entry for GW service 120
Table 7.1: LMP procedures 122
Table 8.1: LC features 124
Table 9.1: Modes 126
Table 9.2: Security aspects 126
Table 9.3: Idle mode procedures 127
Table 11.1: Defined timers 135

Part K:4

INTERCOM PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the intercom functionality within the 3-in-1 phone use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the 3-in-1 phone use case.

CONTENTS

1 Introduction 143
1.1 Scope 143
1.2 Profile Dependencies 143
1.3 Symbols and conventions 144
1.3.1 Requirement status symbols 144
1.3.2 Signalling diagram conventions 144
2 Profile Overview 145
2.1 Profile stack 145
2.2 Configuration and roles 146
2.3 User requirements and scenarios 146
2.4 Profile fundamentals 147
2.5 Feature definitions 147
2.6 Conformance 147
3 Application layer 148
4 TCS Binary 149
4.1 Call Control procedures 149
4.1.1 Call request 149
4.1.2 Call confirmation 149
4.1.3 Call connection 149
4.1.4 Failure of call establishment 149
4.1.5 Call clearing 150
4.1.6 Call information 150
4.2 TCS Binary Message overview 150
4.3 Information Element overview 151
4.3.1 Bearer capability 152
4.3.2 Call class 152
4.3.3 Cause 152
4.4 Link loss 152
5 SDP Interoperability Requirements 153
6 L2CAP Interoperability Requirements 154
6.1 Channel types 154
6.2 Configuration options 154
6.2.1 Maximum Transmission unit 154
6.2.2 Flush timeout option 154
6.2.3 Quality of Service 154

7 Link Manager (LM) Interoperability Requirements 155
7.1 Capability overview 155
8 Link Control (LC) Interoperability Requirements 156
8.1 Capability overview 156
8.2 Class of Device 157
9 Generic Access Profile 158
9.1 Modes 158
9.2 Security aspects 158
9.3 Idle mode procedures 158
10 Annex A (Informative): Signalling flows 159
10.1 Call establishment 159
10.2 Call Clearing 160
11 Timers and counters 161
12 List of Figures 162
13 List of Tables 163

1 INTRODUCTION

1.1 SCOPE

This Intercom profile defines the protocols and procedures that shall be used by devices implementing the intercom part of the usage model called‘3-in-1 phone’. More popularly, this is often referred to as the ‘walkie-talkie’ usage of Bluetooth.

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly. As indicated in the figure, the Intercom profile is dependent only upon the Generic Access Profile – details are provided in Section 9.

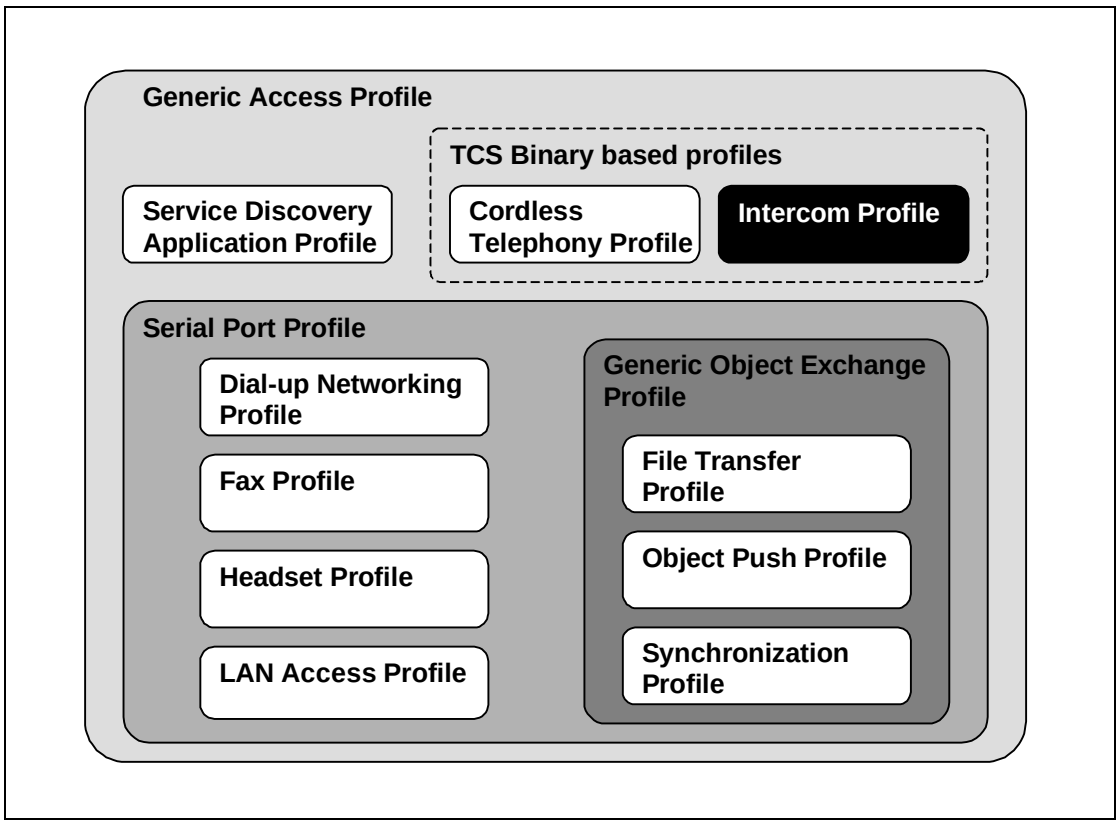


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

- 'M' for mandatory to support
- 'O' for optional to support
- 'X' for excluded (used for capabilities that may be supported by the unit but shall never be used in the use case)
- 'C' for conditional to support
- 'N/A' for not applicable (in the given context it is impossible to use this capability)

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices in this use case. Therefore, these features shall never be activated while a unit is operating as a unit within this use case.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:

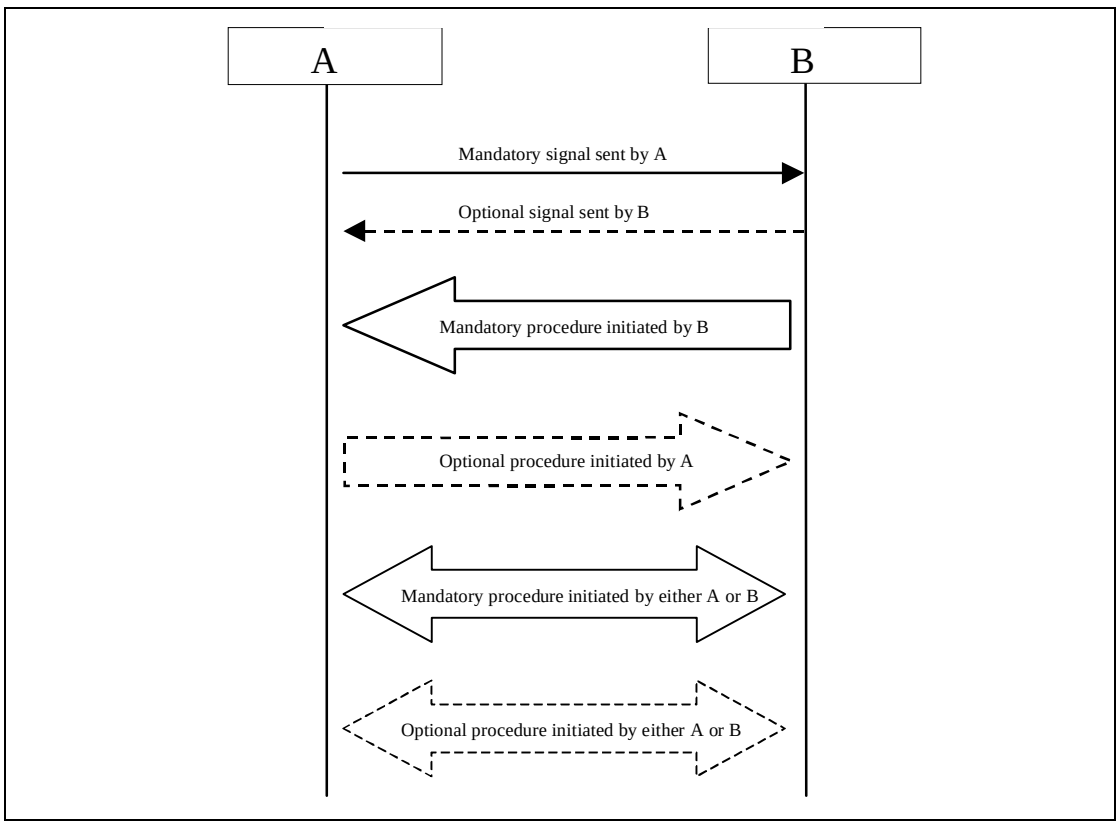


Figure 1.2: Arrows used in signalling diagrams

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 below shows the protocols as used within this profile:

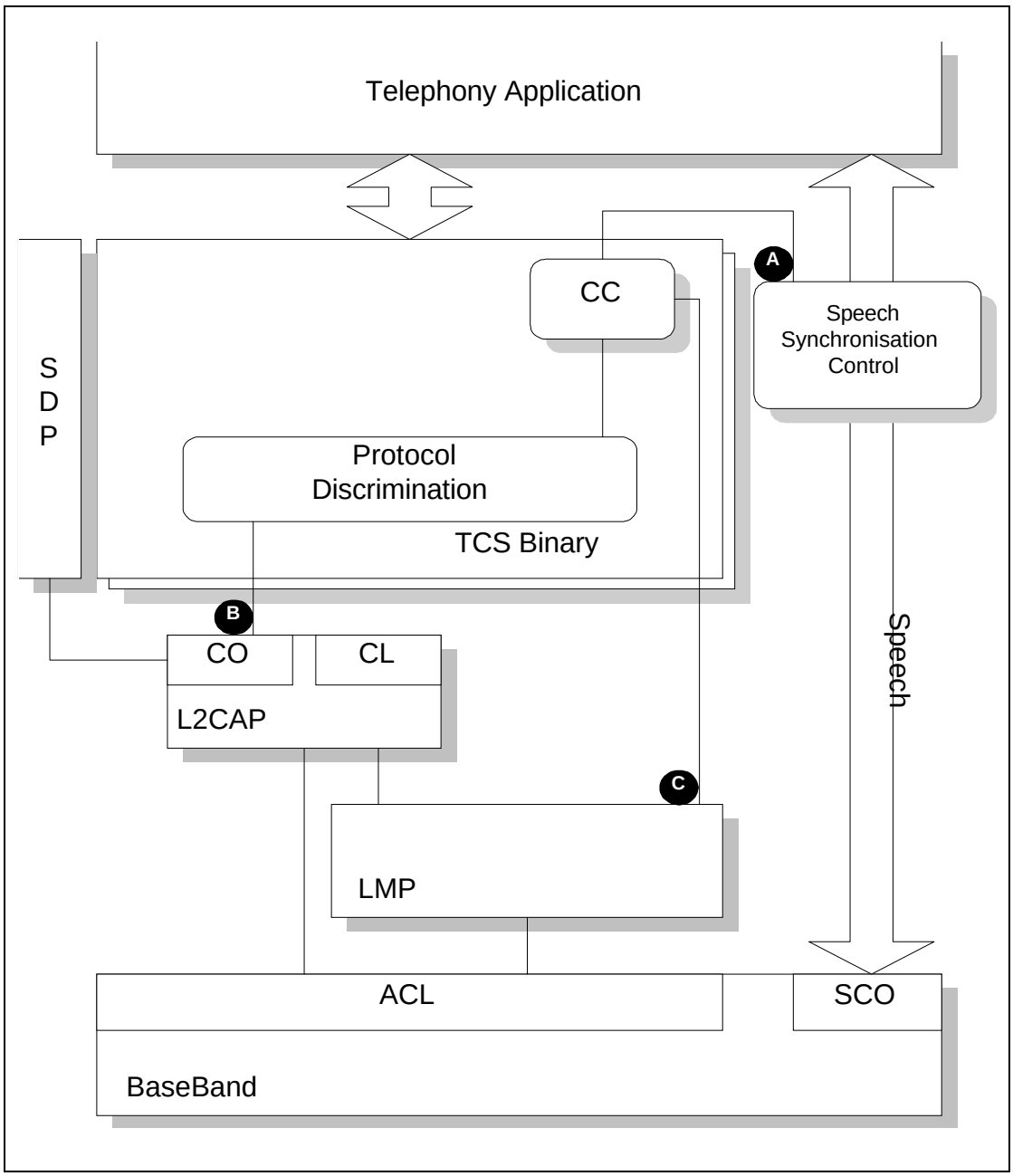


Figure 2.1: Intercom Profile Stack

This profile will define the requirements for each of the layers in the model above.

In the profile, the interfaces in Figure 2.1 above are used for the following purposes:

A) The Call Control entity uses this interface to the speech synchronization control to connect and disconnect the internal speech paths;

B) Used to deliver TCS messages on the connection-oriented (point-to-point) L2CAP channel;

C) Used by the Call Control entity to control the Link Manager directly for the purpose of establishing and releasing SCO links;

Note that, for initialization purposes, it is additionally required to control the LC/Baseband directly, to enable inquiry, paging, inquiry scan, page scan.

2.2 CONFIGURATION AND ROLES

The figure below shows a typical configuration of devices for which the Intercom profile is applicable:

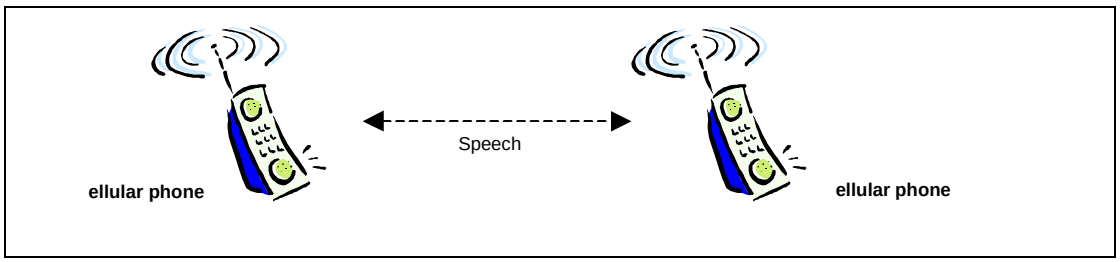


Figure 2.2: Intercom profile, example

As the intercom usage is completely symmetrical, there are no specific roles defined. A device supporting the Intercom profile will generally be denoted as Terminal (TL).

2.3 USER REQUIREMENTS AND SCENARIOS

The Intercom profile defines the protocols and procedures that shall be used by devices implementing the intercom part of the use case called‘3-in-1 phone’.

The scenarios targeted by this use case are typically those where a direct speech link is required between two devices (phone, computer, …), established using telephony-based signalling.

A typical scenario is the following:

- Two (cellular) phone users engaged in a speech call, on a direct phone-to-phone connection using Bluetooth only.

2.4 PROFILE FUNDAMENTALS

Here is a brief summary of the interactions that take place when a terminal wants to establish an intercom call towards another terminal. In the description below, the term initiator (A-party) and acceptor (B-party) will be used to designate the direction of the call.

1. If the initiator of the intercom call does not have the Bluetooth Address of the acceptor, it has to obtain this; e.g. using the Device discovery procedure – see Section 6.4 of Generic Access profile.
2. The profile does not mandate a particular security mode. If users of either device (initiator/acceptor) want to enforce security in the execution of this profile, the authentication procedure (see Section 5.1 of Generic Access profile) has to be performed to create a secure connection.
3. The initiator establishes the link and channel as indicated in Section 7 of the Generic Access profile. Based on the security requirements enforced by users of either device, authentication may be performed and encryption may be enabled.
4. The intercom call is established.
5. After the intercom call has been cleared, the channel and link will be released as well.

2.5 FEATURE DEFINITIONS

Call information **–** The ability to provide additional information during the active phase of a call.

Intercom call – A speech call between two terminals.

On hook **–** The ability to indicate the action of going on-hook (e.g. to terminate a call) and release of all radio resources related to that call.

2.6 CONFORMANCE

When conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities, for which support is indicated are subject to verification as part of the Bluetooth certification program.

3 APPLICATION LAYER

The following text together with the associated sub-clauses defines the feature requirements with regard to this profile.

Table 3.1 below shows the feature requirements made by this profile.

Item no.	Feature	Support
1.	Intercom call	M
2.	On hook	M
3.	Call information	O

Table 3.1: Application layer features

Table 3.2 below maps each feature to the TCS Binary procedures used for that feature and shows whether the procedure is optional, mandatory or conditional for that feature.

Item no.	Feature	Procedure	Ref.	Support
1.	Intercom call	Call request	4.1.1	M
		Call confirmation	4.1.2	M
		Call connection	4.1.3	M
2.	On hook	Call clearing	4.1.5	M
3.	Call information	Call information	4.1.6	M

Table 3.2: Application layer feature to procedure mapping

4 TCS BINARY

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

When describing TCS Binary procedures, this chapter provides additional information concerning lower layer handling. The normative reference for TCS Binary procedures is TCS Binary.

Annex A contains signalling flows that illustrate the procedures in this chapter.

4.1 CALL CONTROL PROCEDURES

4.1.1 Call request

This procedure shall be performed as defined in Section 2.2.1 of TCS Binary. In addition, the following applies: before a call request can be made, a connection-oriented L2CAP channel needs to be established between the two devices, using the procedures as indicated in Section 6. When the L2CAP channel has been established, the terminating side will start timer $T_{ic}(100)$. When, at expiry of timer $T_{ic}(100)$, the terminating side has not received the SETUP message initiating the call request, it may terminate the L2CAP channel. Receiving the SETUP message before expiry of $T_{IC}(100)$ will cancel the timer.

4.1.2 Call confirmation

This procedure shall be performed as defined in Section 2.2.5 of TCS Binary.

4.1.3 Call connection

This procedure shall be performed as defined in Section 2.2.6 of TCS Binary. The following text defines the mandatory requirements with regard to this profile.

The SCO link establishment sub-procedure (see LMP, Section 3.21) shall be initiated before sending a CONNECT.

The speech path shall be connected by a unit when it receives a CONNECT or CONNECT ACKNOWLEDGE.

4.1.4 Failure of call establishment

This procedure shall be performed as defined in Section 2.2.10 of TCS Binary. Additionally, the text in Section 4.1.5 defines the mandatory requirements with regard to this profile concerning call clearing.

4.1.5 Call clearing

All call-clearing and call-collision procedures as defined in Section 2.3 of TCS Binary shall be supported by the TL.

In addition, the following applies: after the last call-clearing message has been sent, a unit shall:

- release the SCO link by invoking the appropriate LMP sub-procedure (see LMP, Section 3.21.5), if not already released.
- terminate the L2CAP channel used for TCS Call-control signalling (if not already terminated) and detach the other unit.

4.1.6 Call information

This procedure shall be performed as defined in Section 2.2.7 of TCS Binary.

4.2 TCS BINARY MESSAGE OVERVIEW

This section defines the requiredTCS Binary messages in the Intercom profile.

Message	Support
Alerting	M
Connect	M
Connect Acknowledge	M
Disconnect	M
Information	O
Release	M
Release Complete	M
Setup	M

Table 4.1: TCS Binary messages

4.3 INFORMATION ELEMENT OVERVIEW

This section together with the associated sub-clauses defines the requiredin-formation elements used in TCS Binary messages in the Intercom profile.

Information element	Support
Message type	M
Audio control	O
Bearer capability	M
Call class	M
Called party number	O
Calling party number	O
Cause	M
Clock offset	N/A
Company-specific	O
Configuration data	N/A
Destination CID	N/A
Keypad facility	O
Progress indicator	N/A
SCO handle	M
Sending complete	O
Signal	O

Table 4.2: TCS Binary information elements

The following subsections define restrictions that apply to the contents of the TCS Binary information elements in the Intercom profile. Note that, in the tables, only fields where restrictions apply are shown. If a field is not shown in a table, it means that all values defined in Section 7 of TCS Binary for that field are allowed.

For those information elements not listed below, no restrictions apply.

4.3.1 Bearer capability

The following restrictions apply to the contents of the Bearer capability information element:

Field	Values allowed
Link type	SCO, None
User information layer 1	CVSD

Table 4.3: Restrictions to contents of Bearer capability information element

4.3.2 Call class

The following restrictions apply to the contents of the Call class information element:

Field	Values allowed
Call class	Intercom call

Table 4.4: Restrictions to contents of Call class information element

4.3.3 Cause

The following restrictions apply to the contents of the Cause information element:

Field	Values allowed
Cause value	#16 – “Normal call clearing” #17 – “User busy”, #18 – “No user responding”, #19 – “No answer from user (user alerted)”, #21 – “Call rejected by user” #34 – “No circuit/channel available”, #41 – “Temporary failure”, #44 – “Requested circuit/channel not available”, #58 – “Bearer capability not presently available”, #65 – “Bearer capability not implemented”, #69 – “Requested facility not implemented”, #102 – “Recovery on timer expiry”

Table 4.5: Restrictions to contents of Cause information element

4.4 LINK LOSS

If a unit in a CC state other than *Null* detects loss of link, it shall immediately go to the *Null* state. Call clearing procedures shall in this case not be performed.

5 SDP INTEROPERABILITY REQUIREMENTS

Table 5.1 lists all intercom-related entries in the SDP database. For each field, the Status column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonic's used in the Value column as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column) can be found in the Bluetooth Assigned Numbers document (http://www.bluetooth.org/assigned-numbers.htm).

Item	Definition	Type	Value	AttrID	Status	Default
ServiceClassIDList					M	
ServiceClass0		UUID	Intercom		M	
ServiceClass1		UUID	Generic Telephony		M	
Protocol Descriptor List					M	
Protocol0		UUID	L2CAP		M	
Protocol1		UUID	TCS-BIN		M	
BluetoothProfileDe-scriptorList					O	
Profile0	Sup-ported Profiles	UUID	Intercom		M	Intercom
Param0	Profile Version	Uint16	0x0100*		M	0x0100
Service Name	Display-able Text name	String	Service-provider defined		O	"Intercom"

Table 5.1: Service Record

*. Indicating version 1.0

6 L2CAP INTEROPERABILITY REQUIREMENTS

The following text together with the associated sub-clauses define the mandatory requirements with regard to this profile.

6.1 CHANNEL TYPES

In this profile, only connection-oriented channels are used. In the PSM field of the Connection Request packet, the default value for TCS-BIN, 0x0005 (see Section 3.2 of Assigned Numbers) shall be used.

6.2 CONFIGURATION OPTIONS

This section describes the usage of configuration options.

6.2.1 Maximum Transmission unit

The minimum MTU that a L2CAP implementation used for this profile should support is 3 octets.

6.2.2 Flush timeout option

The flush timeout value used for both the GW and the TL shall be the default value of 0xFFFF.

6.2.3 Quality of Service

Negotiation of Quality of Service is optional.

7 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

7.1 CAPABILITY OVERVIEW

In the table below, all LM capabilities are listed. In the table it is shown what LMP features are mandatory to support with respect to this profile and which are optional.

	Procedure	Support in LMP	Support
1.	Authentication	M	
2.	Pairing	M	
3.	Change link key	M	
4.	Change the current link key	M	
5.	Encryption	O	
6.	Clock offset request	M	
7.	Slot offset information	O	
8.	Timing accuracy information request	O	
9.	LMP version	M	
10.	Supported features	M	
11.	Switch of master slave role	O	
12.	Name request	M	
13.	Detach	M	
14.	Hold mode	O	
15.	Sniff mode	O	
16.	Park mode	O	
17.	Power control	O	
18.	Channel quality driven DM/DH	O	
19.	Quality of service	M	
20.	SCO links	O	M
21.	Control of multi-slot packets	O	
22.	Paging scheme	O	
23.	Link supervision	M	
24.	Connection establishment	M	

Table 7.1: LMP procedures

8 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

8.1 CAPABILITY OVERVIEW

The following table lists all capabilities on the LC level.

	Capabilities	Support
1.	Inquiry	
2.	Inquiry scan	
3.	Paging	
4.	Page scan	
A	Type R0	
B	Type R1	
C	Type R2	
5.	Packet types	
A	ID packet	
B	NULL packet	
C	POLL packet	
D	FHS packet	
E	DM1 packet	
F	DH1 packet	
G	DM3 packet	
H	DH3 packet	
I	DM5 packet	
J	DH5 packet	
K	AUX packet	X
L	HV1 packet	
M	HV2 packet	
N	HV3 packet	
O	DV packet	
6.	Inter-piconet capabilities	

Table 8.1: Baseband/LC capabilities

	Capabilities	Support
7.	Voice codec	
A	A-law	
B	µ-law	
C	CVSD	M

Table 8.1: Baseband/LC capabilities

8.2 CLASS OF DEVICE

The Class of Device field shall be set to the following:

1. Set the ‘Generic Telephony’ bit in the Service Class field
2. Indicate ‘Phone’ as Major Device class

9 GENERIC ACCESS PROFILE

This section defines the support requirements for the capabilities as defined in Generic Access Profile.

9.1 MODES

The table shows the support status for Modes within this profile.

	Procedure	Support
1	Discoverability modes	
	Non-discoverable mode	M
	Limited discoverable mode	O
	General discoverable mode	M
2	Connectability modes	
	Non-connectable mode	N/A
	Connectable mode	M
3	Pairing modes	
	Non-pairable mode	O
	Pairable mode	C3
C3: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise optional		

Table 9.1: Modes

9.2 SECURITY ASPECTS

No changes to the requirements as stated in the Generic Access Profile.

9.3 IDLE MODE PROCEDURES

The table shows the support status for Idle mode procedures within this profile.

	Procedure	Support
1	General inquiry	M
2	Limited inquiry	O
3	Name discovery	O
4	Device discovery	O
5	Bonding	O

Table 9.2: Idle mode procedures

10 ANNEX A (INFORMATIVE): SIGNALLING FLOWS

This annex contains signalling diagrams that are used to clarify the interworking between units. This annex is informative only. The diagrams do not represent all possible signalling flows as defined by this profile.

10.1 CALL ESTABLISHMENT

The figure below shows the allowed signalling flow in the successful case:

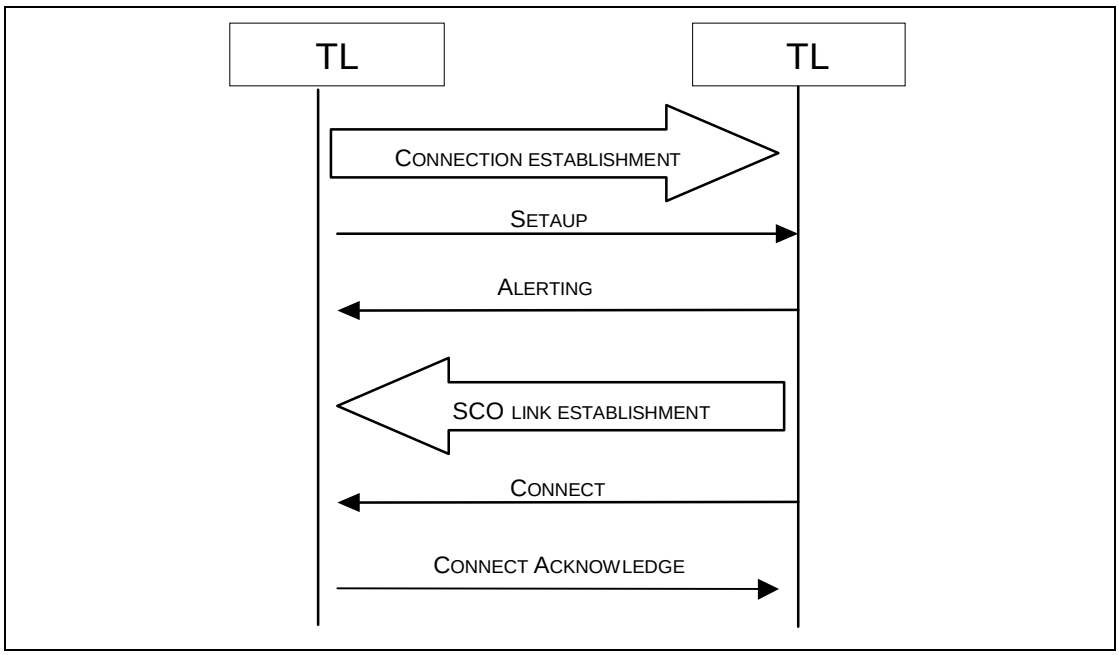


Figure 10.1: Call establishment

10.2 CALL CLEARING

The figure below shows the allowed signalling flow for the call clearing:

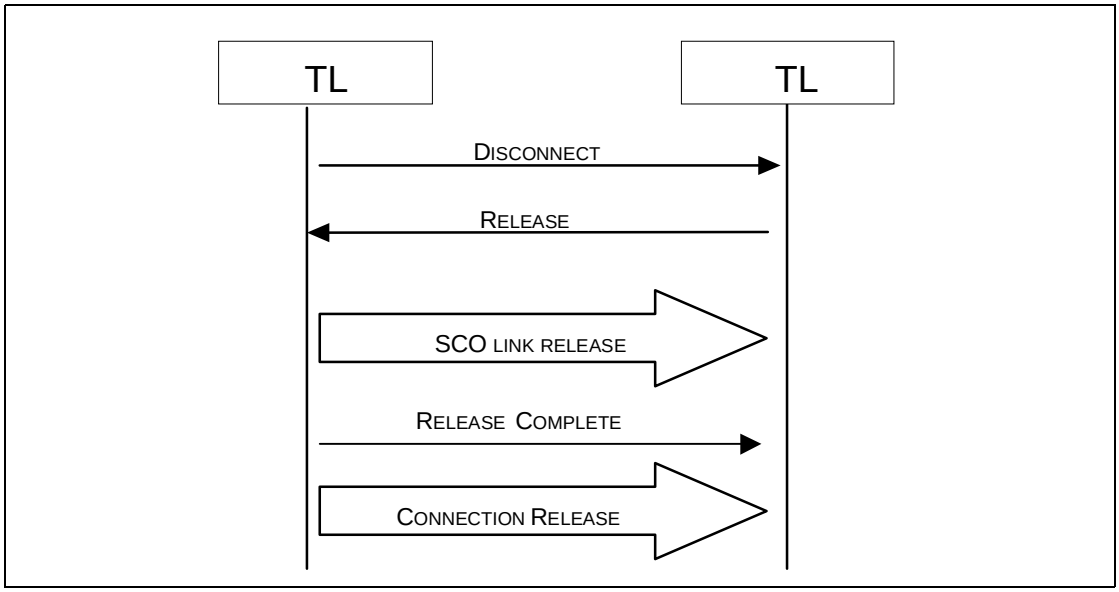


Figure 10.2: Call Clearing signalling flow, successful case

11 TIMERS AND COUNTERS

Timer name	Proposed value	Description	Comment
$T_{IC}(100)$	10s	Time between L2CAP connection establishment and call request initiation	

12 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles....143
Figure 1.2: Arrows used in signalling diagrams....144
Figure 2.1: Intercom Profile Stack....145
Figure 2.2: Intercom profile, example....146
Figure 10.1: Call establishment....159
Figure 10.2: Call Clearing signalling flow, successful case....160

13 LIST OF TABLES

Table 3.1: Application layer features....148
Table 3.2: Application layer feature to procedure mapping....148
Table 4.1: TCS Binary messages....150
Table 4.2: TCS Binary information elements....151
Table 4.3: Restrictions to contents of Bearer capability information element....152
Table 4.4: Restrictions to contents of Call class information element....152
Table 4.5: Restrictions to contents of Cause information element....152
Table 5.1: Service Record....153
Table 7.1: LMP procedures....155
Table 8.1: Baseband/LC capabilities....156
Table 9.1: Modes....158
Table 9.2: Idle mode procedures....158

Part K:5

SERIAL PORT PROFILE

This profile defines the requirements for Bluetooth devices necessary for setting up emulated serial cable connections using RFCOMM between two peer devices. The requirements are expressed in terms of services provided to applications, and by defining the features and procedures that are required for interoperability between Bluetooth devices.

CONTENTS

1 Introduction 170
1.1 Scope 170
1.2 Bluetooth Profile Structure 170
1.3 Symbols and conventions 170

2 Profile overview 171
2.1 Profile stack 171
2.2 Configurations and roles 172
2.3 User requirements and scenarios 172
2.4 Profile fundamentals 173
2.5 Conformance 173

3 Application layer 174
3.1 Procedure overview 174
3.1.1 Establish link and set up virtual serial connection 174
3.1.2 Accept link and establish virtual serial connection 175
3.1.3 Register Service record in local SDP database 175
3.2 Power mode and link loss handling 176

4 RFCOMM Interoperability Requirements 177
4.1 RS232 control signals 178
4.2 Remote Line Status indication 178
4.3 Remote Port Negotiation 178

5 L2CAP Interoperability Requirements 179
5.1 Channel types 179
5.2 Signalling 179
5.3 Configuration options 180
5.3.1 Maximum Transmission unit 180
5.3.2 Flush Timeout 180
5.3.3 Quality of Service 180

6 SDP Interoperability Requirements 181
6.1 SDP Service Records for Serial Port Profile 181
6.2 SDP Procedures 182

7 Link Manager (LM) Interoperability Requirements 183
7.1 Capability overview 183
7.2 Error behavior 183
7.3 Link policy 183

8 Link Control (LC) Interoperability Requirements 184
8.1 Capability overview 184
8.2 Inquiry 185
8.3 Inquiry scan 185
8.4 Paging 185
8.5 Error behavior 185
9 References 186
10 List of Figures 187
11 List of Tables 188

FOREWORD

Interoperability between devices from different manufacturers is provided for a specific service and use case, if the devices conform to a Bluetooth SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications, and gives an unambiguous description of the air interface for specified service(s) and use case(s).

All defined features are process-mandatory. This means that if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

The Serial Port Profile defines the protocols and procedures that shall be used by devices using Bluetooth for RS232 (or similar) serial cable emulation.

The scenario covered by this profile deals with legacy applications using Bluetooth as a cable replacement, through a virtual serial port abstraction (which in itself is operating system-dependent).

1.2 BLUETOOTH PROFILE STRUCTURE

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

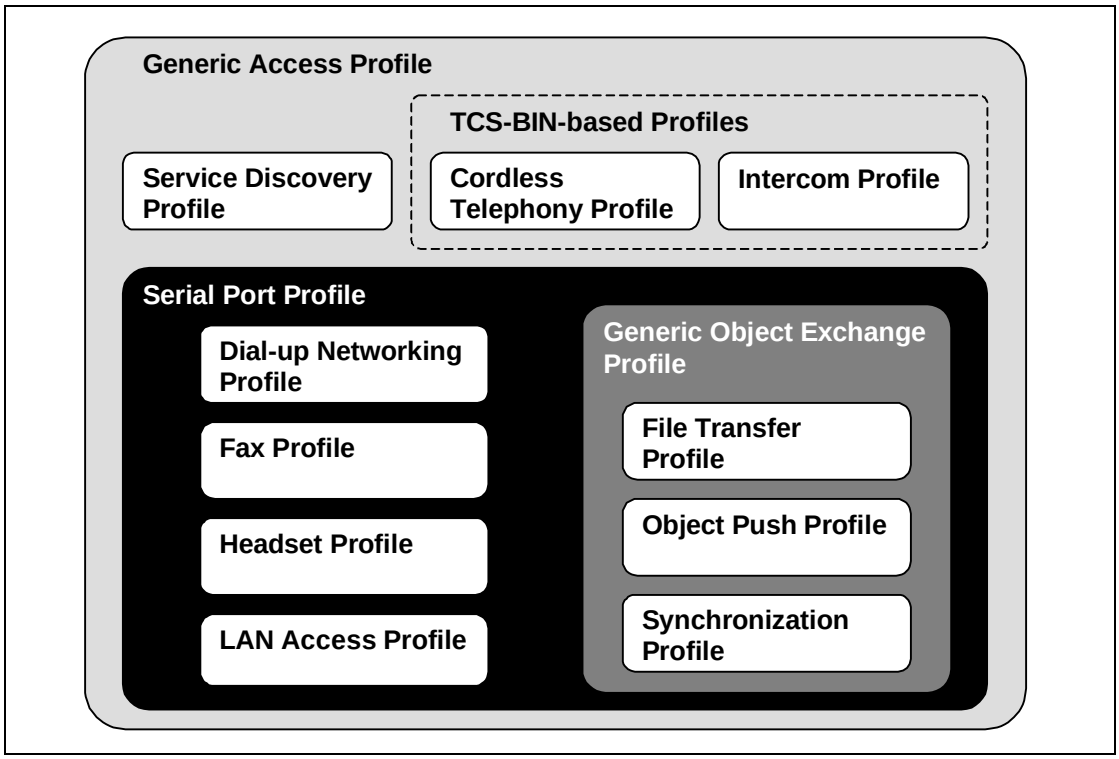


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [9].

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

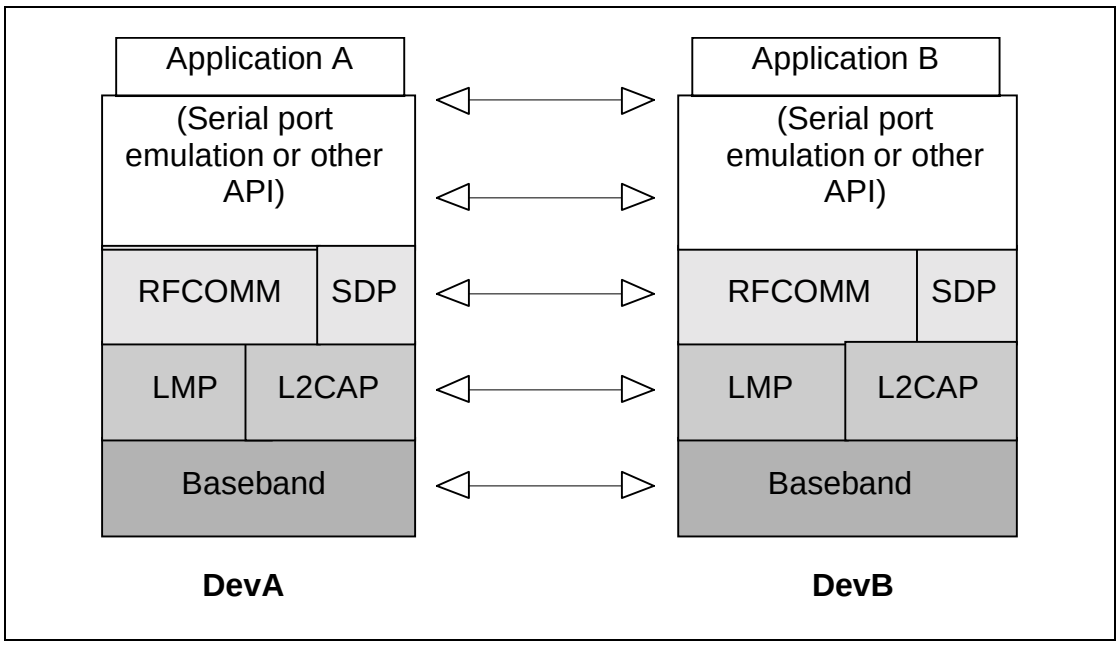


Figure 2.1: Protocol model

The Baseband [1], LMP [2] and L2CAP [3] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [4] is the Bluetooth adaptation of GSM TS 07.10 [5], providing a transport protocol for serial port emulation. SDP is the Bluetooth Service Discovery Protocol [6].

The port emulation layer shown in Figure 2.1 is the entity emulating the serial port, or providing an API to applications.

The applications on both sides are typically legacy applications, able and wanting to communicate over a serial cable (which in this case is emulated). But legacy applications cannot know about Bluetooth procedures for setting up emulated serial cables, which is why they need help from some sort of Bluetooth-aware helper application on both sides. (These issues are not explicitly addressed in this profile; the major concern here is for Bluetooth interoperability.)

2.2 CONFIGURATIONS AND ROLES

The figure below shows one possible configuration of devices for this profile:

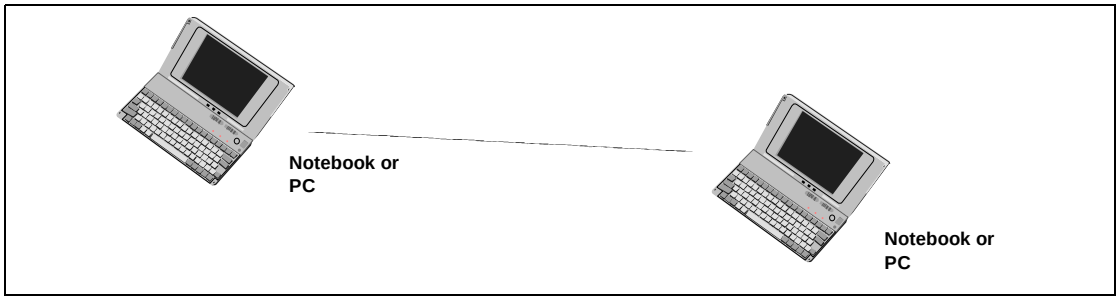


Figure 2.2: Serial Port profile, example with two notebooks.

The following roles are defined for this profile:

Device A (DevA) – This is the device that takes initiative to form a connection to another device (DevA is the *Initiator* according to Section 2.2 in GAP [9]).

Device B (DevB) – This is the device that waits for another device to take initiative to connect. (DevB is the *Acceptor* according to Section 2.2 in GAP [9]).

Note that the order of connection (from DevA to DevB) does not necessarily have to have anything to do with the order in which the legacy applications are started on each side respectively.

Informational note: For purposes of mapping the Serial Port profile to the conventional serial port architecture, both DevA and DevB can be either a Data Circuit Endpoint (DCE) or a Data Terminal Endpoint (DTE). (The RFCOMM protocol is designed to be independent of DTE-DCE or DTE-DTE relationships.)

2.3 USER REQUIREMENTS AND SCENARIOS

The scenario covered by this profile is the following:

- Setting up virtual serial ports (or equivalent) on two devices (e.g. PCs) and connecting these with Bluetooth, to emulate a serial cable between the two devices. Any legacy application may be run on either device, using the virtual serial port as if there were a real serial cable connecting the two devices (with RS232 control signalling).

This profile requires support for one-slot packets only. This means that this profile ensures that data rates up to 128 kbps can be used. Support for higher rates is optional.

Only one connection at a time is dealt with in this profile. Consequently, only point-to-point configurations are considered. However, this should not be construed as imposing any limitation on concurrence; multiple executions of this profile should be able to run concurrently in the same device. This also includes taking on the two different roles (as DevA and DevB) concurrently.

2.4 PROFILE FUNDAMENTALS

For the execution of this profile, use of security features such as authorization, authentication and encryption is optional. Support for authentication and encryption is mandatory, such that the device can take part in the corresponding procedures if requested from a peer device. If use of security features is desired, the two devices are paired during the connection establishment phase (if not earlier), see GAP, Section 7.

Bonding is not explicitly used in this profile, and thus, support for bonding is optional.

Link establishment is initiated by DevA. Service discovery procedures have to be performed to set up an emulated serial cable connection.

There are no fixed master slave roles.

RFCOMM is used to transport the user data, modem control signals and configuration commands.

2.5 CONFORMANCE

When conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities and optional and conditional capabilities, for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 APPLICATION LAYER

This section describes the feature requirements on units complying with the Serial Port profile.

This profile is built upon the Generic Access Profile [9].

- When reading [9], the A-party (the connection initiator) is equivalent to DevA and the B-party is equivalent to the DevB.
- All the mandatory requirements defined in [9] are mandatory for this profile.
- Unless otherwise stated below, all the optional requirements defined in [9] are optional for this profile.

3.1 PROCEDURE OVERVIEW

Table 3.1 shows the required procedures:

	Procedure	Support in DevA	Support in DevB
1.	Establish link and set up virtual serial connection.	M	X
2.	Accept link and establish virtual serial connection.	X	M
3.	Register Service record for application in local SDP database.	X	M

Table 3.1: Application layer procedures

3.1.1 Establish link and set up virtual serial connection

This procedure refers to performing the steps necessary to establish a connection to an emulated serial port (or equivalent) in a remote device. The steps in this procedure are:

1. Submit a query using SDP to find out the RFCOMM Server channel number of the desired application in the remote device.
   This might include a browsing capability to let the user select among available ports (or services) in the peer device. Or, if it is known exactly which service to contact, it is sufficient look up the necessary parameters using the Service Class ID associated with the desired service.
2. Optionally, require authentication of the remote device to be performed. Also optionally, require encryption to be turned on.
3. Request a new L2CAP channel to the remote RFCOMM entity.
4. Initiate an RFCOMM session on the L2CAP channel.

5. Start a new data link connection on the RFCOMM session, using the aforementioned server channel number.

After step 5, the virtual serial cable connection is ready to be used for communication between applications on both sides.

Note: If there already exists an RFCOMM session between the devices when setting up a new data link connection, the new connection must be established on the existing RFCOMM session. (This is equivalent to skipping over steps 3 and 4 above.)

Note: The order between steps 1 and 2 is not critical (may be the other way round).

3.1.2 Accept link and establish virtual serial connection

This procedure refers to taking part in the following steps:

1. If requested by the remote device, take part in authentication procedure and, upon further request, turn on encryption.
2. Accept a new channel establishment indication from L2CAP.
3. Accept an RFCOMM session establishment on that channel.
4. Accept a new data link connection on the RFCOMM session. This may trigger a local request to authenticate the remote device and turn on encryption, if the user has required that for the emulated serial port being connected to (and authentication/encryption procedures have not already been carried out).

Note: steps 1 and 4 may be experienced as isolated events when there already exists an RFCOMM session to the remote device.

3.1.3 Register Service record in local SDP database

This procedure refers to registration of a service record for an emulated serial port (or equivalent) in the SDP database. This implies the existence of a Service Database, and the ability to respond to SDP queries.

All services/applications reachable through RFCOMM need to provide an SDP service record that includes the parameters necessary to reach the corresponding service/application, see Section 6.1. In order to support legacy applications running on virtual serial ports, the service registration must be done by some helper-application, which is aiding the user in setting up the port.

3.2 POWER MODE AND LINK LOSS HANDLING

Since the power requirements may be quite different for units active in the Serial Port profile, it is not required to use any of the power-saving modes. However, requests to use a low-power mode shall, if possible, not be denied.

If sniff, park, or hold mode is used, neither RFCOMM DLCs nor the L2CAP channel are released.

If a unit detects the loss of link, RFCOMM shall be considered having been shut down. The disconnect DLC and shutdown RFCOMM procedures referenced in Section 4 shall not be performed. Before communication on higher layers can be resumed, the Initialize RFCOMM session procedure has to be performed.

4 RFCOMM INTEROPERABILITY REQUIREMENTS

This section describes the requirements on RFCOMM in units complying with the Serial Port profile.

	Procedure	Ability to initiate		Ability to respond	
		DevA	DevB	DevA	DevB
1.	Initialize RFCOMM session	M1	X1	X1	M1
2.	Shutdown RFCOMM session	M	M	M	M
3.	Establish DLC	M	X1	X1	M
4.	Disconnect DLC	M	M	M	M
5.	RS232 control signals	C1	C1	M	M
6.	Transfer information	M	M	N/A1	N/A1
7.	Test command	X	X	M	M
8.	Aggregate flow control	C1	C1	M	M
9.	Remote Line Status indication	O	O	M	M
10.	DLC parameter negotiation	O	O	M	M
11.	Remote port negotiation	O	O	M	M

Table 4.1: RFCOMM capabilities

M1: The ability to have more than one RFCOMM session operational concurrently is optional in the RFCOMM protocol. Although support of concurrence is encouraged where it makes sense, this profile does not mandate support of concurrent RFCOMM sessions in either DevA or DevB.

X1: Within the execution of the roles defined in this profile, these abilities will not be used.

N/A1: Information transfer is unacknowledged in the RFCOMM protocol.

C1: Which flow control mechanism to use (per-DLC, aggregate, or both) is an implementation issue. But, if an implementation cannot guarantee that there will always be buffers available for data received, the ability to send either per-DLC flow control or aggregate flow control is mandatory.

Some of the procedures are further commented in subsections below.

4.1 RS232 CONTROL SIGNALS

According to TS 07.10 [5], section 5.4.6.3.7, all devices are required to send information on all changes in RS232 control signals with the Modem Status Command.

However, since RFCOMM can be used with an adaptation layer implementing any kind of API (not only virtual serial ports), it is optional to use all RS232 control signals except flow control (the RTR signal in TS 07.10 [5]). This signal can be mapped on RTS/CTS or XON/XOFF or other API mechanisms, which is an implementation issue.

Informative note: To provide interoperability between devices actually using all RS232 control signals, and devices not using them, the former type of implementation must set the states of the appropriate signals in APIs/connectors to suitable default values depending on RFCOMM DLC state. The implementation must not rely on receiving any RS232 control information from peer devices. The dependency on RFCOMM DLC state may mean that DSR/DTR as well as DCD are set to high level when an RFCOMM DLC has been established, and that the same signals are set to low level if the corresponding DLC is closed for any reason.

4.2 REMOTE LINE STATUS INDICATION

It is required to inform the other device of any changes in RS232 line status with the Remote Line Status indication command, see [5], section 5.4.6.3.10, if the local device relays information from a physical serial port (or equivalent) where overrun-, parity- or framing-errors may occur.

4.3 REMOTE PORT NEGOTIATION

DevA may inform DevB of RS232 port settings with the Remote Port Negotiation Command, directly before DLC establishment. See [5], section 5.4.6.3.9. There is a requirement to do so if the API to the RFCOMM adaptation layer exposes those settings (e.g. baud rate, parity).

DevB is allowed to send the Remote Port Negotiation command.

Informative note: the information conveyed in the remote port negotiation procedure is expected to be useful only in type II devices (with physical serial port) according to section 1.2 in RFCOMM [4], or if data pacing is done at an emulated serial port interface for any reason. RFCOMM as such will not artificially limit the throughput based on baud rate settings, see RFCOMM [4], chapter 2.

5 L2CAP INTEROPERABILITY REQUIREMENTS

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

	Procedure	Support in DevA/DevB
1.	Channel types	
	Connection-oriented channel	M
	Connectionless channel	X1
2.	Signalling	
	Connection Establishment	M
	Configuration	M
	Connection Termination	M
	Echo	M
	Command Rejection	M
3.	Configuration Parameter Options	
	Maximum Transmission Unit	M
	Flush Timeout	M
	Quality of Service	O

Table 5.1: L2CAP procedures

X1: Connectionless channel is not used within the execution of this profile, but concurrent use by other profiles/applications is not excluded.

5.1 CHANNEL TYPES

In this profile, only connection-oriented channels shall be used. This implies that broadcasts will not be used in this profile.

In the PSM field of the Connection Request packet, the value for RFCOMM defined in the Assigned Numbers document [8], section 3.2 must be used.

5.2 SIGNALLING

Only DevA may issue an L2CAP Connection Request within the execution of this profile. Other than that, the Serial Port Profile does not impose any additional restrictions or requirements on L2CAP signalling.

5.3 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the Serial Port Profile.

5.3.1 Maximum Transmission unit

This profile does not impose any restrictions on MTU sizes over the restrictions stated in L2CAP [3], section 6.1.

5.3.2 Flush Timeout

Serial Port data is carried over a reliable L2CAP channel. The flush timeout value shall be set to its default value 0xffff.

5.3.3 Quality of Service

Negotiation of Quality of Service is optional in this profile.

Recommendation: Implementations should try to keep an upper limit of 500 milliseconds on the latency incurred when going back from a low power mode to active mode.

6 SDP INTEROPERABILITY REQUIREMENTS

6.1 SDP SERVICE RECORDS FOR SERIAL PORT PROFILE

There are no SDP Service Records related to the Serial Port Profile in DevA.

The following table is a description of the Serial Port related entries in the SDP database of DevB. It is allowed to add further attributes to this service record.

Item	Definition	Type/Size	Value	AttributeID
ServiceClassIDList			Note1	0x0001
ServiceClass0	SerialPort / Note3	UUID	Note1	
ProtocolDescriptorList				0x0004
Protocol0	L2CAP	UUID	L2CAP /Note1	
Protocol1	RFCOMM	UUID	RFCOMM /Note1	
ProtocolSpecificParameter0	Server Channel	Uint8	N = server channel #	
ServiceName	Displayable text name	DataElement/ String	“COM5” / Note4	Note2

Table 6.1: SDP Service Record

Notes:

1. Defined in the Assigned Numbers document [8].
2. For national language support for all “displayable” text string attributes, an offset has to be added to the LanguageBaseAttributeIDList value for the selected language (see the SDP Specification [6], section 5.1.14 for details).
3. The ‘SerialPort’ class of service is the most generic type of service. Addition of other, more specific services classes are not excluded by this profile.
4. The ServiceName attribute value suggested here is merely an example; a helper application setting up a serial port may give the port a more descriptive name.

6.2 SDP PROCEDURES

To retrieve the service records in support of this profile, the SDP client entity in DevA connects and interacts with the SDP server entity in DevB via the SDP and L2CAP procedures presented in sections 5 and 6 of SDAP [7]. In accordance to SDAP, DevA plays the role of the LocDev, while DevB plays the role of the RemDev.

7 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

7.1 CAPABILITY OVERVIEW

In addition to the requirements on supported procedures stated in the Link Manager specification itself (see Section 3 in the Link Manager Protocol), this profile also requires support for Encryption both in DevA and DevB.

7.2 ERROR BEHAVIOR

If a unit tries to use a mandatory feature, and the other unit replies that it is not supported, the initiating unit shall send an LMP_detach with detach reason "unsupported LMP feature."

A unit shall always be able to handle the rejection of the request for an optional feature.

7.3 LINK POLICY

There are no fixed master-slave roles for the execution of this profile.

This profile does not state any requirements on which low-power modes to use, or when to use them. That is up to the Link Manager of each device to decide and request as seen appropriate, within the limitations of the latency requirement stated in Section 5.3.3.

8 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

8.1 CAPABILITY OVERVIEW

The following table lists all capabilities on the LC level, and the extra requirements added to the ones in the baseband specification by this profile.

	Capabilities	Support in DevA	Support in DevB
1.	Inquiry		X1
2.	Inquiry scan	X1	
3.	Paging		X1
4.	Page scan		
A	Type R0	X1	
B	Type R1	X1	
C	Type R2	X1	
5.	Packet types		
A	ID packet		
B	NULL packet		
C	POLL packet		
D	FHS packet		
E	DM1 packet		
F	DH1 packet		
G	DM3 packet		
H	DH3 packet		
I	DM5 packet		
J	DH5 packet		
K	AUX packet	X1	X1
L	HV1 packet		
M	HV2 packet		
N	HV3 packet		
O	DV packet		
6.	Inter-piconet capabilities		

Table 8.1: Baseband/LC capabilities

	Capabilities	Support in DevA	Support in DevB
7.	Voice codec		
A	A-law		
B	µ-law		
C	CVSD		

Table 8.1: Baseband/LC capabilities

X1: These capabilities are not used within the execution of this profile, but concurrent use by other profiles/applications is not excluded.

8.2 INQUIRY

When inquiry is invoked in DevA, it shall use the General Inquiry procedure, see GAP [9], Section 6.1.

Only DevA may inquire within the execution of this profile.

8.3 INQUIRY SCAN

For inquiry scan, (at least) the GIAC shall be used, according to one of the discoverable modes defines in GAP [9], Section 4.1.2. and Section 4.1.3. That is, it is allowed to only use the Limited discoverable mode, if appropriate for the application(s) residing in DevB.

In the DevB INQUIRY RESPONSE messages, the Class of Device field will not contain any hint as to whether DevB is engaged in the execution of the Serial Port Profile or not. (This is due to the fact the generalized Serial Port service for legacy applications delivered by this profile does not fit within any of the major Service Class bits in the Class Of Device field definition.)

8.4 PAGING

Only DevA may page within the execution of this profile. The paging step will be skipped in DevA when execution of this profile begins when there already is a baseband connection between DevA and DevB. (In such a case the connection may have been set up as a result of previous paging by DevB.)

8.5 ERROR BEHAVIOR

Since most features on the LC level have to be activated by LMP procedures, errors will mostly be caught at that layer. However, there are some LC procedures that are independent of the LMP layer, e.g. inquiry or paging. Misuse of such features is difficult or sometimes impossible to detect. There is no mechanism defined to detect or prevent such improper use.

9 REFERENCES

[1] Bluetooth Special Interest Group, Bluetooth baseband specification

[2] Bluetooth Special Interest Group, Link Manager Protocol

[3] Bluetooth Special Interest Group, L2CAP Specification

[4] Bluetooth Special Interest Group, RFCOMM with TS 07.10

[5] ETSI, TS 101 369 (GSM 07.10) version 6.3.0

[6] Bluetooth Special Interest Group, Service Discovery Protocol (SDP)

[7] Bluetooth Special Interest Group, Service Discovery Application Profile

[8] Bluetooth Special Interest Group, Bluetooth Assigned Numbers
http://www.bluetooth.org/assigned-numbers.htm

[9] Bluetooth Special Interest Group, Generic Access Profile

10 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles....170
Figure 2.1: Protocol model....171
Figure 2.2: Serial Port profile, example with two notebooks. 172

11 LIST OF TABLES

Table 3.1: Application layer procedures 174
Table 4.1: RFCOMM capabilities 177
Table 5.1: L2CAP procedures 179
Table 6.1: SDP Service Record 181
Table 8.1: Baseband/LC capabilities 184

Part K:6

HEADSET PROFILE

This profile defines the requirements for Bluetooth devices necessary to support the Headset use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Headset use case.

CONTENTS

1 Introduction 193
1.1 Scope 193
1.2 Profile Dependencies 193
1.3 Symbols and conventions 194
1.3.1 Requirement status symbols 194
1.4 Signalling diagram conventions 195
2 Profile Overview 196
2.1 Profile stack 196
2.2 Configuration and roles 197
2.3 User requirements and scenarios 198
2.4 Profile fundamentals 198
2.5 Conformance 199
3 Application layer 200
4 Headset Control Interoperability Requirements 201
4.1 Introduction 201
4.2 Audio Gateway Initiated ACL Connection Establishment 201
4.3 Headset Initiated ACL Connection Establishment 202
4.4 Audio connection release 203
4.5 Audio connection transfer 204
4.5.1 Audio connection transfer from AG to HS 204
4.5.2 Audio connection transfer from HS to AG 205
4.6 Remote audio volume control 205
4.7 AT Commands and Result Codes 207
4.7.1 General 207
4.7.2 AT capabilities re-used from V.250 207
4.7.3 Bluetooth-defined AT capabilities 208
4.8 Lower layer handling 208
4.8.1 Connection handling without PARK mode 209
4.8.1.1 Connection establishment 209
4.8.1.2 Connection release 209
4.8.2 Connection handling with PARK mode 209
4.8.2.1 Connection establishment 209
4.8.2.2 Connection release 210

5 Serial Port Profile 211
5.1 RFCOMM Interoperability Requirements 211
5.2 L2CAP Interoperability Requirements 211
5.3 SDP Interoperability Requirements 212
5.4 Link Manager (LM) Interoperability Requirements 213
5.5 Link Control (LC) Interoperability Requirements 214
5.5.1 Class of Device 214
6 Generic Access Profile 215
6.1 Modes 215
6.2 Security aspects 215
6.3 Idle mode procedures 215
7 References 216
8 List of Figures 217
9 List of Tables 218

1 INTRODUCTION

1.1 SCOPE

This Headset profile defines the protocols and procedures that shall be used by devices implementing the usage model called ‘Ultimate Headset’. The most common examples of such devices are headsets, personal computers, and cellular phones.

The headset can be wirelessly connected for the purposes of acting as the device’s audio input and output mechanism, providing full duplex audio. The headset increases the user’s mobility while maintaining call privacy.

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

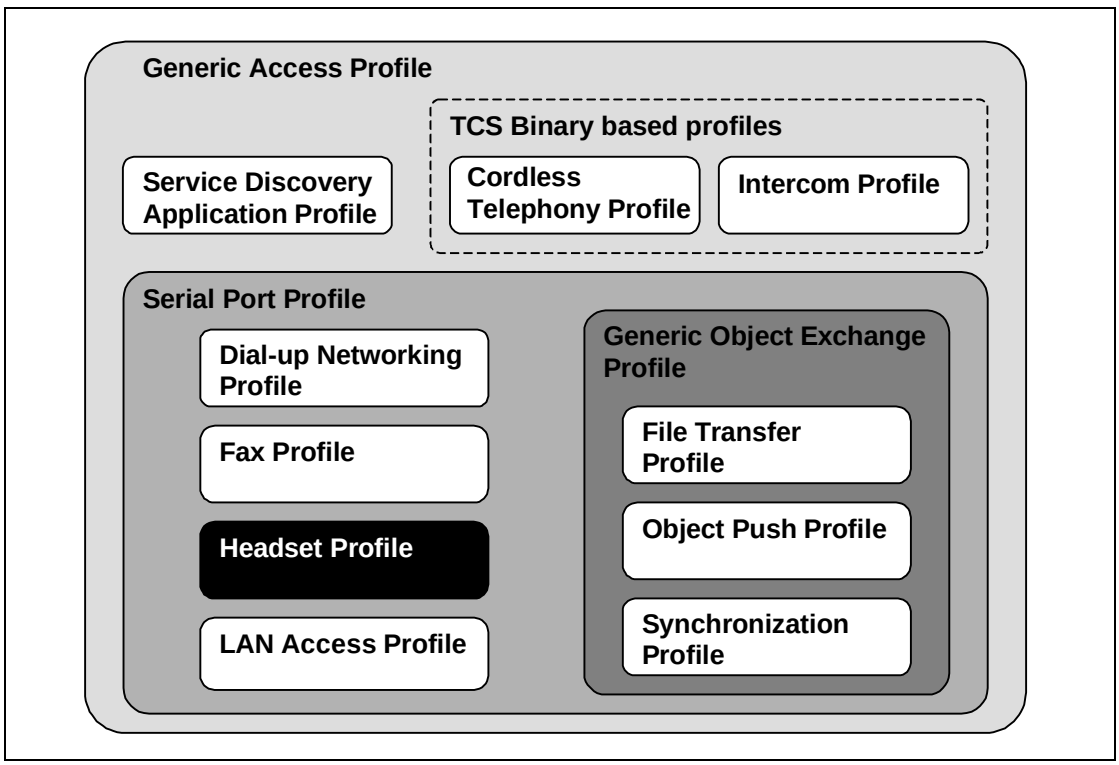


Figure 1.1: Bluetooth Profiles

As indicated in the figure, the Headset profile is dependent upon both the Serial Port Profile and the Generic access profile – details are provided in Section 5, “Serial Port Profile,” on page219 and Section 6, “Generic Access Profile,” on page223 .

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

- ‘M’ for mandatory to support
- ‘O’ for optional to support
- ‘X’ for excluded (used for capabilities that may be supported by the unit but shall never be used in this use case)
- ‘C’ for conditional to support
- ‘N/A’ for not applicable (in the given context it is impossible to use this capability)

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices in this use case. Therefore, these features shall never be activated while a unit is operating as a unit within this use case.

1.4 SIGNALLING DIAGRAM CONVENTIONS

The following arrows are used in diagrams describing procedures:

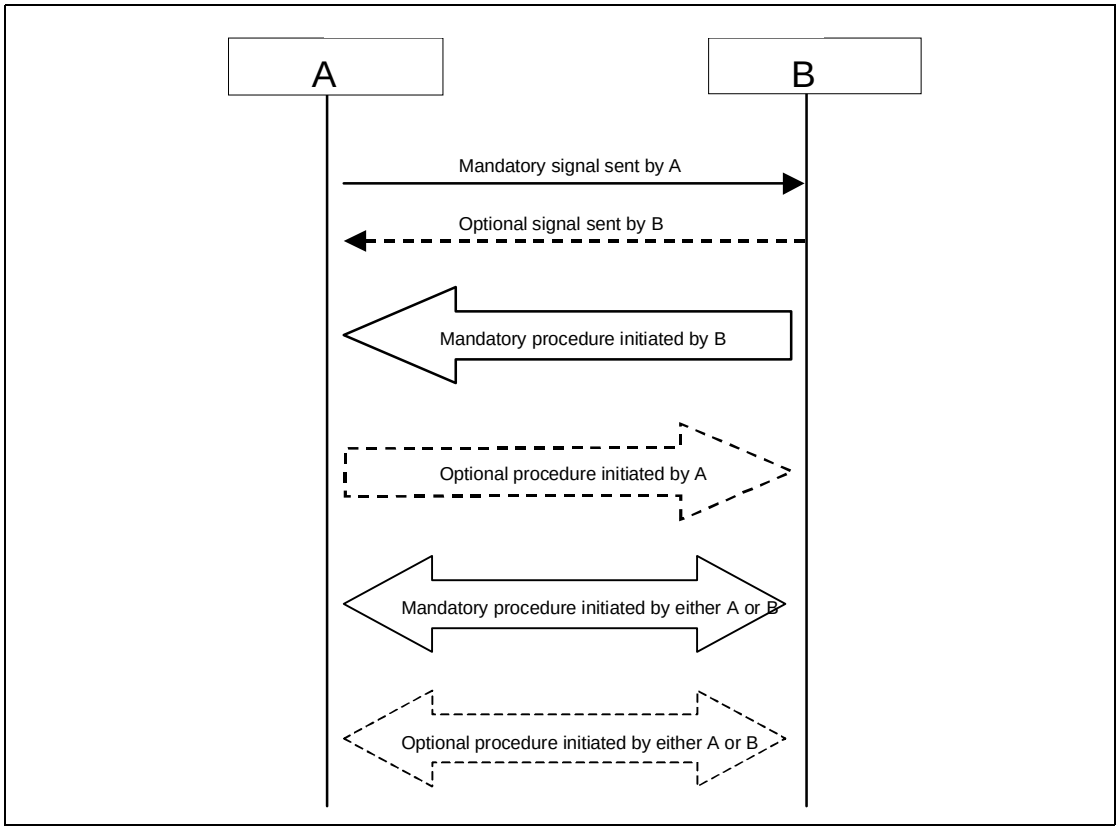


Figure 1.2: Arrows used in signalling diagrams

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

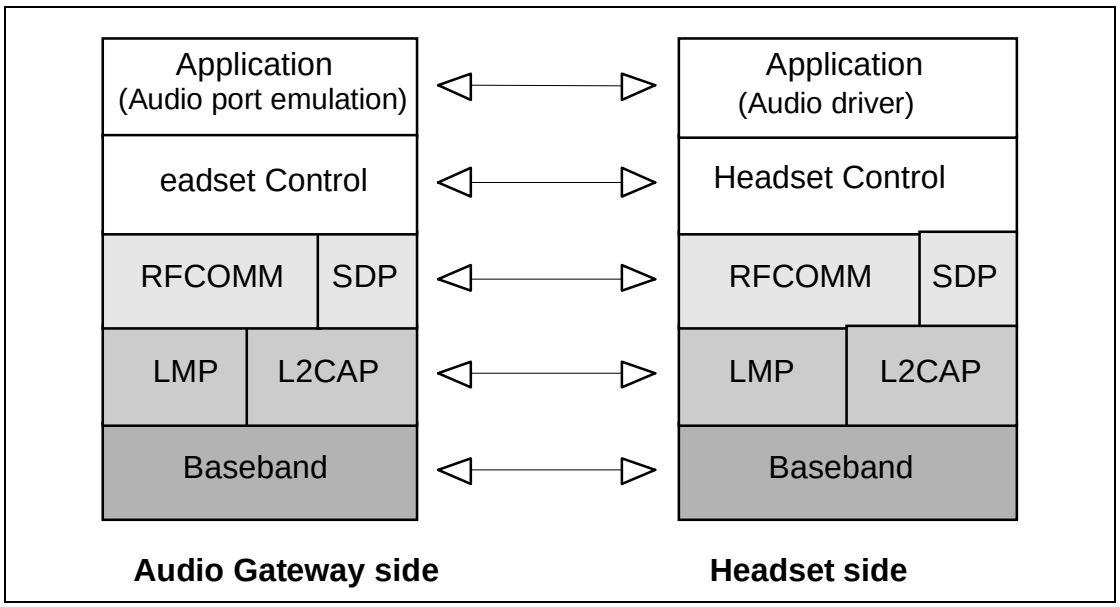


Figure 2.1: Protocol model

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth adaptation of GSM TS 07.10 [5]. SDP is the Bluetooth Service Discovery Protocol. Headset Control is the entity responsible for headset specific control signalling; this signalling is AT command based.

Note: although not shown in the model above, it is assumed by this profile that Headset Control has access to some lower layer procedures (for example SCO link establishment).

The audio port emulation layer shown in Figure 2.1 is the entity emulating the audio port on the cellular phone or PC, and the audio driver is the driver software in the headset.

For the shaded protocols/entities in Figure 2.1, the Serial Port Profile is used as base standard. For these protocols, all requirements stated in the Serial Port Profile apply except in those cases where this profile explicitly states deviations.

2.2 CONFIGURATION AND ROLES

The figures below show two typical configurations of devices for which the Headset profile is applicable:

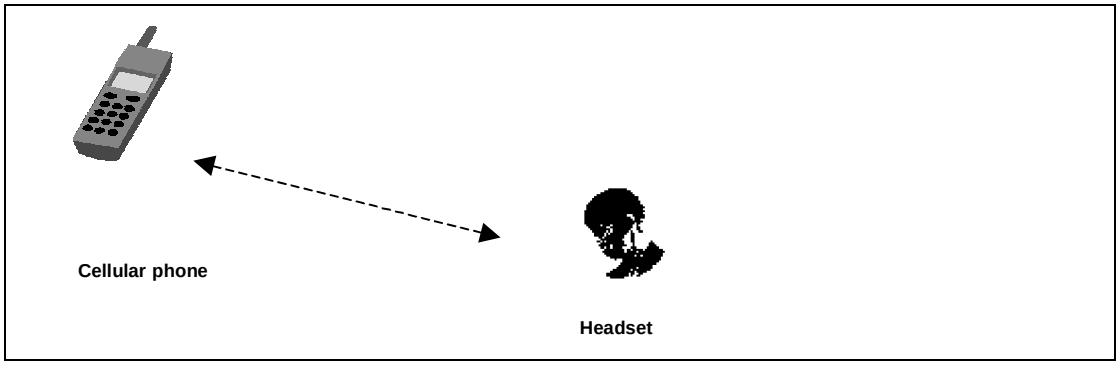


Figure 2.2: Headset profile, example with cellular phone

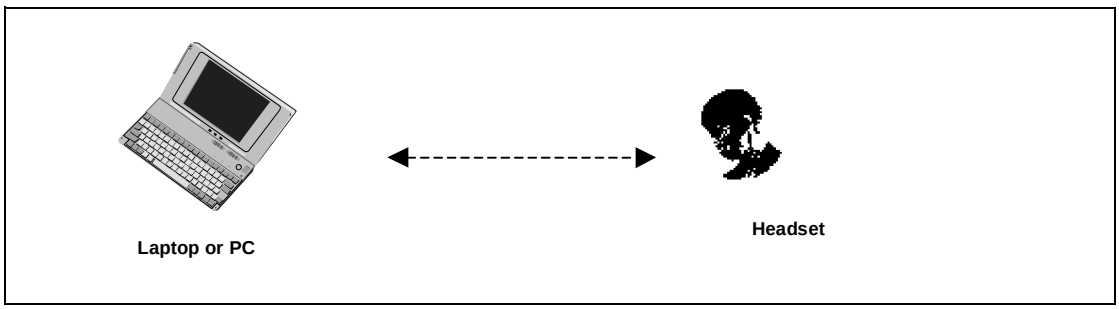


Figure 2.3: Headset profile, example with personal computer

The following roles are defined for this profile:

Audio Gateway **(AG)** – This is the device that is the gateway of the audio, both for input and output. Typical devices acting as Audio Gateways are cellular phones and personal computer.

Headset **(HS)** – This is the device acting as the Audio Gateway's remote audio input and output mechanism.

These terms are in the rest of this document only used to designate these roles.

2.3 USER REQUIREMENTS AND SCENARIOS

The Headset profile defines the protocols and procedures that shall be used by devices implementing the use case called ‘Ultimate Headset’.

The following restrictions apply to this profile:

a) For this profile, it is assumed that the ultimate headset use case is the only use case active between the two devices;

b) The profile mandates the usage of CVSD for transmission of audio (for the Bluetooth part). The resulting audio is monophonic, with a quality that – under normal circumstances – will not have perceived audio degradation.

c) Between headset and audio gateway, only one audio connection at a time is supported;

d) The audio gateway controls the SCO link establishment and release. The headset directly connects (disconnects) the internal audio streams upon SCO link establishment (release). Valid speech exists on the SCO link in both directions, once established;

e) The profile offers only basic interoperability – for example, handling of multiple calls at the audio gateway is not included;

f) The only assumption on the headset’s user interface is the possibility to detect a user initiated action (e.g. pressing a button).

2.4 PROFILE FUNDAMENTALS

A headset may be able to use the services of audio gateway without the creation of a secure connection. It is up to the user, if he/she wants to enforce security on devices that support authentication and/or encryption in the execution of this profile. If baseband authentication and/or encryption is desired, the two devices have to create a secure connection, using the GAP authentication procedure as described in Section 5.1 of the Generic Access profile. This procedure may then include entering a PIN code, and will include creation of link keys. In most cases, the headset will be a device with a limited user interface, so the (fixed) pin code of the headset will be used during the GAP authentication procedure.

A link has to be established when a call is initiated or received. Normally, this requires paging of the other device but, optionally, it may require unparking.

There are no fixed master/slave roles.

The audio gateway and headset provide serial port emulation. For the serial port emulation, RFCOMM is used. The serial port emulation is used to transport the user data including modem control signals and AT commands from the headset to the audio gateway. AT commands are parsed by the audio gateway and responses are sent to the headset.

2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated as mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 APPLICATION LAYER

This section describes the feature requirements on units complying with the Headset profile.

Table 3.1 shows the feature requirements made by this profile.

	Feature	Support in HS	Support in AG
1.	Incoming audio connection	M	M
2.	Outgoing audio connection	M	O
3.	Audio connection transfer	M	M
4.	Remote audio volume control	O	O

Table 3.1: Application layer procedures

In the table above, incoming and outgoing shall be interpreted from the headset (HS) point of view.

Table 3.2 maps each feature to the procedures used for that feature. All procedures are mandatory if the feature is supported.

	Feature	Procedure	Ref.
1.	Incoming audio connection	Incoming audio connection establishment	4.2
		Audio connection release	4.4
2.	Outgoing audio connection	Outgoing audio connection establishment	4.3
		Audio connection release	4.4
3.	Audio connection transfer	Audio connection transfer	4.5
4.	Remote audio volume control	Remote audio volume control	4.6

Table 3.2: Application layer feature to procedure mapping

4 HEADSET CONTROL INTEROPERABILITY REQUIREMENTS

4.1 INTRODUCTION

The interoperability requirements for the Headset Control entity are completely contained in this chapter. Section 4.2 until 4.6 specify the requirements for the procedures directly relating to the application layer features.

Section 4.7 specifies the AT commands and results codes used for signalling purposes.

Section 4.8 specifies how the layers beneath the Headset Control entity are used to establish and release a connection.

4.2 AUDIO GATEWAY INITIATED ACL CONNECTION ESTABLISHMENT

Upon an internal or user generated event, the AG will initiate connection establishment (see Section 4.8), and once the connection is established, will send an unsolicited result code RING to alert the user (if necessary). The RING may be repeated for as long as the connection establishment is pending.The SCO link establishment can take place anytime after the ACL connection establishment

Optionally, the AG may provide an in-band ringing tone[1]. In this case, first SCO link establishment takes place. In this case, the SCO link establishment takes place first.

1. The in-band ringing tone is used to alert the user in the headset earpiece when the user is wearing the headset on his head.

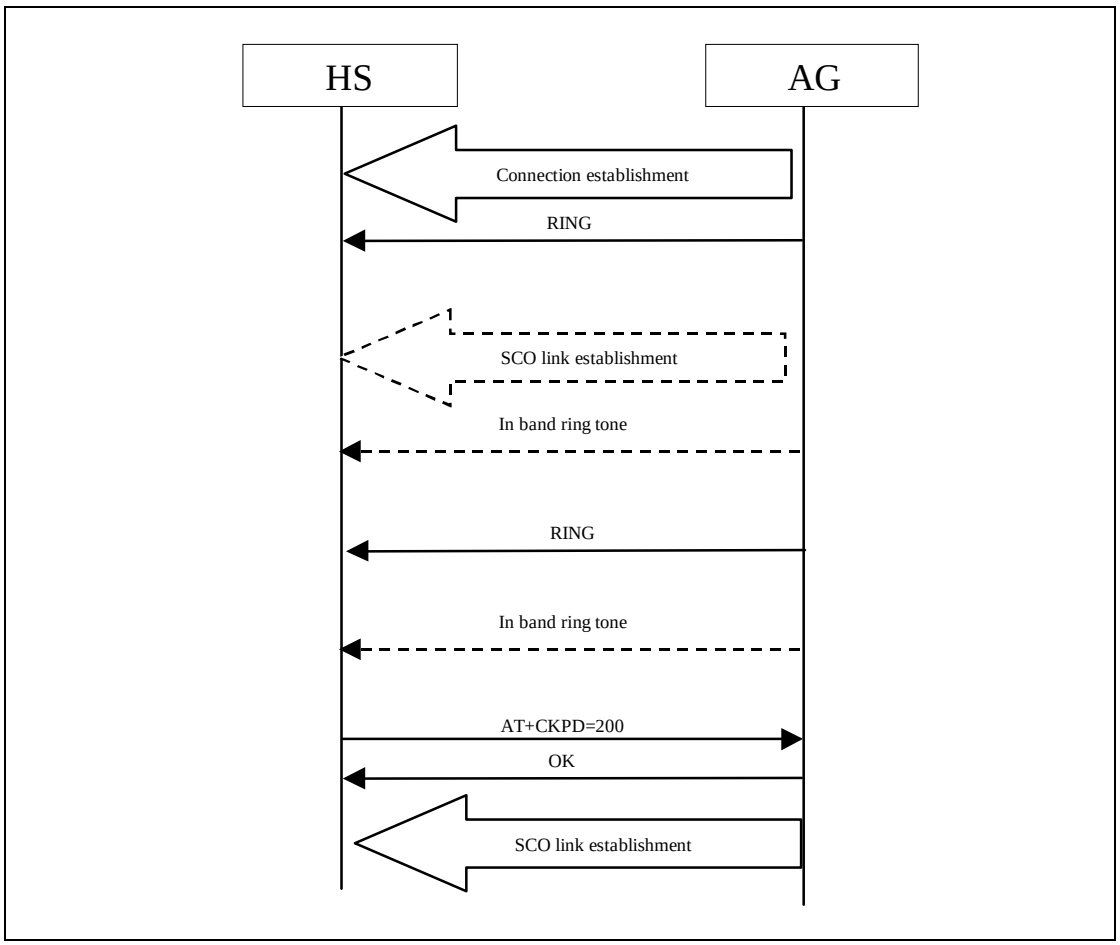


Figure 4.1: Incoming audio connection establishment

In cases where the user is alerted, the user accepts the incoming audio connection by performing a user-initiated action (e.g. pressing a button on the headset). When the user initiates the action, the HS will send the AT+CKPD command (see Section 4.7) to the AG, whereupon the AG shall establish the SCO link, if not already established earlier.

4.3 HEADSET INITIATED ACL CONNECTION ESTABLISHMENT

A headset initiated ACL connection is established (see Section 4.8) on the HS by a user-initiated action (e.g. pressing a button on the headset). Upon connection establishment, the HS shall send the AT+CKPD command to the AG.

The AG may initiate a SCO connection after the completion of the ACL connection establishment before receiving the AT+CKPD command from the HS. This may be desirable when the AG is a mobile phone. In all cases, the AG is responsible for establishing the SCO link if needed. Further internal actions may be needed on the AG to internally establish and/or route the audio stream to the HS[2].

In the figure, the SCO link connection should be possible prior to receiving the AT+CKPD message.

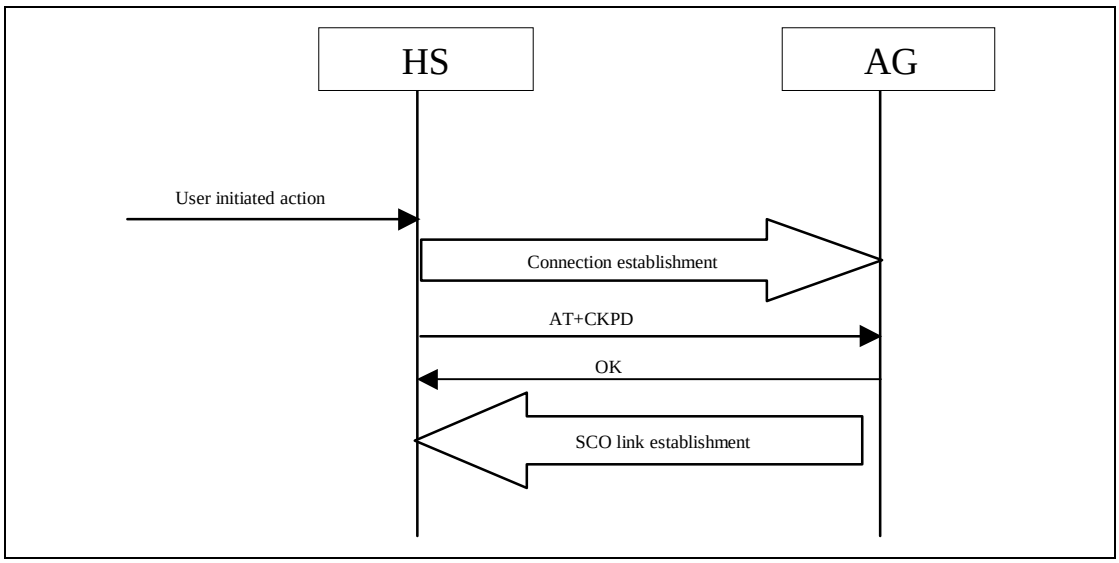


Figure 4.2: Outgoing audio connection establishment

4.4 AUDIO CONNECTION RELEASE

A call can be terminated either on the HS or on the AG. On the HS based upon the button being pushed, on the AG based upon internal actions or user intervention.

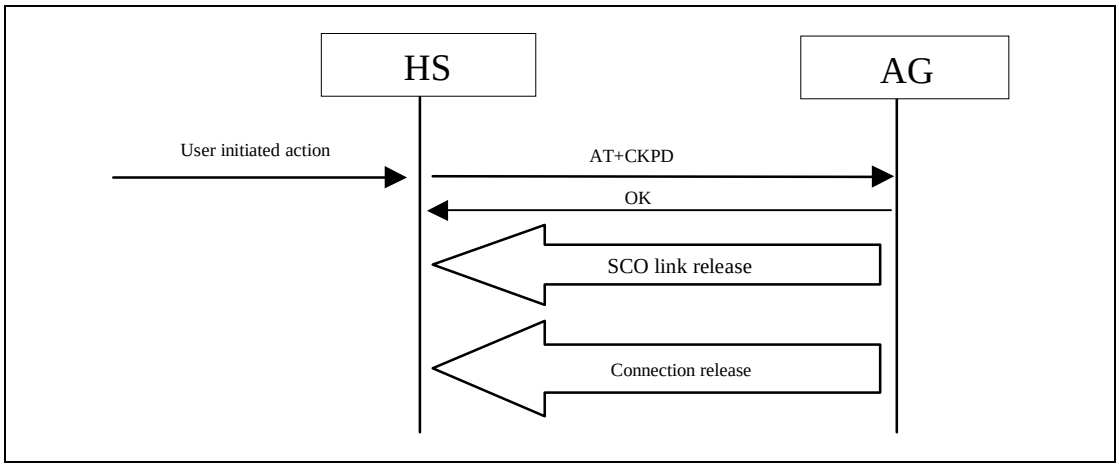


Figure 4.3: Audio connection release – HS initiated

2. For a cellular phone a cellular call may need to be established, e.g. using last dialled number, pre-programmed number. For a personal computer this e.g. relates to playing a .wav file, or audio CD.

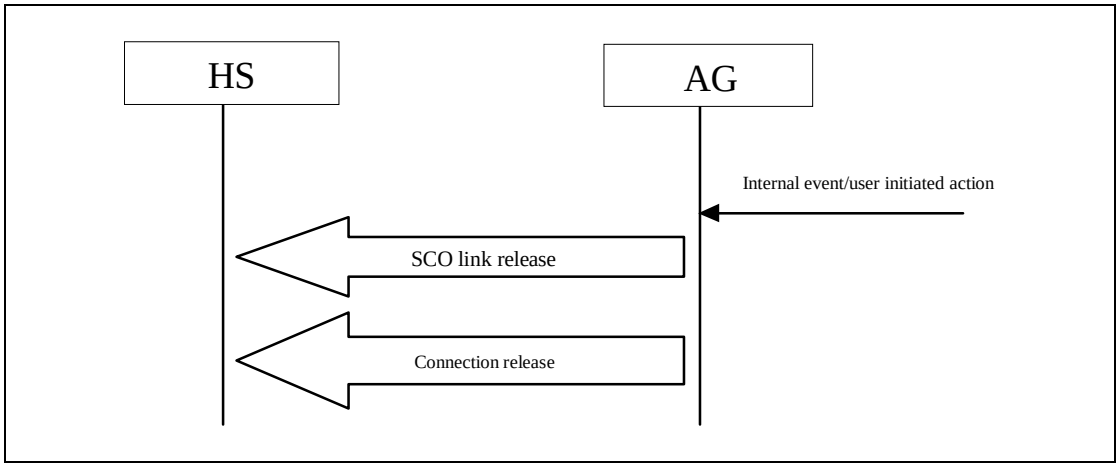


Figure 4.4: Audio connection release – AG initiated

Irrespective of the initiating side, the AG is responsible for releasing the connection (see Section 4.8).

4.5 AUDIO CONNECTION TRANSFER

An audio connection can be transferred from AG to HS or from HS to AG. The connection is transferred to the device initiating the transfer.

4.5.1 Audio connection transfer from AG to HS

The audio connection transfer from AG to HS is initiated by a user action on the HS side, which results in an AT+CKPD command being sent to the AG.

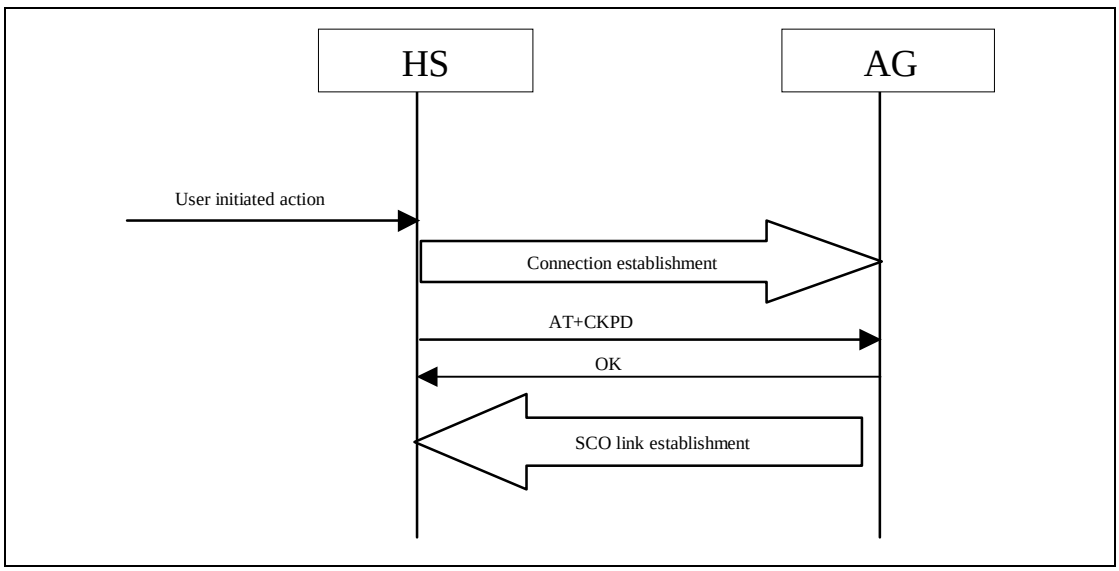


Figure 4.5: Audio connection transfer from AG to HS

4.5.2 Audio connection transfer from HS to AG

The audio connection transfer from HS to AG is initiated by a user action on the AG.

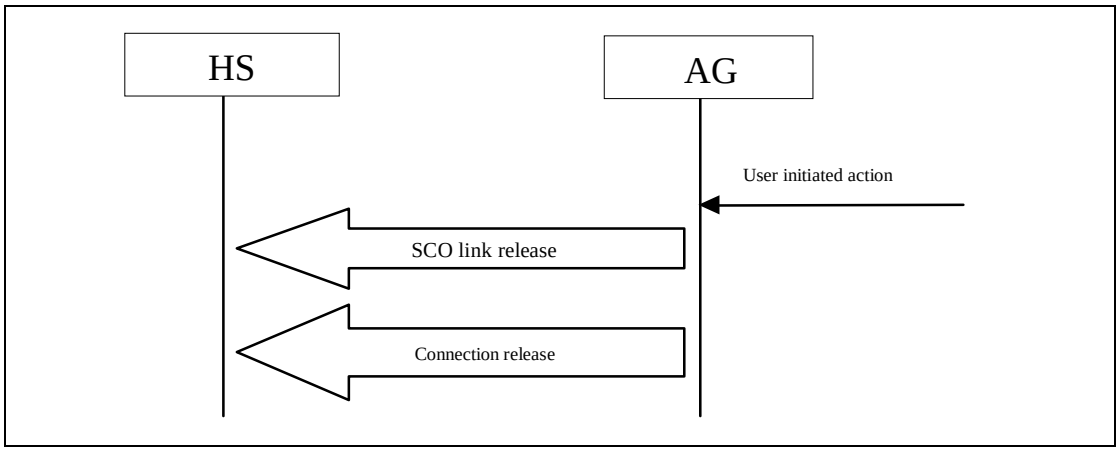


Figure 4.6: Audio connection transfer from HS to AG

4.6 REMOTE AUDIO VOLUME CONTROL

The AG can control the gain of the microphone and speaker of the HS by sending unsolicited result codes +VGM and +VGS respectively. There is no limit to the amount and order of result codes, as long as there is an active audio connection ongoing. When supporting the remote audio volume control, an implementation is not mandated to support both the control of the microphone volume and speaker volume.

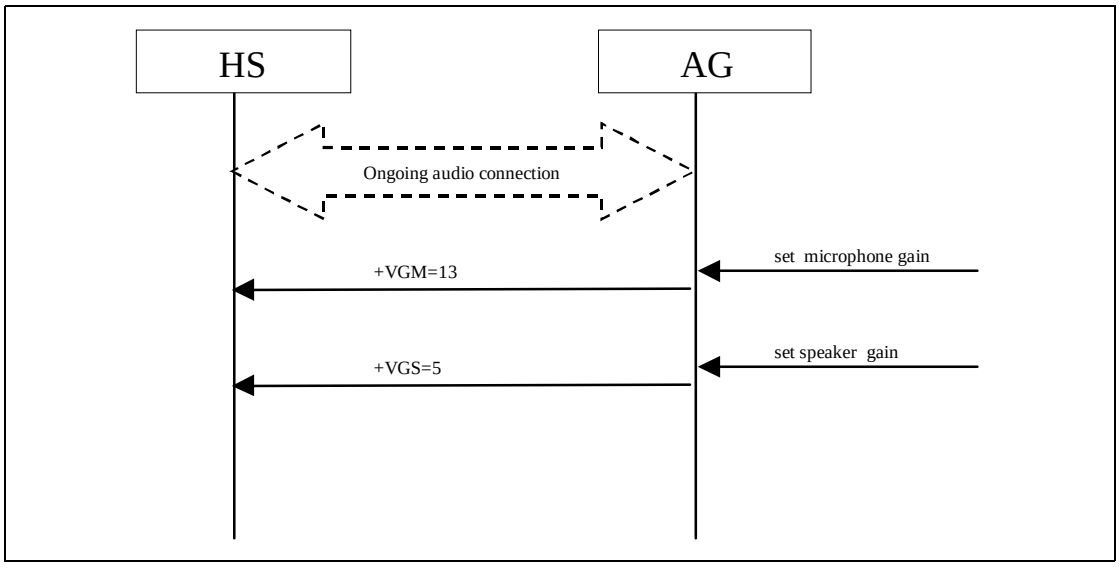


Figure 4.7: Audio volume control – example flow

Both the speaker and microphone gain are represented as parameter to the +VGS and +VGM, on a scale from 0 to 15. The values are absolute values, relating to a particular (implementation-dependent) volume level controlled by the HS.

The HS may store the VGS and VGM settings at connection release, to restore the volume levels at the next connection establishment. At connection establishment, the HS shall inform the AG of the (restored) volume levels using the AT commands +VGS and +VGM. In case physical mechanisms(buttons, dials etc.)means are implemented on the HS to control the volume levels, the HS shall also use the AT commands +VGS and +VGM to inform the AG of any changes in the volume levels.

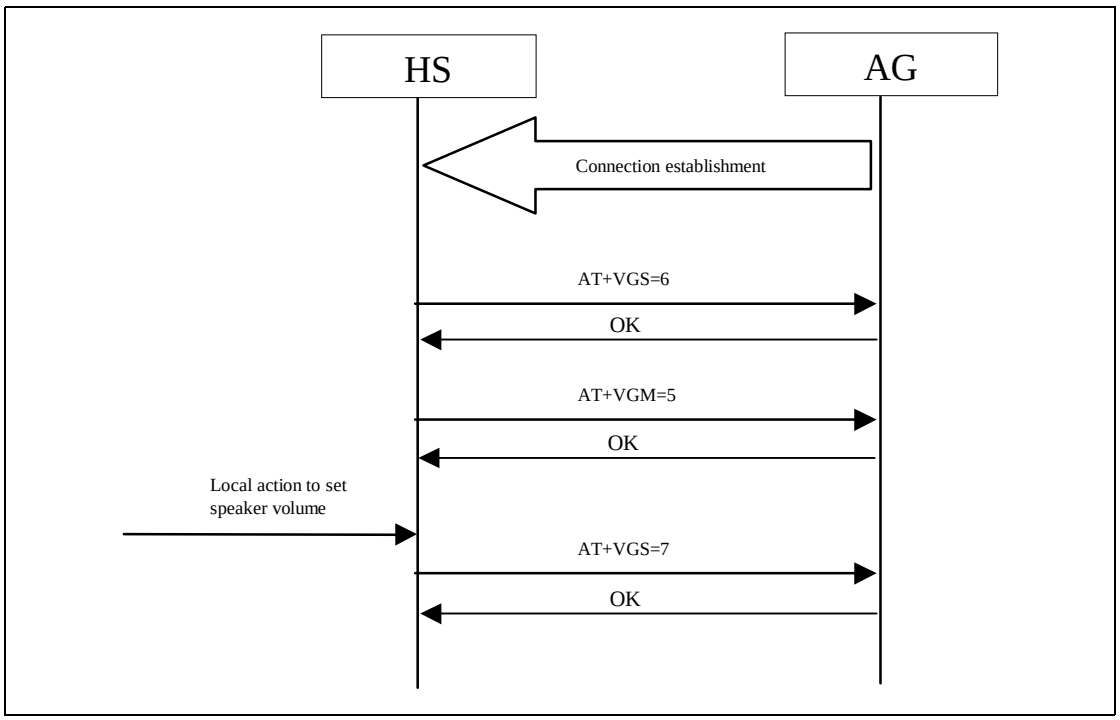


Figure 4.8: Volume level synchronization – example flow

4.7 AT COMMANDS AND RESULT CODES

4.7.1 General

The command line termination character shall be carriage return (IA5 0/13). The response formatting character shall be line feed (IA5 0/10). The AG shall not echo command characters[1]. The AG shall transmit result codes, using the verbose (rather than numeric) format.

1This is the opposite of the default recommended by V.250[1]

The format for a command from the HS to the AG is thus:

AT<cmd>=<value><cr>

If the command is processed successfully, the resulting response from the AG to the HS is:

<cr><lf>OK<cr><lf>

If the command is not processed successfully, the resulting response from the AG to the HS is:

<cr><lf>ERROR<cr><lf>

The format for an unsolicited result code (such as RING) from the AG to the HS is:

<cr><lf><result code><cr><lf>

The headset profile uses a subset of AT commands and result codes from existing standards. These are listed in Section 4.7.2. For those AT commands and result codes where no existing commands applied, Section 4.7.3 defines additional ones.

4.7.2 AT capabilities re-used from V.250

The mandatory set of AT commands and unsolicited result codes are indicated in Table 4.1 below.

AT capability	Description
RING	The Incoming call indication of V.250 [1], Section 6.3.4.

Table 4.1: Mandatory AT capabilities

4.7.3 Bluetooth-defined AT capabilities

Optionally, the AT capabilities as indicated in Table 4.2 may be supported.

AT capability	Syntax	Description	Values
Microphone gain	+VGM=<gain>	Unsolicited result code issued by the AG to set the microphone gain of the HS. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Speaker gain	+VGS=<gain>	Unsolicited result code issued by the AG to set the speaker gain of the HS. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
AT capability	**Syntax**	**Description**	**Values**
Microphone gain level report	+VGM=<gain>	Command issued by the HS to report the current microphone gain level setting to the AG. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Speaker gain level indication report	+VGS=<gain>	Command issued by the HS to report the current speaker gain level setting to the AG. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Headset button press	+CKPD=200	Command issued by HS to indicate that the button has been pressed	

Table 4.2: Optional AT capabilities

4.8 LOWER LAYER HANDLING

This section describes how the layers below the Headset Control entity are used to establish and release a connection.Section 4.8.1 describes how connections are handled when the PARK mode is not supported. Section 4.8.2 describes how connections are handled when the PARK mode is supported.

4.8.1 Connection handling without PARK mode

4.8.1.1 Connection establishment

Both the HS and the AG can initiate connection establishment. If there is no RFCOMM session between the AG and the HS, the initiating device shall first initialize RFCOMM. Connection establishment shall be performed as described in Section 7.3 of GAP and Section 3 of SPP.

4.8.1.2 Connection release

When the audio connection is released, the connection may be released as well. The AG always initiates connection release.

4.8.2 Connection handling with PARK mode

4.8.2.1 Connection establishment

If the PARK mode is supported, the connection is established once (e.g. on the first request for an audio connection). Later, when an audio connection is required, the parked device is unparked. In this section, for correct interpretation of the flows given in Section 4.2 to 4.6, the connection establishment is referred to as *initial* connection establishment, whereas the unparking is referred to as connection establishment.

Initial connection establishment shall be performed as described in Section 7.3 of GAP and Section 3 of SPP. Both sides may initiate the initial connection establishment. After initial connection establishment, the park mode is activated.

In Figure 4.9 the behavior is described in case an audio connection needs to be established – the parked device will be unparked. The unpark can be initiated from either side, depending where the request for an audio connection originated. If the PARK mode is used, neither RFCOMM DLCs nor the L2CAP channel is released.

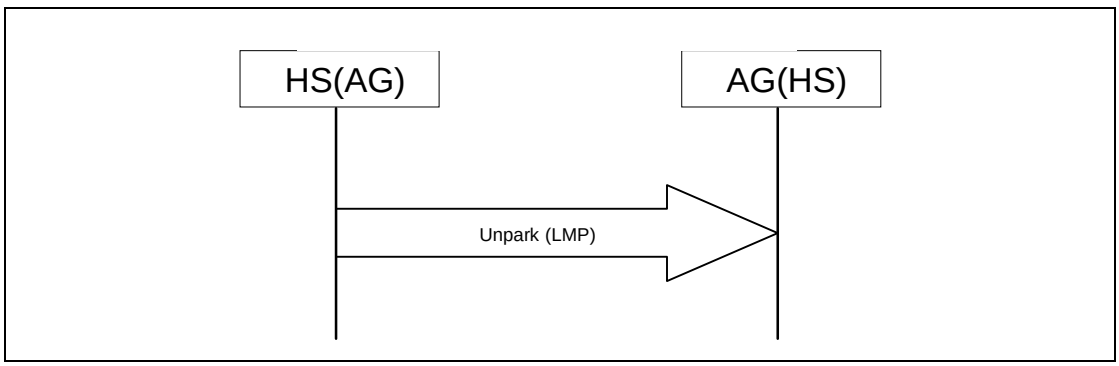


Figure 4.9: Connection establishment – Unparking a parked device

4.8.2.2 Connection release

When the audio connection is released, the connection is parked again, as indicated in Figure 4.10.

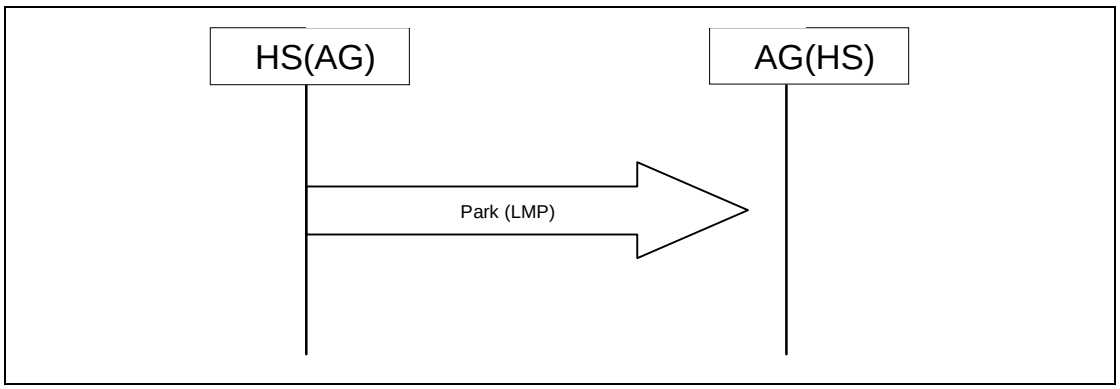


Figure 4.10: Connection release – Parking

When the audio connection is released, the complete connection may alternatively be released. The AG always initiates connection release.

5 SERIAL PORT PROFILE

This profile requires compliance with the Serial Port Profile. The following text together with the associated sub-clauses defines the requirements with regard to this profile, in addition to the requirements as defined in the Serial Port Profile.

As with the headset profile, both the AG and the HS can initiate connection establishment. For the purposes of reading the Serial Port Profile, both the AG and the HS can assume the role of Device A and B.

5.1 RFCOMM INTEROPERABILITY REQUIREMENTS

For the RFCOMM layer, no additions to the requirements as stated in the Serial Port Profile Section 4 shall apply.

5.2 L2CAP INTEROPERABILITY REQUIREMENTS

For the L2CAP layer, no additions to the requirements as stated in the Serial Port Profile Section 5 shall apply.

5.3 SDP INTEROPERABILITY REQUIREMENTS

This profile defines following service records for the headset and the audio gateway respectively.

The codes assigned to the mnemonics used in the Value column as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column) can be found in the Bluetooth Assigned Numbers section.

Item	Definition	Type	Value	AttrID	Status	Default
ServiceClassIDList					M	
ServiceClass0		UUID	Headset		M	
ServiceClass1		UUID	Generic Audio		M	
ProtocolDescriptorList					M	
Protocol0		UUID	L2CAP		M	
Protocol1		UUID	RFCOMM		M	
Protocol Specific Parameter0	Server Channel	Uint8	N=server channel #		M	
BluetoothProfile DescriptorList					O	
Profile0	Supported Profiles	UUID	Headset		M	Headset
Param0	Profile Version	Uint16	0x0100*		M	0x0100
ServiceName	Display-able Text name	String	Service-provider defined		O	‘Headset’
Remote audio volume control		Bool-ean	True/False		O	False

Table 5.1: Service Record for Headset

*. Indicating version 1.0

Item	Definition	Type	Value	AttrID	Status	Default
ServiceClassIDList					M	
ServiceClass0		UUID	Headset Audio Gateway		M	
ServiceClass1		UUID	Generic Audio		M	
ProtocolDescriptorList					M	
Protocol0		UUID	L2CAP		M	
Protocol1		UUID	RFCOMM		M	
Protocol Specific Parameter0	Server Channel	Uint8	N=server channel #		M	
BluetoothProfile DescriptorList						
Profile0	Supported Profile	UUID	Headset		M	Headset
Param0	Profile Version	Uint16	0x0100*		M	0x0100
ServiceName	Display-able Text name	String	Service-provider defined		O	'Voice gateway'

Table 5.2: Service Record for the Audio Gateway

*. Indicating version 1.0

5.4 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

In addition to the requirements for the Link Manager as stated in the "Serial Port Profile" on page 171, this profile mandates support for SCO links, in both the HS and AG.

5.5 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

In the table below, changes to the support status as listed in the Serial Port Profile, Section 8, Table 8.1 on page 191 are listed.

	Capability	Support in AG	Support in HS
1.	Inquiry		X
2.	Inquiry scan	X	
3.	Paging		
4.	Page scan		
A	Type R0		
B	Type R1		
C	Type R2		
7.	Voice codec		
C	CVSD	M	M

Table 5.3: LC capabilities

5.5.1 Class of Device

A device which is active in the HS role shall, in the Class of Device field:

1. Set the bit ‘Audio’ in the Service Class field
2. Indicate ‘Audio’ as Major Device class
3. Indicate “Headset” as the Minor Device class

An inquiring AG may use this to filter the inquiry responses.

6 GENERIC ACCESS PROFILE

This section defines the support requirements for the capabilities as defined in Generic Access Profile.

6.1 MODES

The table shows the support status for Modes within this profile.

	Procedure	Support in HS	Support in AG
1	Discoverability modes		
	Non-discoverable mode	M	N/A
	Limited discoverable mode	O	N/A
	General discoverable mode	M	N/A
2	Connectability modes		
	Non-connectable mode	N/A	N/A
	Connectable mode	M	M
3	Pairing modes		
	Non-pairable mode	O	O
	Pairable mode	O	O

Table 6.1: Modes

6.2 SECURITY ASPECTS

No changes to the requirements as stated in the Generic Access Profile.

6.3 IDLE MODE PROCEDURES

The table shows the support status for Idle mode procedures within this profile.

	Procedure	Support in HS	Support in AG
1	General inquiry	N/A	M
2	Limited inquiry	N/A	O
3	Name discovery	N/A	O
4	Device discovery	N/A	O
5	Bonding	O(	O(

Table 6.2: Idle mode procedures

7 REFERENCES

[1] International Telecommunication Union, “ITU-T Recommendation V.250”

[2] ETS 300 916 (GSM 07.07) version 5.6.0

8 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 193
Figure 1.2: Arrows used in signalling diagrams 195
Figure 2.1: Protocol model 196
Figure 2.2: Headset profile, example with cellular phone 197
Figure 2.3: Headset profile, example with personal computer 197
Figure 4.1: Incoming audio connection establishment 202
Figure 4.2: Outgoing audio connection establishment 203
Figure 4.3: Audio connection release – HS initiated 203
Figure 4.4: Audio connection release – AG initiated 204
Figure 4.5: Audio connection transfer from AG to HS 204
Figure 4.6: Audio connection transfer from HS to AG 205
Figure 4.7: Audio volume control – example flow 205
Figure 4.8: Volume level synchronization – example flow 206
Figure 4.9: Connection establishment – Unparking a parked device 209
Figure 4.10: Connection release – Parking 210

9 LIST OF TABLES

Table 3.1: Application layer procedures 200
Table 3.2: Application layer feature to procedure mapping 200
Table 4.1: Mandatory AT capabilities 207
Table 4.2: Optional AT capabilities 208
Table 5.1: Service Record for Headset 212
Table 5.2: Service Record for the Audio Gateway 213
Table 5.3: LC capabilities 214
Table 6.1: Modes 215
Table 6.2: Idle mode procedures 215

Part K:7

DIAL-UP NETWORKING PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the Dial-up Networking use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Dial-up Networking use case.

CONTENTS

1 Introduction 223
1.1 Scope 223
1.2 Bluetooth Profile Structure 223
1.3 Symbols and conventions 224
1.3.1 Requirement status symbols 224
1.3.2 Signalling diagram conventions 225
1.3.3 Notation for timers and counters 225

2 Profile overview 226
2.1 Profile stack 226
2.2 Configurations and roles 227
2.3 User requirements and scenarios 228
2.4 Profile fundamentals 229
2.5 Conformance 229

3 Application layer 230
3.1 Service overview 230
3.2 Data calls 230
3.3 Fax service 230
3.4 Voice calls 230

4 Dialling and Control Interoperability Requirements 231
4.1 AT command set used 231
4.1.1 Command syntax 231
4.1.2 Commands 231
4.1.3 Result codes 233
4.2 Call progress audio feedback 233
4.3 Escape sequence 234

5 Serial Port Profile Interoperability Requirements 235
5.1 RFCOMM Interoperability Requirements 235
5.2 L2CAP Interoperability Requirements 235
5.3 SDP Interoperability Requirements 235
5.4 Link Manager (LM) Interoperability Requirements 236
5.5 Link Control (LC) Interoperability Requirements 237
5.5.1 Class of Device usage 237

6 Generic Access Profile Interoperability Requirements 238
6.1 Modes 238
6.2 Security aspects 238
6.3 Idle mode procedures 239
6.3.1 Bonding 239

7 References 240

8 List of Figures 241

9 List of Tables 242

1 INTRODUCTION

1.1 SCOPE

The Dial-up Networking Profile defines the protocols and procedures that shall be used by devices implementing the usage model called 'Internet Bridge' (see Bluetooth SIG MRD). The most common examples of such devices are modems and cellular phones.

The scenarios covered by this profile are the following:

- Usage of a cellular phone or modem by a computer as a wireless modem for connecting to a dial-up internet access server, or using other dial-up services
- Usage of a cellular phone or modem by a computer to receive data calls

1.2 BLUETOOTH PROFILE STRUCTURE

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

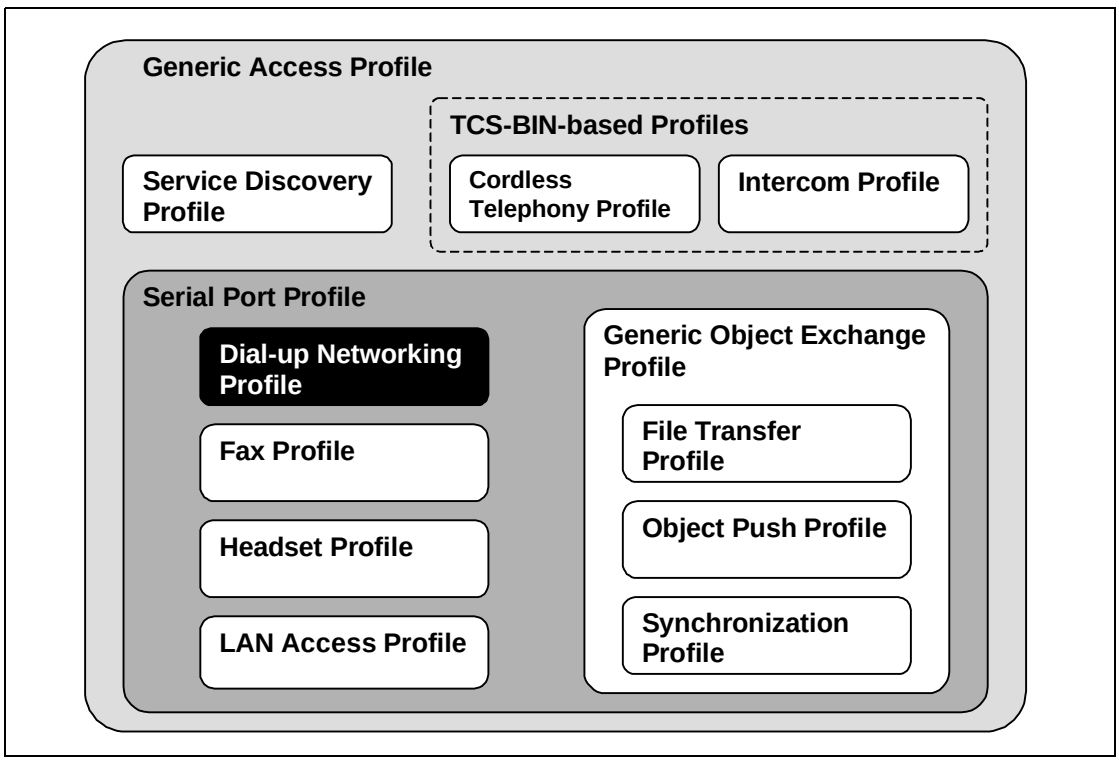


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

‘M’ for mandatory to support (used for capabilities that shall be used in the profile);

‘O’ for optional to support (used for capabilities that can be used in the profile);

‘C’ for conditional support (used for capabilities that will be used in case a certain other capability is supported);

‘X’ for excluded (used for capabilities that may be supported by the unit but which shall never be used in the profile);

‘N/A’ for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:

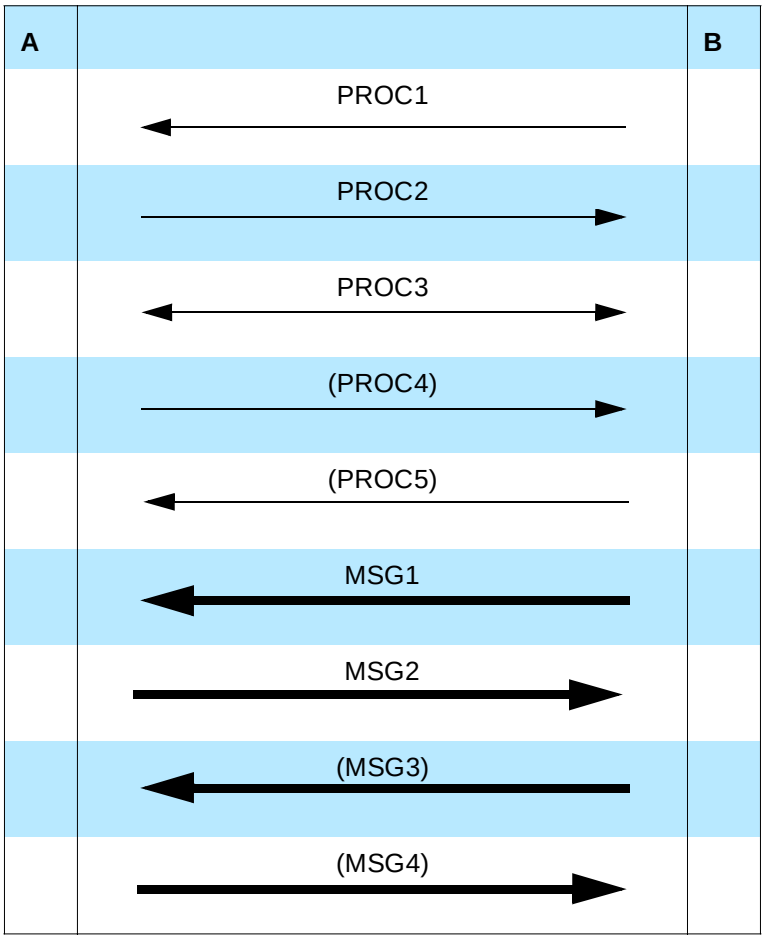


Table 1.1: Arrows used in signalling diagrams

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A and B). PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates an optional message from B to A.

1.3.3 Notation for timers and counters

Timers and counters may be introduced specific to this profile. To distinguish them from timers (counters) used in the Bluetooth protocol specifications and other profiles, these timers (counters) are named in the following format: 'T_{DNF}*nnn*' ('N_{DNF}*nnn*').

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

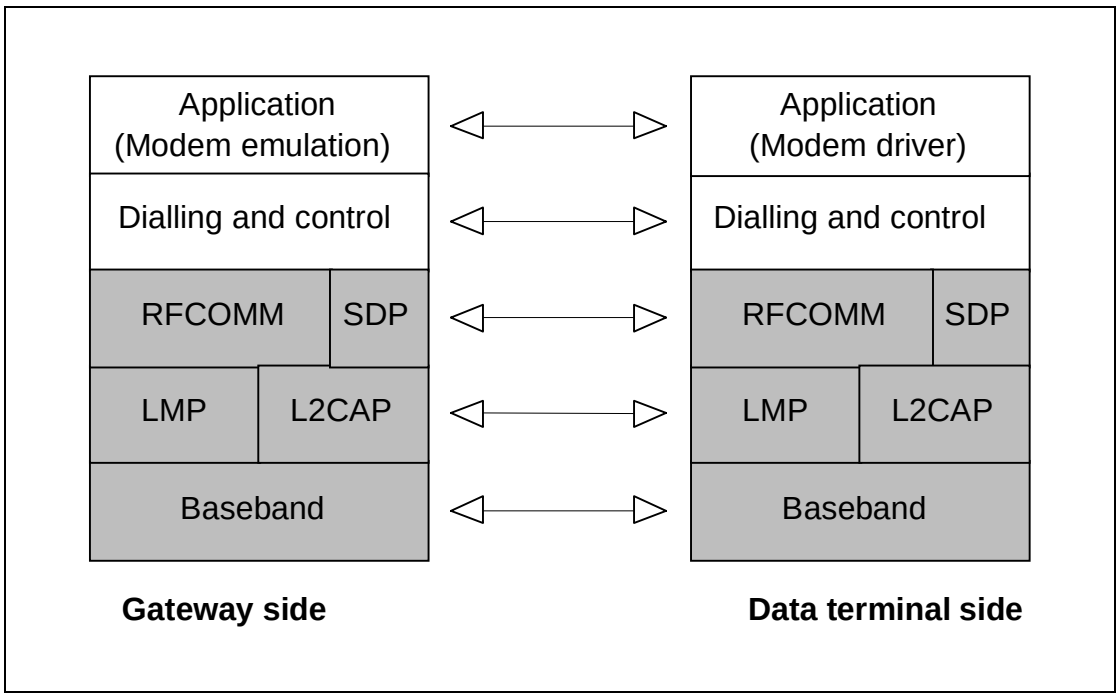


Figure 2.1: Protocol model

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth adaptation of GSM TS 07.10 [5], used for providing serial port emulation. SDP is the Bluetooth Service Discovery Protocol. Dialling and control (see Section 4) is the commands and procedures used for automatic dialling and control over the asynchronous serial link provided by the lower layers.

The modem emulation layer shown in Figure 2.1 is the entity emulating the modem, and the modem driver is the driver software in the data terminal.

For the shaded protocols/entities in Figure 2.1, The Serial Port Profile is used as base standard. For these protocols, all requirements stated in Serial Port Profile apply, except in those cases where this profile explicitly states deviations.

Note: Although not shown in the model above, it is assumed by this profile that the application layer has access to some lower layer procedures (for example SCO link establishment).

2.2 CONFIGURATIONS AND ROLES

The figures below show two typical configurations of devices for this profile:

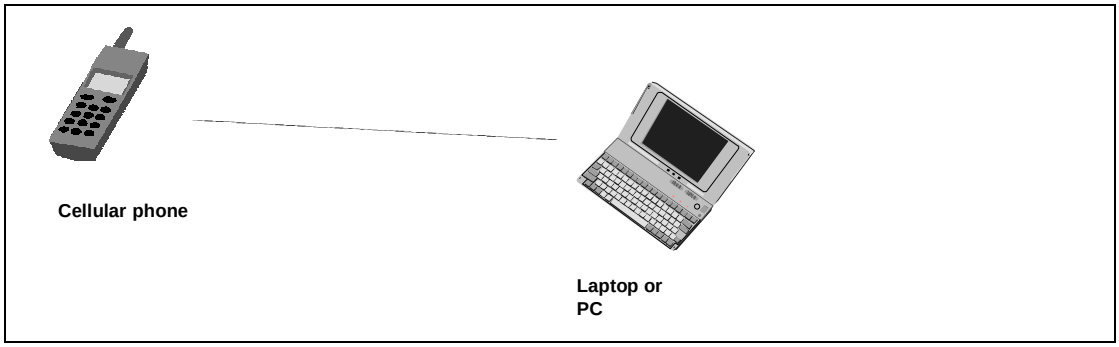


Figure 2.2: Dial-up Networking profile, example with cellular phone

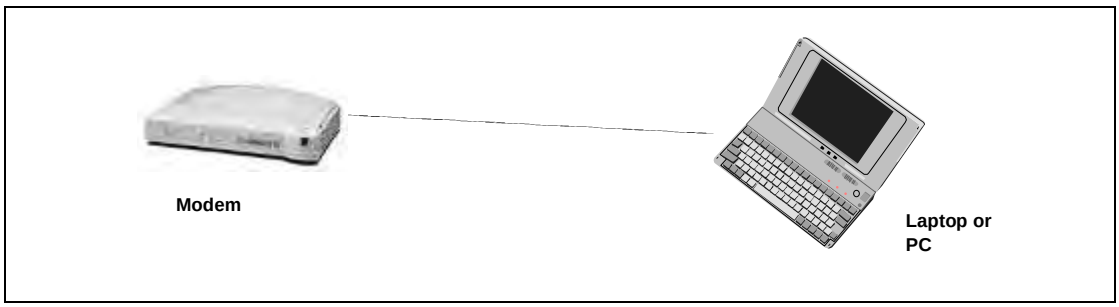


Figure 2.3: Dial-up Networking profile, example with modem

The following roles are defined for this profile:

Gateway (GW) – This is the device that provides access to the public network. Typical devices acting as gateways are cellular phones and modems.

Data Terminal (DT) – This is the device that uses the dial-up services of the gateway. Typical devices acting as data terminals are laptops and desktop PCs.

In the rest of this document, these terms are only used to designate these roles.

For purposes of mapping the Dial-up Networking profile to the conventional modem system architecture, the GW is considered Data Circuit Endpoint (DCE), and the DT is considered Data Terminal Endpoint (DTE).

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Usage of a GW by a DT as a wireless modem for connecting to a dial-up internet access server or using other dial-up services
- Usage of a GW by a DT to receive data calls

The following restrictions apply to this profile:

a) The modem is not required to be able to report and/or discriminate between different call types for incoming calls.

b) This profile requires support for one-slot packets only. This means that this profile ensures that data rates up to 128 kbps can be used. Support for higher rates are optional.

c) Only one call at a time is supported.

d) The profile only supports point-to-point configurations.

e) There is no way defined in this profile to discriminate between two SCO channels originating from the same device. It is therefore manufacturer-specific as to how to deal with the situation where there are multiple applications requiring the use of multiple SCO channels originating from the same device.

f) Before a cellphone or modem can be used with a PC/Laptop for the first time, an initialization procedure must be performed. This typically involves manually activating initialization support, and entering a PIN code on the PC/Laptop keyboard (see Generic Access Profile for more details). This procedure may have to be repeated under certain circumstances.

g) This profile does not support multiple instances of its implementation in the same device.

Security is ensured by authenticating the other party upon connection establishment, and by encrypting all user data. The baseband and LMP mechanisms for authentication and encryption are used.

2.4 PROFILE FUNDAMENTALS

Before a DT can use the services of a GW for the first time, the two devices have to initialize. Initialization includes exchanging a PIN code, creation of link keys and service discovery.

A link has to be established before calls can be initiated or received. This requires paging of the other device. Link establishment is always initiated by the DT.

There are no fixed master/slave roles.

The GW and DT provide serial port emulation. For the serial port emulation, the serial port profile (see Serial Port Profile) is used. The serial port emulation is used to transport the user data, modem control signals and AT commands between the GW and the DT. AT-commands are parsed by the GW and responses are sent to the DT.

An SCO link is used to transport audio.

For security purposes, authentication is used, and all user data is encrypted. For this, the baseband/LMP mechanisms are used.

2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 APPLICATION LAYER

This section describes the service requirements on units active in the Dial-up Networking profile.

3.1 SERVICE OVERVIEW

Table 3.1 shows the required services:

	Services	Support in DT	Support in GW
1.	Data call without audio feedback	M	M
2.	Data call with audio feedback	O	O
3.	Fax services without audio feedback	N/A	N/A
4.	Fax services with audio feedback	N/A	N/A
5.	Voice call	N/A	N/A

Table 3.1: Application layer procedures

3.2 DATA CALLS

The support of data calls is mandatory for both GWs and DTs. Optionally, audio feedback may be provided (see Section 4.2).

The GW shall emulate a modem connected via a serial port. The Serial Port Profile is used for RS-232 emulation, and a modem emulation entity running on top of the serial port profile provides the modem emulation.

3.3 FAX SERVICE

The support of fax is not covered by this profile. Refer to Fax Profile.

3.4 VOICE CALLS

The support of voice calls is not covered by this profile.

4 DIALLING AND CONTROL INTEROPERABILITY REQUIREMENTS

4.1 AT COMMAND SET USED

To guarantee that basic functionality can always be provided, it is required that a GW device supports the commands and responses as defined in the following sub-clauses.

The commands are based on ITU-T V.250 and GSM 07.07.

4.1.1 Command syntax

For the exchange of the commands, responses and unsolicited results codes, the format, syntax and procedures of ITU-TV.250 [6] apply.

4.1.2 Commands

The table below lists all commands that shall be supported by the GW.

Name	Description	Reference
&C	Circuit 109 (Received line signal detector) Behavior	Shall be supported as defined in [6].
&D	Circuit 108 (Data terminal ready) Behavior	Shall be supported as defined in [6].
&F	Set to Factory-defined Configuration	Shall be supported as defined in [6].
+GCAP	Request Complete Capabilities List	Shall be supported as defined in [6].
+GMI	Request Manufacturer Identification	Shall be supported as defined in [6].
+GMM	Request Model Identification	Shall be supported as defined in [6].
+GMR	Request Revision Identification	Shall be supported as defined in [6].
A	Answer	Shall be supported as defined in [6].
D	Dial	Shall be supported *either* as defined in [6] *or* as defined in [10].
E	Command Echo	Shall be supported as defined in [6].
H	Hook Control	Shall be supported as defined in [6].
L	Monitor Speaker Loudness	Shall be supported as defined in [6].
M	Monitor Speaker Mode	Shall be supported as defined in [6].

Table 4.1: Required commands

Name	Description	Reference
O	Return to Online Data State	Shall be supported as defined in [6].
P	Select Pulse Dialling	Shall be supported as defined in [6].
Q	Result Code Suppression	Shall be supported as defined in [6].
S0	Automatic Answer	Shall be supported as defined in [6].
S10	Automatic Disconnect Delay	Shall be supported as defined in [6].
S3	Command Line Termination Character	Shall be supported as defined in [6].
S4	Response Formatting Character	Shall be supported as defined in [6].
S5	Command Line Editing Character	Shall be supported as defined in [6].
S6	Pause Before Blind Dialling	Shall be supported as defined in [6].
S7	Connection Completion Timeout	The setting of this parameter may be ignored. If not ignored, it shall be supported as defined in [6].
S8	Comma Dial Modifier Time	Shall be supported as defined in [6].
T	Select Tone Dialling	Shall be supported as defined in [6].
V	DCE Response Format	Shall be supported as defined in [6].
X	Result Code Selection and Call Progress Monitoring Control	Shall be supported as defined in [6].
Z	Reset To Default Configuration	Shall be supported as defined in [6].

Table 4.1: Required commands

4.1.3 Result codes

The table below lists all result codes that shall be supported by the GW.

Name	Description	Reference
OK	Acknowledges execution of a command.	Shall be supported as defined in [6].
CONNECT	Connection has been established.	Shall be supported as defined in [6].
RING	The DCE has detected an incoming call signal from the network.	Shall be supported as defined in [6].
NO CARRIER	The connection has been terminated, or the attempt to establish a connection failed.	Shall be supported as defined in [6].
ERROR	Error.	Shall be supported as defined in [6].
NO DIALTONE	No dial-tone detected.	Shall be supported as defined in [6].
BUSY	Busy signal detected.	Shall be supported as defined in[6].

Table 4.2: Required result codes

4.2 CALL PROGRESS AUDIO FEEDBACK

The GW or DT may optionally be able to provide audio feedback during call establishment. This clause applies only to gateways/data terminals that are able to provide audio feedback.

SCO links are used to transport the digitized audio over the Bluetooth link. The GW shall take all initiatives for SCO link establishment. The setting of the M parameter (see [6], Section 6.3.14) controls whether audio feedback is provided by the GW.

If a GW provides audio feedback for a call, the GW shall use the initiate SCO link procedure (see Link Manager protocol) to establish the audio link when the DCE goes off-hook.

Depending on the setting of the M parameter, the GW releases the audio link when the DCE has detected a carrier or when the DCE goes on-hook. The remove SCO link procedure (see [Link Manager protocol]) shall be used for audio link release.

If SCO link establishment fails, the call establishment shall proceed without the audio feedback.

This profile assumes that the DT is not active in any other profile which uses SCO links while it is operating in the Dial-up Networking profile. Therefore, the behavior in a situation where multiple SCO links are established simultaneously is undefined.

4.3 ESCAPE SEQUENCE

It is recommended that the GW supports an escape sequence (i.e. a sequence of characters which causes the GW to leave the online data state and go to the online command state). This profile does not mandate a particular escape sequence – it is up to the implementer of the profile if and how returning to command mode is supported.

5 SERIAL PORT PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the Serial Port Profile. For the purposes of reading the Serial Port Profile, the GW shall always be considered to be Device B and the DT shall always be considered to be Device A.

The following text together with the associated sub-clauses define the requirements with regards to this profile, in addition to the requirements defined in Serial Port Profile.

5.1 RFCOMM INTEROPERABILITY REQUIREMENTS

For RFCOMM, no additions to the requirements stated in Serial Port Profile apply.

5.2 L2CAP INTEROPERABILITY REQUIREMENTS

For the L2CAP layer, no additions to the requirements stated in Serial Port Profile apply.

5.3 SDP INTEROPERABILITY REQUIREMENTS

Table 5.1 lists all entries in the SDP database of the GW defined by this profile. The ‘Status’ column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonics used in the ‘Value’ column, and the codes assigned to the attribute identifiers, can be found in the Bluetooth Assigned Numbers section.

Item	Definition:	Type:	Value:	Status	Default
Service Class ID List				M	
Service Class #0		UUID	Dial-up Networking	M	
Service Class #1		UUID	Generic Networking	O	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	RFCOMM	M	

Table 5.1: Service Database Entries

Item	Definition:	Type:	Value:	Status	Default
Parameter for Protocol #1	Server Channel	UInt8	1,2,3,...,30	M	
Service Name	Displayable Text name	String	Service-provider defined	O	'Dial-up networking'
Audio Feedback Support		Boolean	True/False	O	False
BluetoothProfileDescriptorList				M	
Profile #0		UUID	Dial-up Networking	M	
Parameter for Profile #0	Version	UInt16	0x0100*	O	0x100

Table 5.1: Service Database Entries

*. Indicating version 1.0

5.4 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

In addition to the requirements for the Link Manager as stated in the "Serial Port Profile" on page 171, this profile requires support for SCO links, in both the GW and DT. The support is conditional upon the ability to provide audio feedback."

5.5 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

In the table below, all LC capabilities required by this profile are listed.

	Capabilities	Support in GW	Support in DT
5.	Packet types		
N	HV3 packet	C1	C2
7.	Voice codec		
C	CVSD	C1	C2
C1: The support for this capability is mandatory for gateways that are able to provide audio feedback to the DT. C2: The support for this capability is mandatory for data terminals that are able to provide audio feedback to the user.			

Table 5.2: Baseband/LC capabilities

5.5.1 Class of Device usage

A device which is active in the GW role of the Dial-up Networking profile shall, in the Class of Device field:

1. Set the bits ‘Telephony’ and ‘Networking’ in the Service Class field (see Bluetooth Assigned Numbers)
2. Indicate ‘Phone’ as Major Device class (see Bluetooth Assigned Numbers)

This may be used by an inquiring device to filter the inquiry responses.

6 GENERIC ACCESS PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the Generic Access Profile.

This section defines the support requirements with regards to procedures and capabilities defined in Generic Access Profile.

6.1 MODES

The table shows the support status for Modes within this profile.

	Procedure	Support in DT	Support in GW
1	Discoverability modes		
	Non-discoverable mode	N/A	O
	Limited discoverable mode	N/A	O
	General discoverable mode	N/A	O
2	Connectability modes		
	Non-connectable mode	N/A	X
	Connectable mode	N/A	M
3	Pairing modes		
	Non-pairable mode	M	O
	Pairable mode	O	M

Table 6.1: Modes

6.2 SECURITY ASPECTS

The table shows the support status for Security aspects within this profile

	Procedure	Support in DT	Support in GW
1	Authentication	M	M
2	Security modes		
	Security mode 1	N/A	X
	Security mode 2	C1	C1
	Security mode 3	C1	C1
C1: Support for at least one of the security modes 2 and 3 is mandatory.			

Table 6.2: Security aspects

6.3 IDLE MODE PROCEDURES

The table shows the support status for Idle mode procedures within this profile

	Procedure	Support in DT	Support in GW
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	O	N/A
4	Device discovery	O	N/A
5	Bonding	M (Note 1)	M (Note 1)
Note 1: See section 6.3.1			

Table 6.3: Idle mode procedures

6.3.1 Bonding

It is mandatory for the DT to support initiation of bonding, and for the GW to accept bonding.

7 REFERENCES

[1] Bluetooth Baseband specification

[2] Bluetooth Link Manager Protocol

[3] Bluetooth Logical Link Control and Adaptation Protocol Specification

[4] RFCOMM with TS 07.10

[5] TS 101 369 (GSM 07.10) version 6.1.0

[6] International Telecommunication Union, “ITU-T Recommendation V.250”

[7] Bluetooth Service Discovery Protocol

[8] John Webb, “Bluetooth SIG MRD”, version 1.0 Draft

[9] Bluetooth Serial Port Profile

[10] ETS 300 916 (GSM 07.07) version 5.6.0

[11] Bluetooth Fax Profile

[12] Bluetooth Assigned Numbers
http://www.bluetooth.org/assigned-numbers.htm

8 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 223
Figure 2.1: Protocol model 226
Figure 2.2: Dial-up Networking profile, example with cellular phone 227
Figure 2.3: Dial-up Networking profile, example with modem 227

9 LIST OF TABLES

Table 1.1: Arrows used in signalling diagrams 225
Table 3.1: Application layer procedures 230
Table 4.1: Required commands 231
Table 4.2: Required result codes 233
Table 5.1: Service Database Entries 235
Table 5.2: Baseband/LC capabilities 237
Table 6.1: Modes 238
Table 6.2: Security aspects 238
Table 6.3: Idle mode procedures 239

Part K:8

FAX PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the Fax use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Fax use case.

CONTENTS

1 Introduction 246
1.1 Scope 246
1.2 Profile Dependencies 246
1.3 Symbols and conventions 247
1.3.1 Requirement status symbols 247
1.3.2 Signalling diagram conventions 248
2 Profile overview 249
2.1 Profile stack 249
2.2 Configurations and roles 250
2.3 User requirements and scenarios 251
2.4 Profile fundamentals 251
2.5 Conformance 252
3 Application layer 253
3.1 Service overview 253
3.2 Data calls 253
3.3 Fax service 253
3.4 Voice calls 253
4 Dialling and Control Interoperability Requirements 254
4.1 AT command set used 254
4.1.1 Command syntax, Protocols and Result Codes 254
4.1.2 Fax Service Class selection procedure 254
4.2 Call progress audio feedback 255
5 Serial Port Profile 256
5.1 RFCOMM Interoperability Requirements 256
5.2 L2CAP Interoperability Requirements 256
5.3 SDP Interoperability Requirements 256
5.4 Link Manager (LM) Interoperability Requirements 257
5.5 Link Control (LC) Interoperability Requirements 258
5.5.1 Class of Device usage 258
6 Generic Access Profile Interoperability Requirements 259
6.1 Modes 259
6.2 Security aspects 260
6.3 Idle mode procedures 260
6.3.1 Bonding 260
7 References 261
8 List of Figures 262
9 List of Tables 263

1 INTRODUCTION

1.1 SCOPE

The Fax profile defines the protocols and procedures that shall be used by devices implementing the fax part of the usage model called‘Data Access Points, Wide Area Networks’ (see Bluetooth SIG MRD).

A Bluetooth cellular phone or modem may be by a computer as a wireless fax modem to send or receive a fax message.

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

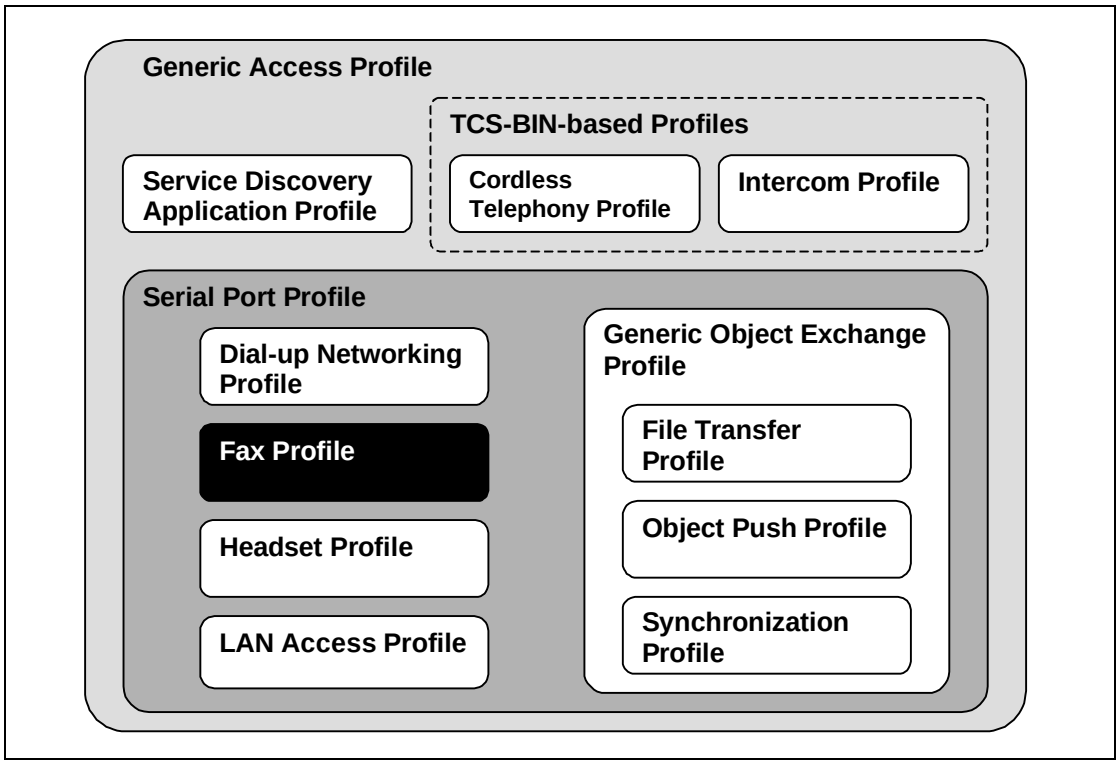


Figure 1.1: Bluetooth Profiles

As indicated in the figure, the Fax profile is dependent upon both the Serial Port Profile and the Generic access profile – details are provided in Section 5 Serial Port Profile on page 265 and Section 6 Generic Access Profile Interoperability Requirements on page 268.

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

- ‘M’ for mandatory to support
- ‘O’ for optional to support
- ‘X’ for excluded (used for capabilities that may be supported by the unit but which shall never be used in the use case)
- ‘C’ for conditional to support
- ‘N/A’ for not applicable (in the given context it is impossible to use this capability)

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices in this use case. Therefore, these features shall never be activated while a unit is operating as a unit within this use case.

Within the scope of this Fax profile, the expression ‘Fax class” is used as a shortcut to ‘facsimile service class” as defined by [2], [3], [4] and [4]. This is not to be confused with Bluetooth service class.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:

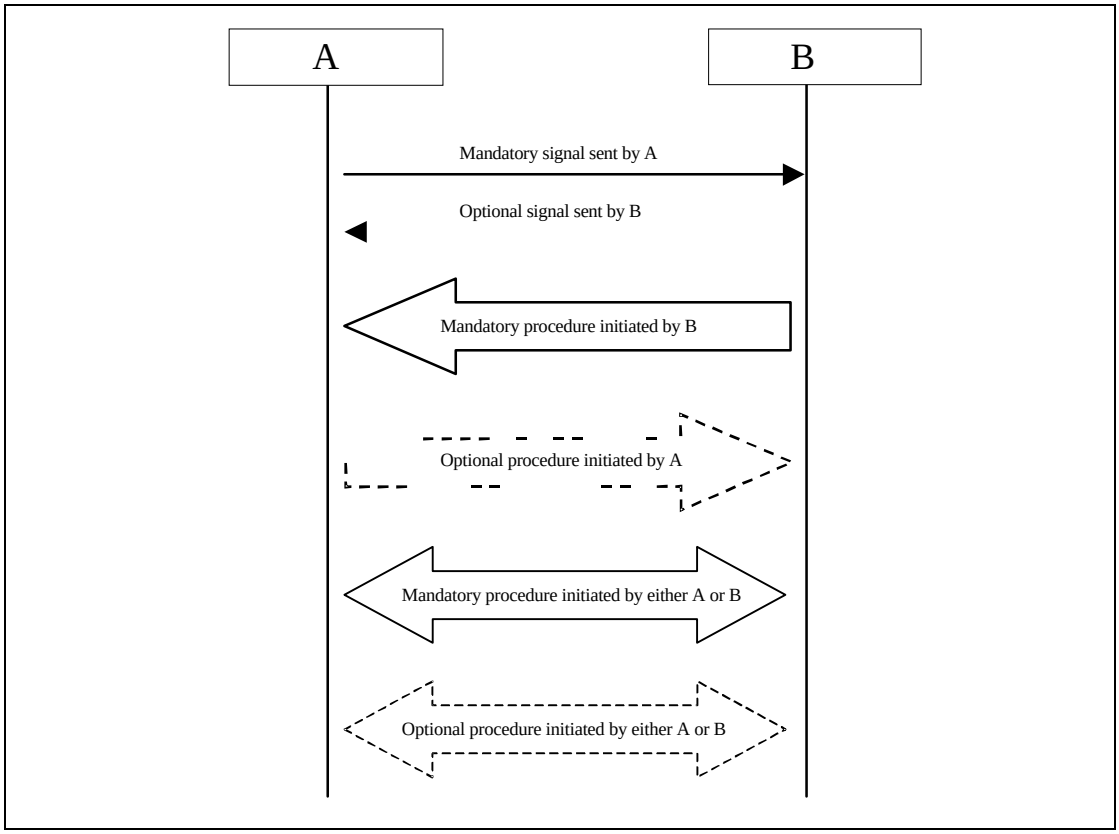


Figure 1-2 Arrows used in signalling diagrams

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

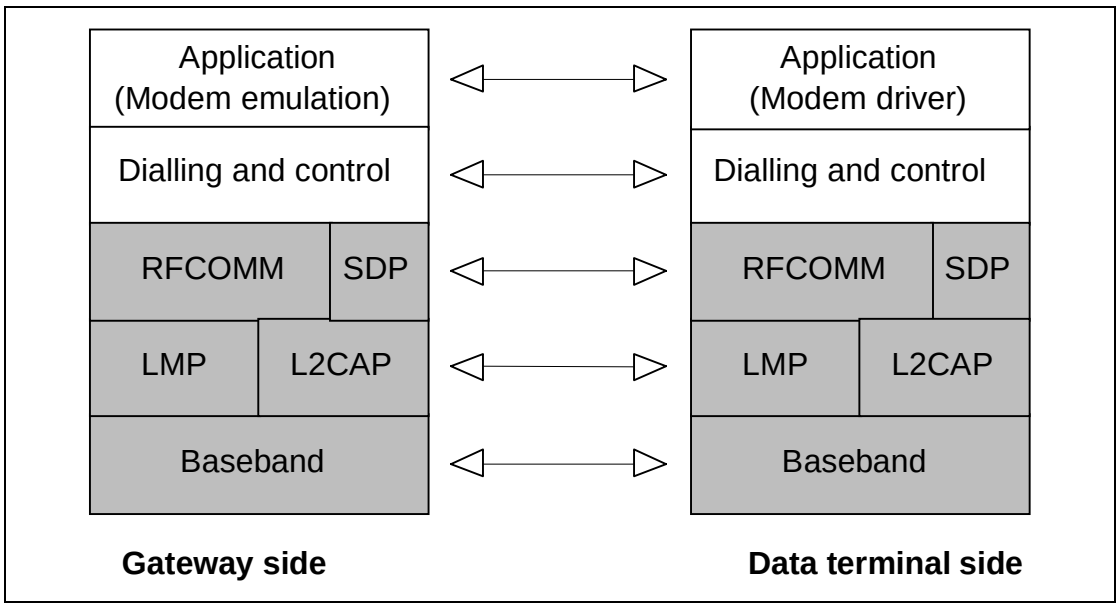


Figure 2.1: Protocol model

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth adaptation of GSM TS 07.10 [1], used for providing serial port emulation. SDP is the Bluetooth Service Discovery Protocol. Dialling and control (see Section 4) defines the commands and procedures used for automatic dialling and control over the asynchronous serial link provided by the lower layers.

The modem emulation layer shown in Figure 2.1 is the entity emulating the modem, and the modem driver is the driver software in the data terminal.

For the shaded protocols/entities in Figure 2.1, The Serial Port Profile is used as base standard. For these protocols, all requirements stated in Serial Port Profile apply, except in those cases where this profile explicitly states deviations.

Note: Although not shown in the model above, it is assumed by this profile that the application layer has access to some lower layer procedures (for example SCO link establishment).

2.2 CONFIGURATIONS AND ROLES

The figures below show two typical configurations of devices for this profile:

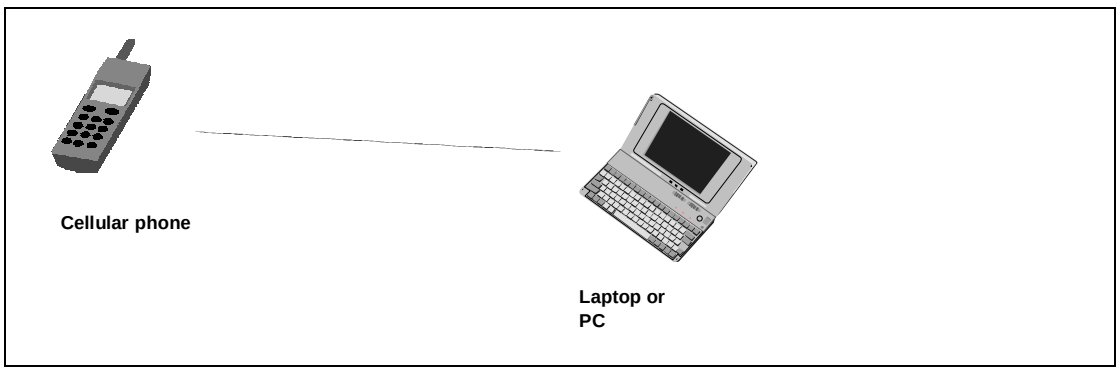


Figure 2.2: Fax profile, example with cellular phone

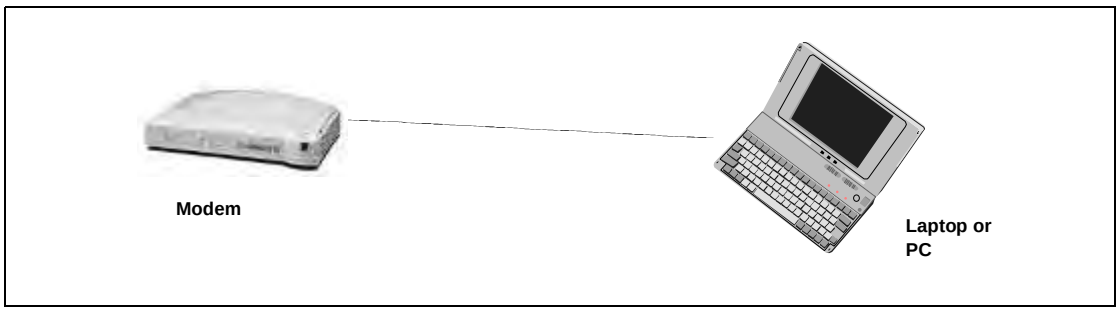


Figure 2.3: Fax profile, example with modem

The following roles are defined for this profile:

Gateway (GW) – This is the device that provides facsimile services. Typical devices acting as gateway are cellular phones and modems.

Data Terminal (DT) – This is the device that uses the facsimile services of the gateway. Typical devices acting as data terminals are laptops and desktop PCs.

In the rest of this document, these terms are only used to designate these roles.

For purposes of mapping the Fax profile to the conventional modem system architecture, the GW is considered Data Circuit Endpoint (DCE), and the DT is considered Data Terminal Endpoint (DTE).

2.3 USER REQUIREMENTS AND SCENARIOS

The Fax profile defines the usage of a GW by a DT as a wireless modem to send or receive fax messages

The following restrictions apply to this profile:

a) The GW (cellphone or modem) is not required to be able to report and/or discriminate between different call types for incoming calls.

b) This profile requires support for one-slot packets only. This means that this profile ensures that data rates up to 128 kbps can be used. Support for higher rates are optional.

c) Only one call at a time is supported.

d) The profile only supports point-to-point configurations.

e) Since in this profile there is no way defined to discriminate between 2 SCO channels originating from the same device, it is manufacturer specific as to deal with the situation where there are multiple applications requiring the use of multiple SCO channels originating from the same device.

f) This profile does not support multiple instances of its implementation in the same device.

2.4 PROFILE FUNDAMENTALS

Here is a brief summary of the interactions that take place when a DT wants to use the facsimile services of a GW.

1. If the DT does not have the Bluetooth Address of the GW, the DT has to obtain the address; e.g. using the Device discovery procedure, see Section 6.4 of Generic Access profile.
2. The Fax profile mandates the use of a secure connection through the authentication procedure (see Section 5.1 of Generic Access profile), and encryption of all user data through the baseband / LMP encryption mechanisms (see Section 8 of the Generic Access profile).
3. Link establishment is always initiated by the DT.
4. There are no fixed master / slave roles.
5. The fax call is established.
6. The GW and DT provide serial port emulation. For the serial port emulation, the serial port profile (see Serial Port Profile) is used. The serial port emulation is used to transport the user data, modem control signals and AT commands between the GW and the DT. AT-commands are parsed by the GW and responses are sent to the DT.
7. An optional SCO link may be used to transport fax audio feedback.
8. After the fax call has been cleared, the channel and link will be released as well.

2.5 CONFORMANCE

When conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities, for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 APPLICATION LAYER

This section describes the service requirements on units active in the Fax profile.

3.1 SERVICE OVERVIEW

Table 3.1 shows the required services:

	Services	Support in DT	Support in GW
1.	Data call without audio feedback	N/A	N/A
2.	Data call with audio feedback	N/A	N/A
3.	Fax services without audio feedback	M	M
4.	Fax services with audio feedback	O	O
5.	Voice call	N/A	N/A

Table 3.1: Application layer procedures

3.2 DATA CALLS

The support of data calls is not covered by this profile.
Refer to Dial-up Networking Profile.

3.3 FAX SERVICE

At least one of the following fax classes of service is mandatory for both the GW and the paired DT (see Section 4.1.2):

Fax Class 1 TIA-578-A[2] and ITU T.31 [4]

Fax Class 2.0 TIA-592 [3]and ITU T.32 [5]

Fax Service Class 2 – No industry standard exists (manufacturer specific).

Optionally, audio feedback may be provided (see Section 4.2).

The GW shall emulate a modem connected via a serial port. The Serial Port Profile is used for RS-232 emulation, and RFCOMM running on top of the serial port profile provides the modem emulation.

3.4 VOICE CALLS

The support of voice calls is not covered by this profile. Refer to Cordless Telephony Profile.

4 DIALLING AND CONTROL INTEROPERABILITY REQUIREMENTS

4.1 AT COMMAND SET USED

To guarantee that basic functionality can always be provided, it is required that a GW device supports the commands and responses as defined in the supported fax class of service(s):

Fax Class 1 TIA-578-A [2] and ITU T.31 [4]

Fax Class 2.0 TIA-592 [3]] and ITU T.32 [5]

Fax Service Class 2 – No industry standard exists (manufacturer specific).

4.1.1 Command syntax, Protocols and Result Codes

Refer to each specific implemented fax service class document for a description of the required commands, protocols and result codes.

4.1.2 Fax Service Class selection procedure

This profile does not require a specific service class of fax. This profile supports 2 standards-based fax ‘classes’ – fax class 1 [2], [4] and fax class 2.0 [3], [5] – and a third manufacturer-specific pseudo-standard, fax class 2 (no industry reference standard exists).

The DT shall check the GW SDP or perform an ‘AT+FCLASS” command to discover the fax class of service(s) supported by the GW.

Bluetooth devices implementing this profile must support a minimum of one fax service class, but may support any or all fax services classes.

4.2 CALL PROGRESS AUDIO FEEDBACK

The GW or DT may optionally be able to provide audio feedback during call establishment. This clause applies only to gateways/data terminals that are able to provide audio feedback.

SCO links are used to transport the digitized audio over the Bluetooth link. The GW shall take all initiatives for SCO link establishment. The setting of the M parameter (see [6], Section 6.3.14) controls whether the GW provides audio feedback.

If a GW provides audio feedback for a call, the GW shall use the ‘initiate SCO link’ procedure (see Link Manager protocol) to establish the audio link when the DCE goes off-hook.

Depending on the setting of the M parameter, the GW releases the audio link when the DCE has detected a carrier or when the DCE goes on-hook. The ‘remove SCO link’ procedure (see [Link Manager protocol]) shall be used for audio link release.

If SCO link establishment fails, the call establishment shall proceed without the audio feedback.

This profile assumes that the DT is not active in any other profile that uses SCO links while it is operating in the Fax profile. Therefore, behavior is not defined for a situation where multiple SCO links are established simultaneously.

5 SERIAL PORT PROFILE

This profile requires compliance to the Serial Port Profile. For the purposes of reading the Serial Port Profile, the GW shall always be considered to be Device B and the DT shall always be considered to be Device A.

The following text together with the associated sub-clauses define the requirements with regards to this profile in addition to the requirements defined in the Serial Port Profile.

5.1 RFCOMM INTEROPERABILITY REQUIREMENTS

For RFCOMM, no additions to the requirements stated in Serial Port Profile apply.

5.2 L2CAP INTEROPERABILITY REQUIREMENTS

For the L2CAP layer, no additions to the requirements stated in Serial Port Profile apply.

5.3 SDP INTEROPERABILITY REQUIREMENTS

Table 5.1 lists all entries in the SDP database of the GW defined by this profile. The ‘Status’ column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonics used in the ‘Value’ column and the codes assigned to the attribute identifiers can be found in Bluetooth Assigned Numbers.

Item	Definition:	Type:	Value:	Status	Default
Service Class ID List				M	
Service Class #0		UUID	Fax	M	
Service Class #1		UUID	Generic Telephony	O	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	RFCOMM	M	
Parameter for Protocol #1	Server Channel	UInt8	N = server channel #	M	
Service Name	Displayable Text name	String	Service-provider defined	O	'Fax'
Audio Feedback Support		Boolean	True/False	O	False
Fax Class 1 Support		Boolean	True/False	O	False
Fax Class 2.0 Support		Boolean	True/False	O	False
Fax Class 2 Support		Boolean	True/False	O	False
BluetoothProfile Descrip-torList				M	
Profile #0		UUID	Fax	M	
Parameter for Profile #0	Version	UInt16	0x0100*	O	0x100

Table 5.1: Service Database Entries

*. Indicating version 1.0

5.4 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

In addition to the requirements for the Link Manager as stated in the "Serial Port Profile" on page 171, this profile requires support for SCO links, in both the GW and DT. The support is conditional upon the ability to provide audio feedback."

5.5 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

In the table below, all LC capabilities required by this profile are listed.

	Capabilities	Support in GW	Support in DT
5.	Packet types		
N	HV3 packet	C1	C2
7.	Voice codec		
C	CVSD	C1	C2
C1: The support for this capability is mandatory for gateways that are able to provide audio feedback to the DT. C2: The support for this capability is mandatory for data terminals that are able to provide audio feedback to the user.			

Table 5.2: Baseband/LC capabilities

5.5.1 Class of Device usage

A device which is active in the GW role of the Fax profile shall, in the Class of Device field:

1. Set the ‘Telephony’ bit in the Service Class field (see Bluetooth Assigned Numbers)
2. Indicate ‘Phone’ as Major Device class (see Bluetooth Assigned Numbers)

This may be used by an inquiring device to filter the inquiry responses.

6 GENERIC ACCESS PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the Generic Access Profile.

This section defines the support requirements with regards to procedures and capabilities defined in Generic Access Profile.

6.1 MODES

The table shows the support status for Modes within this profile.

	Procedure	Support in DT	Support in GW
1	Discoverability modes		
	Non-discoverable mode	N/A	O
	Limited discoverable mode	N/A	O
	General discoverable mode	N/A	O
2	Connectability modes		
	Non-connectable mode	N/A	X
	Connectable mode	N/A	M
3	Pairing modes		
	Non-pairable mode	M	O
	Pairable mode	O	M
C1: If limited discoverable mode is supported, non-discoverable mode is mandatory, otherwise optional.			
C2: A Bluetooth device shall support at least one discoverable mode (limited or/and general).			
C3: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise optional.			

Table 6.1: Modes

6.2 SECURITY ASPECTS

The table shows the support status for Security aspects within this profile.

	Procedure	Support in DT	Support in GW
1	Authentication	M	M
2	Security modes		
	Security mode 1	N/A	X
	Security mode 2	C1	C1
	Security mode 3	C1	C1
C1: Support for at least one of the security modes 2 and 3 is mandatory			

Table 6.2: Security aspects

6.3 IDLE MODE PROCEDURES

The table shows the support status for Idle mode procedures within this profile.

	Procedure	Support in DT	Support in GW
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	O	N/A
4	Device discovery	O	N/A
5	Bonding	M (Note 1)	M (Note 1)
Note 1: See section 6.3.1			

Table 6.3: Idle mode procedures

6.3.1 Bonding

It is mandatory for the DT to support initiation of bonding, and for the GW to accept bonding.

7 REFERENCES

[1] TS 101 369 (GSM 07.10) version 6.1.0

[2] TIA-578-A Facsimile Digital Interface. Asynchronous Facsimile DCE Control Standard, Service Class 1

[3] TIA-592 Facsimile Digital Interface. Asynchronous Facsimile DCE Control Standard, Service Class 2

[4] ITU T.31 Asynchronous Facsimile DCE Control – Service Class 1

[5] ITU T.32 Asynchronous Facsimile DCE Control – Service Class 2

[6] International Telecommunication Union, “ITU-T Recommendation V.250”

8 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 246
Figure 2.1: Protocol model 249
Figure 2.2: Fax profile, example with cellular phone 250
Figure 2.3: Fax profile, example with modem 250

9 LIST OF TABLES

Table 3.1: Application layer procedures 253
Table 5.1: Service Database Entries 257
Table 5.2: Baseband/LC capabilities 258
Table 6.1: Modes 259
Table 6.2: Security aspects 260
Table 6.3: Idle mode procedures 260

Part K:9

LAN ACCESS PROFILE

This document is a LAN Access Profile for Bluetooth devices. Firstly, this profile defines how Bluetooth-enabled devices can access the services of a LAN using PPP. Secondly, this profile shows how the same PPP mechanisms are used to form a network consisting of two Bluetooth-enabled devices.

CONTENTS

1 Introduction 269
1.1 Scope 269
1.2 Profile Dependencies 270
1.3 Symbols and conventions 270
2 Profile overview 271
2.1 Protocol stack 271
2.2 Configurations and roles 271
2.3 User requirements and scenarios 272
2.4 Profile fundamentals 274
2.5 Conformance 274
3 User interface aspects 275
3.1 Security 275
3.2 Generic Modes 276
3.3 Additional Parameters 277
4 Application layer 278
4.1 Initialization of LAN Access Point service 278
4.2 Shutdown of LAN Access Point service 279
4.3 Establish LAN Connection 279
4.4 Lost LAN Connection 280
4.5 Disconnect LAN Connection 280
5 PPP 281
5.1 Initialize PPP 281
5.2 Shutdown PPP 282
5.3 Establish PPP Connection 282
5.4 Disconnect PPP Connection 282
5.5 PPP Authentication Protocols 283
6 RFCOMM 284
7 Service Discovery 285
7.1 SDP service records 285
8 L2CAP 287
9 Link Manager 288
9.1 Profile Errors 289
10 Link control 290

11 Management Entity Procedures 291
11.1 Link Establishment 291
11.1.1 No responses to inquiry 291
11.1.2 No response to paging 292
11.1.3 Pairing 292
11.1.4 Errors 292
11.2 Maximum Number of users 292
12 APPENDIX A (Normative): Timers and counters 293
13 APPENDIX B (Normative): Microsoft Windows 294
13.1 PC-2-PC configuration 294
14 APPENDIX C (Informative): Internet Protocol (IP) 295
14.1 IP Interfaces 295
14.1.1 Interface Enabled 295
14.1.2 Interface Disabled 295
14.2 The IPCP Protocol 295
14.2.1 IPCP Connection 296
14.2.2 IP Address Allocation 296
14.2.3 DNS and NBNS addresses 296
14.2.4 NetBIOS over IP 296
15 List of Figures 297
16 List of Tables 298
17 References 299
17.1 Normative references 299
17.2 Informative references 299

1 INTRODUCTION

1.1 SCOPE

This profile defines LAN Access using PPP over RFCOMM. There may be other means of LAN Access in the future.

- PPP is a widely deployed means of allowing access to networks. PPP provides authentication, encryption, data compression and multi-protocol facilities. PPP over RFCOMM has been chosen as a means of providing LAN Access for Bluetooth devices because of the large installed base of devices equipped with PPP software.
- It is recognized that PPP is capable of supporting various networking protocols (e.g. IP, IPX, etc.). This profile does not mandate the use of any particular protocol. However, since IP is recognized as the most important protocol used in today's networks, additional IP-related information is provided in an appendix. The use of these other PPP protocols is not discussed.
- This profile does not deal with conferencing, LAN emulation, ad-hoc networking or any other means of providing LAN Access. These functions are, or may be, dealt with in other Bluetooth profiles.

This profile defines how PPP networking is supported in the following situations.

a) LAN Access for a single Bluetooth device.

b) LAN Access for multiple Bluetooth devices.

c) PC to PC (using PPP networking over serial cable emulation).

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile does have dependencies – direct and indirect – on the profile(s) within which it is contained, as illustrated in the figure.
In particular, this LAN Access profile is dependent on the Serial Port and Generic Access profiles.

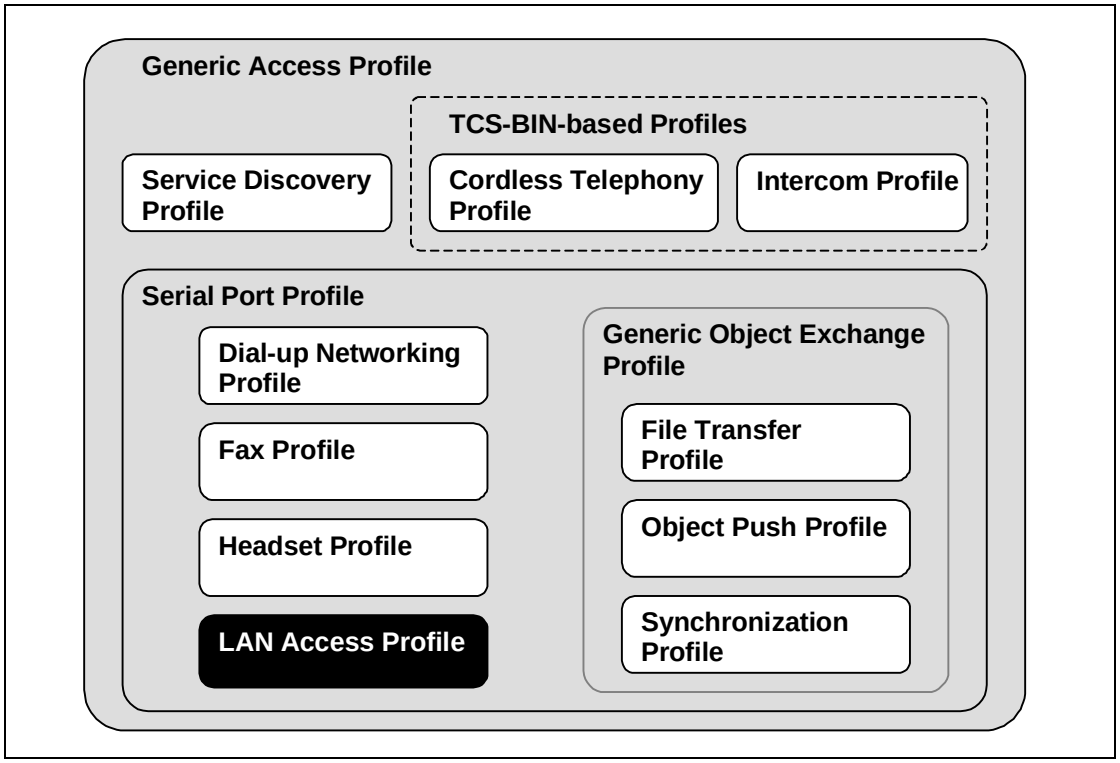


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [13].

2 PROFILE OVERVIEW

2.1 PROTOCOL STACK

The figure below shows the protocols and entities used in this profile.

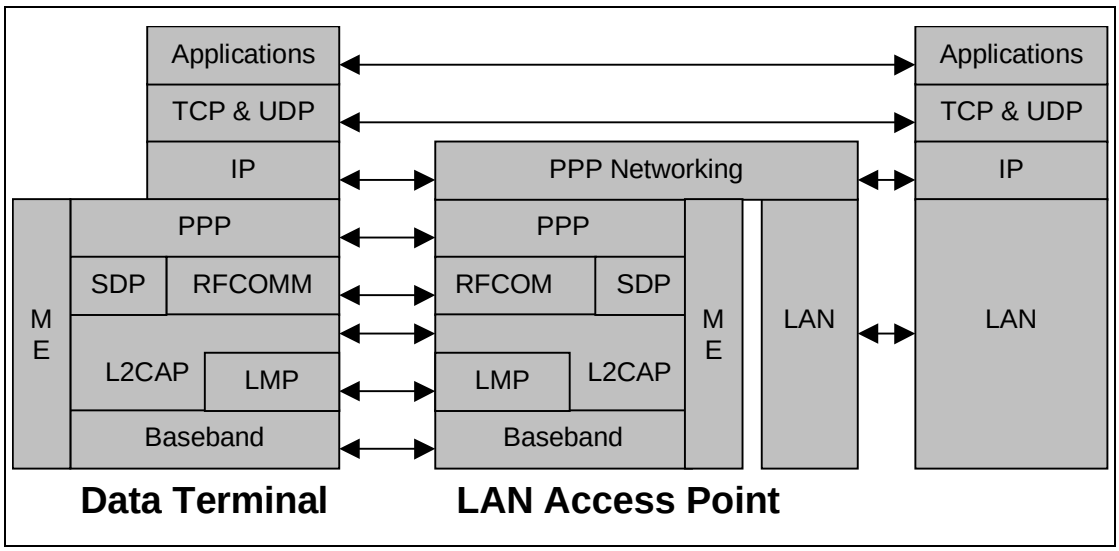


Figure 2.1: Protocol Stack

PPP is the IETF Point-to-Point Protocol [8]. PPP-Networking is the means of taking IP packets to/from the PPP layer and placing them onto the LAN. This mechanism is not defined by this profile but is a well-understood feature of Remote Access Server products.

The Baseband [1], LMP [2] and L2CAP [3] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [4] is the Bluetooth adaptation of GSM TS 07.10 [5]. SDP is the Bluetooth Service Discovery Protocol [6].

ME is the Management Entity which coordinates procedures during initialization, configuration and connection management.

2.2 CONFIGURATIONS AND ROLES

The following roles are defined for this profile.

LAN Access Point **(LAP) –** This is the Bluetooth device that provides access to a LAN (e.g. Ethernet, Token Ring, Fiber Channel, Cable Modem, Firewire, USB, Home Networking). The LAP provides the services of a PPP Server. The PPP connection is carried over RFCOMM. RFCOMM is used to transport the PPP packets and it can also be used for flow control of the PPP data stream.

Data Terminal **(DT)** – This is the device that uses the services of the LAP. Typical devices acting as data terminals are laptops, notebooks, desktop PCs and PDAs. The DT is a PPP client. It forms a PPP connection with a LAP in order to gain access to a LAN.

This profile assumes that the LAP and the DT each have a single Bluetooth radio.[1]

2.3 USER REQUIREMENTS AND SCENARIOS

The following scenarios are covered by this profile.

1. A single DT uses a LAP as a wireless means for connecting to a Local Area Network (LAN). Once connected, the DT will operate as if it were connected to the LAN via dial-up networking. The DT can access all of the services provided by the LAN.

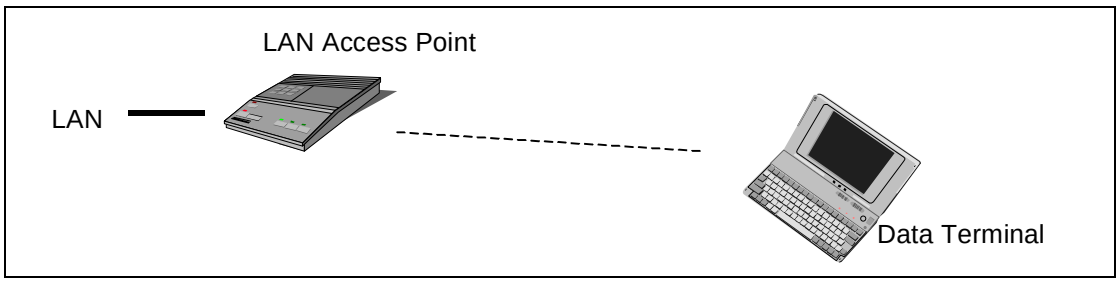


Figure 2.2: LAN Access by a single DT.

1. Products with multiple radios can still conform to this profile. The LAP and DT roles can be adopted independently by each radio.

2. Multiple DTs use a LAP as a wireless means for connecting to a Local Area Network (LAN). Once connected, the DTs will operate as if they were connected to the LAN via dial-up networking. The DTs can access all of the services provided by the LAN. The DTs can also communicate with each other via the LAP.[2]

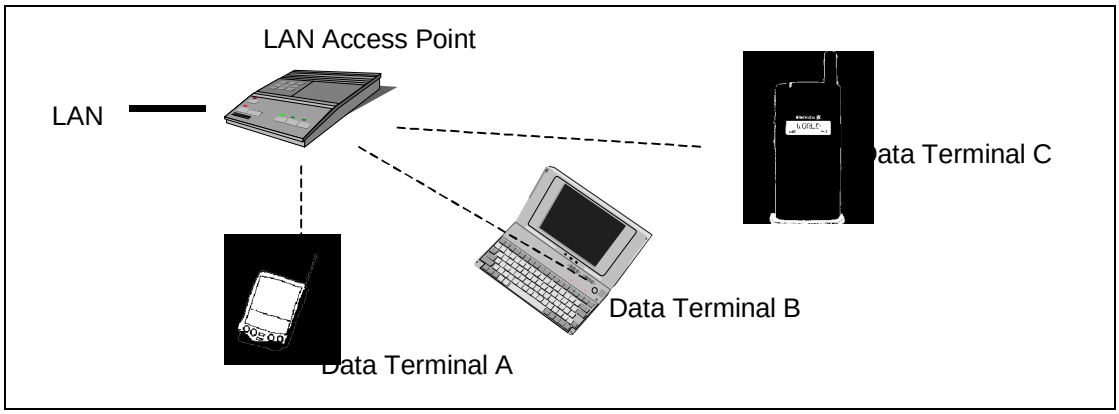


Figure 2.3: LAN Access by multiple DTs.

3. PC to PC connection. This is where two Bluetooth devices can form a single connection with each other. This scenario is similar to a direct cable connection commonly used to connect two PCs. In this scenario, one of the devices will take the role of a LAP, the other will be a DT. See Appendix 13.1 for more details of how this can be configured.

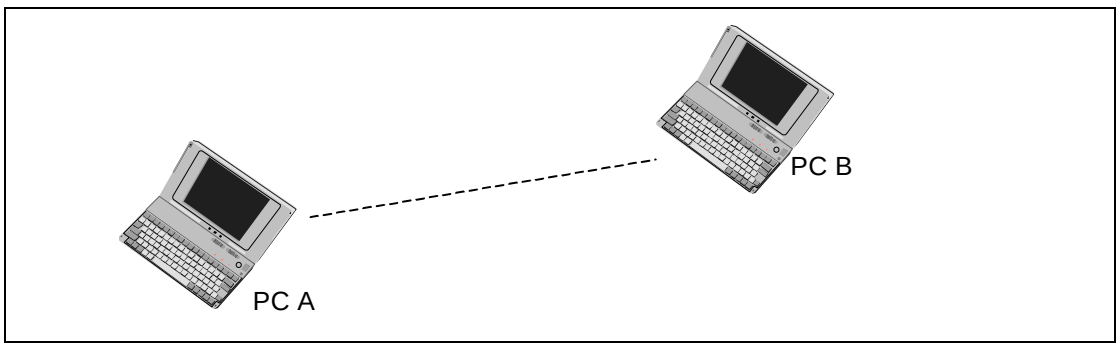


Figure 2.4: PC to PC connection.

Some LAP products may have an internal LAN or use the PSTN to access the Internet or corporate networks. The dial-up mechanisms to achieve the Internet connection are specific to the LAP. The DTs are totally unaware of these activities – except maybe in the event of longer connection times and traffic delays.

2. The DTs will be able to communicate with each other only if the required services (e.g. DNS) are available on the LAN.

2.4 PROFILE FUNDAMENTALS

Here is a brief summary of the interactions between a DT and aLAP. Subsequent sections in this profile provide more detail for each of the following steps.

1. The first step is to find a LAP that is within radio range and is providing a PPP/RFCOMM/L2CAP service. For example, the DT user could use some application to find and select a suitable LAP.
2. If there is no existing baseband physical link, then the DT requests a baseband physical link with the selected LAP. At some point after the physical link establishment, the devices perform mutual authentication. Each device insists that encryption is used on the link – see Section 3.1.
3. The DT establishes a PPP/RFCOMM/L2CAP connection.
4. Optionally, the LAP may use some appropriate PPP authentication mechanism (e.g. CHAP [21]). For example, the LAP may challenge the DT's user to authenticate himself or herself; the DT must then supply a username and password. If these mechanisms are used and the DT fails to authenticate itself, then the PPP link will be dropped.
5. Using the appropriate PPP mechanisms, a suitable IP address is negotiated between the LAP and the DT.
6. IP traffic can now flow across the PPP connection.
7. At any time the DT or the LAP may terminate the PPP connection.

2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 USER INTERFACE ASPECTS

This profile is built upon the Generic Access Profile.

- When reading Generic Access Profile, DevA (the connection initiator) is equivalent to DT, and DevB is equivalent to the LAP.
- All the mandatory requirements defined in Generic Access Profile are mandatory for this profile.
- Unless otherwise stated below, all the optional requirements defined in Generic Access Profile are optional for this profile.

3.1 SECURITY

It is recognized that security in a wireless environment is of paramount importance.

Both the LAP and the DT must enforce that encryption is operating on the baseband physical link while any PPP traffic is being sent or received. The LAP and the DT will refuse any request to disable encryption. Therefore, Bluetooth pairing must occur as a means of authenticating the users. A PIN or link key must be supplied, even if the default PIN is used. The default PIN for LAN access is one single byte with all bits set to 0. Failure to complete the pairing process will prevent access to the LAN Access service.

A more sophisticated product may require further authentication, encryption and/or authorization.

3.2 GENERIC MODES

The following modes are defined in Section 4 of Generic Access Profile [13]. This profile requires the following support.

Modes	Support in LAP	Support in DT
Discoverability modes		
Non-discoverable mode	O	X
Limited discoverable mode	X	X
General discoverable mode	M	X
Connectability modes		
Non-connectable mode	O	X
Connectable mode	M	X
Pairing modes		
Non-pairable mode	O	X
Pairable mode	M	X

Table 3.1: Generic Mode requirements table

Notes

1. A typical use for the Non-discoverable mode is where the LAP is intended for personal use only. The DT would remember the identity of the LAP and never need to use the Bluetooth inquiry mechanism.
2. A typical use for the General discoverable mode is where the LAP is intended for general use. The DT would not be expected to remember the identity of all the LAPs that it uses. The DT is expected to use the Bluetooth inquiry mechanism to discover the LAPs in range.

3.3 ADDITIONAL PARAMETERS

The following parameter is mandatory for the LAP. Optionally it may be configurable by the LAP administrator.

Maximum number of users. Different products have different capabilities and resource limitations that will limit the number of simultaneous users that they can support. The administrator of the LAP may choose to further limit the number of simultaneous users.[3]

- **Single-user mode** is when the maximum number of users is configured to allow only a single user. In this mode, either the DT or the LAP may be the master of the piconet. Single-user mode means that a single DT has exclusive access to a LAP.[4]
- **Multi-user mode** is when the maximum number of users is configured to allow more than one user. In this mode, the LAP must always become the master of the piconet. If the DT refuses to allow the LAP to become master, then the DT cannot gain access to the LAN.

3. The fewer simultaneous users there are using a Bluetooth radio, the more bandwidth will be available to each. A LAP can be restricted to a single user.
4. There are situations where a DT may wish to connect to a LAP and still remain the master of an existing piconet. For example, a PC is the master of a piconet with connections to a Bluetooth mouse and a Bluetooth video projector. The PC then requires a connection to the LAP, but must remain master of the existing piconet. If, for some reason, the PC can only be a member of one piconet, then the LAP must be a piconet slave. This situation is only possible if the LAP's 'maximum number of users' parameter has been configured to 1; i.e. single-user mode.

4 APPLICATION LAYER

Section	Feature	Support in LAP	Support in DT
4.1	Initialization of LAN Access service	M	X
4.2	Shutdown of LAN Access service	M	X
4.3	Establish LAN Connection	C1	M
4.4	Lost LAN Connection	X	M
4.5	Disconnect LAN Connection	M	M

Table 4.1: Application-layer requirements table

C1: Currently the LAP is not required to initiate a LAN connection establishment. In the future, a LAP may initiate a connection (e.g. as part of some form of LAP-initiated hand-off).

4.1 INITIALIZATION OF LAN ACCESS POINT SERVICE

This procedure initiates the configuration of the device as a LAP. This operation involves setting the following parameters:

- All the configurable parameters defined in Section 3.2. (For example, maximum number of users, discoverability mode, etc.)
- The required Bluetooth PINs and/or link keys.
- Any appropriate PPP configuration options (e.g. authentication, compression) can be configured. In order to ensure interoperability, a LAP shall not require connecting DTs to perform any PPP authentication, until the LAP administrator has configured PPP authentication.

When initialization is complete, the device will be able to accept PPP connections.

For products whose main role is that of a LAP, this initialization procedure is typically run automatically when the device is powered up.

For other products (e.g. PCs, Notebooks, etc.), this initialization procedure allows the user to configure the product as an access point, so that a DT can connect to it.

4.2 SHUTDOWN OF LAN ACCESS POINT SERVICE

This procedure stops the device from acting as a LAP.

- The PPP Server is shutdown – as defined in Section 5.2.
- Optionally, a product may take steps to prevent un-authorized Bluetooth access at a later time by deleting some of the stored link keys.

4.3 ESTABLISH LAN CONNECTION

Normally the DT will initiate the establishment of a connection to the LAN.

1. The first step is to select a LAP and a suitable PPP/RFCOMM service that it provides. This selection may be done in one of the following ways:

- The DT user is presented with a list of LAPs that are within radio range, and the services that they provide. The user can then select a LAP/service from the list provided.
- The DT user is presented with a list of services that are being provided by the LAPs that are within radio range. Where the same service is provided by multiple LAPs (i.e. identical ServiceClass-IDs), the application may choose to show the service only once. The user can then select a service from the list provided. The DT will automatically select a suitable LAP that provides the selected service.
- The DT user enters the name of the service that is required, e.g. ‘network’, or ‘Meeting #1’ (see Section 7.1 for more information on service names). The DT will automatically select a suitable LAP that provides the required service.
- Some application on the DT automatically searches for and selects a suitable LAP/service.

Whatever means is used, the result of the selection process must be a LAP that is within radio range and a PPP/RFCOMM service that it provides.

In all cases, the Bluetooth Service Discovery mechanisms are used to retrieve service information. This service information provides all the information required to create the RFCOMM connection in step 4.

2. Optionally, the DT user (or application) is allowed to enter a Bluetooth Authentication PIN or link key supplied by the application. If no PIN is entered, then a zero-length PIN is used.
3. Optionally, the DT user (or application) is allowed to enter a username and password for PPP authentication. If some PPP authentication mechanism is used and the user does not initially supply the username and password, then he/she may be prompted for them later in the connection establishment.

4. When the user (or application) activates the connection, then a PPP application is started, to attempt to establish a connection to the selected access point/service using the procedures in 12.1.

More complex situations (e.g. hand-off of a DT between LAPs) may require the LAP to initiate the establishment of a connection. These situations are possible, but are outside the scope of this document.

4.4 LOST LAN CONNECTION

If the LAN connection is lost for any reason, then the DT user (or application) must be notified of connection failure.

Optionally, the application may allow re-establishment of the connection to the same (or similar) LAP/service. The application could remember the previous LAP, service, PIN, link key, username and password and use them to allow speedy or automatic re-establishment of the LAN connection. The procedures described in Section 5.1 will be used.

4.5 DISCONNECT LAN CONNECTION

Either the LAP or the DT may terminate the LAN Connection at any time – using the procedures in Section 5.4.

5 PPP

PPP/RFCOMM operation in this profile is similar to PPP operation in normal dial-up networking, except that this profile omits the use of AT commands; PPP starts as soon as the RFCOMM link is established. By contrast, in dial-up networking, AT commands are used to establish the link, then PPP starts communicating.

The LAP exports a PPP Server interface [8]. This specification does not require any particular means of achieving the 'appearance' of a PPP Server. One implementation of a LAP could contain a PPP Server. Alternatively, the LAP could be some kind of PPP proxy, where PPP packets are transferred to/from a PPP server somewhere else on the network.

The following text, together with the associated sub-clauses, defines the mandatory requirements with regard to this profile.

Section	Procedure	Support in LAP	Support in DT
5.1	Initialize PPP	M	X
5.2	Shutdown PPP	M	X
5.3	Establish PPP Connection	M	M
5.4	Disconnect PPP Connection	M	M
5.5	PPP Authentication Protocols	O	O

Table 5.1: PPP capabilities

5.1 INITIALIZE PPP

On the LAP, the existence of a PPP Server shall be registered in the Service Discovery Database. The service attributes are defined in 7.1.

A device in the DT role does not register PPP in the Service Discovery Database. However, it is possible for a device to be both a LAP and a DT; therefore the device could register PPP in the Service Discovery Database as defined above.

PPP is a packet-oriented protocol, whereas RFCOMM expects serial data streams. Therefore, the PPP layer must use the serialization mechanisms described in [9].

5.2 SHUTDOWN PPP

All existing PPP connections are disconnected.

The PPP layer disables or removes the PPP service entry from the Service Discovery Database.

5.3 ESTABLISH PPP CONNECTION

If there is no existing RFCOMM session between the LAP and the DT, then the device initiating the PPP connection shall first initialize RFCOMM (see Section 6). The device obtains the RFCOMM Server channel to use from the service information it discovered earlier.

Using the Link Control Protocol (LCP) [8], the LAP and DT negotiate a PPP link.

Part of the LCP is the negotiation of the Maximum Transmission Unit (MTU) to be used on the PPP link – see [8] for details. This profile places no requirements on the negotiated MTU.[5]

Depending upon its capabilities and configuration (see 3.2), a LAP may have multiple PPP sessions in operation simultaneously.

5.4 DISCONNECT PPP CONNECTION

The following reasons will cause PPP to terminate the connection:

1. User intervention.
2. Failure of the RFCOMM/L2CAP connection. The RFCOMM/L2CAP connection may fail for several reasons. For example, when the radio link has failed or the device has been out of range for an excessive amount of time; see [3].
3. Termination by the LAP, if the access point can no longer provide the appropriate service. The reasons that would cause this are very dependent on the implementation of the LAP, but they could include (a) detection of duplicate IP addresses, (b) loss of connection to the LAN, (c) loss of connectivity to the PPP Server, or (d) loss of connection to the required IP subnet.
4. Some implementation-specific policy decision made by an application that is running on the LAP or the DT.

PPP handles each of the above situations differently. Reasons 1, 3 and 4 above result in a controlled disconnection at each protocol layer. Reason 2 above requires different processing.

5. Some products may use the LCP negotiation process to insist on specific values for the MTU. For example, a simple LAP with an Ethernet connection may wish to have a suitable MTU, so that IP packets do not require fragmentation when transmitted from Bluetooth to Ethernet.

When the PPP connection is terminated, either by user intervention or automatically by the LAP, then the PPP layer takes the following steps:

1. Gracefully terminate the IPCP connections (as defined in [24]). This will cause the IP interface to be disabled.
2. Gracefully terminate the LCP connections (as defined in [8]),
3. Disconnect the RFCOMM connection (as defined in Serial Port Profile)

When the RFCOMM/L2CAP connection suddenly terminates, then the PPP layer takes the following steps:

1. Terminate the IPCP connections (as defined in [24]). This will cause the IP interface to be disabled.
2. Terminate the LCP connections (as defined in [8]).

5.5 PPP AUTHENTICATION PROTOCOLS

Optionally, a LAP may be configured to use one or more of the PPP authentication protocols. These protocols allow a network administrator to control access to the network. The use of these PPP protocols does not form part of this profile. They are mentioned here for information only.

PPP supports a number of authentication protocols including the following:

- PPP Challenge Handshake Authentication Protocol (CHAP) [21]
- Microsoft PPP CHAP Extensions [22]
- PPP Authentication [23]
- PPP Extensible Authentication Protocol (EAP) [23]

Typically, the user needs to supply a username and password in order to gain authorization to use the PPP connection. If the authentication fails, then the PPP connection is normally dropped.

The PPP authentication protocols are independent of the Bluetooth authentication mechanisms. A network administrator may choose to use any combination of the PPP and Bluetooth mechanisms.

6 RFCOMM

This section describes the requirements on RFCOMM in units complying with the LAN Access Profile.

This profile is built upon the Serial Port Profile [10]. The requirements defined in the Serial Port Profile Section 4, “RFCOMM Interoperability Requirements,” on page 184, apply to this profile.

- When reading [10], DevA (the connection initiator) is equivalent to DT and DevB is equivalent to the LAP.
- All the mandatory requirements defined in the Serial Port Profile Section 4 on page 184 are mandatory for this profile.
- All the optional requirements defined in the Serial Port Profile Section 4 on page 184 are optional for this profile.

In addition:

1. In order to maximize packet throughput, it is recommended that RFCOMM should make use of the 3 and 5 slot baseband packets.
2. As defined in [4] section 2, the speed of RFCOMM connections is not configurable by the user. RFCOMM will transfer the data as fast as it can. The actual transfer rate will vary, depending upon the other Bluetooth traffic on the baseband link. In particular, the connection speed will not be artificially held at some typical serial port value; e.g. 19200.

7 SERVICE DISCOVERY

A LAP will be capable of providing one or more services for connecting to a LAN. For example, different services could provide access to different IP sub-nets on the LAN. The DT's user must be able to choose which of the LAN access services he or she requires.

7.1 SDP SERVICE RECORDS

Each LAP will provide one Service Class for PPP/RFCOMM services. A LAP may contain multiple instances of this Service Class; e.g. access to multiple subnets. Where the access point provides more than one PPP/RFCOMM service, the service selection is based on service attributes. These services are made public via the SDP.

The service record will have the following attributes. The syntax and usage of these attributes is defined in [6].

Item	Definition	Mand. /Opt.	Type	Value	Default Value
ServiceClassIDList		Mand.			
ServiceClass0	UUID for "LAN Access using PPP"	Mand.	UUID	See [11]	See [11]
ProtocolDescriptorList		Mand.			
Protocol0	UUID for L2CAP protocol	Mand.	UUID	See [11]	See [11]
Protocol1	UUID for RFCOMM protocol	Mand.	UUID	See [11]	See [11]
Parameter0	Server channel	Mand.	UInt8	varies	varies
ProfileDescriptorList		Opt.			
Profile #0	Uuid for "LAN Access using PPP"		UUID	see [11]	see [11]
Parameter0	Version "1.00"		Uint16	0x0100	0x0100
ServiceName	Displayable name	Opt.	String	Configurable	'LAN Access using PPP'
ServiceDescription	Displayable Information	Opt.	String	Configurable	'LAN Access using PPP'
ServiceAvailability	Load Factor	Opt.	Uint8	Dynamic	Dynamic
IpSubnet	Displayable Information	Opt.	String	Configurable	Configurable

The actual values of universal attribute IDs are defined in the Assigned Numbers specification [11] section 4. Values that are of the type UUID are defined in the Assigned Numbers specification [11] section 4.

- The ServiceName attribute is a short user-friendly name for the service; e.g. ‘Corporate Network’, ‘Conference#1’, etc.
- The ServiceDescription attribute is a longer description of the service. For example. “This network is provided for our guests. It provides free Internet Access and printing services. No username or password are required.“
- The ServiceAvailability attribute may be used in conjunction with the Load-Factor field of the CoD defined for LAN Access Points – see [11] section 1.2.6.
- The IpSubnet attributeID is (0x0200). This attribute is a displayable string containing subnet definition of the network, e.g. “191.34.12.0/24”. The first 4 numbers define the IP subnet in dotted-decimal notation. The fifth number, after the “/” character, is the number of subnet bits to use in the subnet mask; e.g. 24 means a subnet mask of 255.255.255.0.

8 L2CAP

This section describes the requirements on L2CAP in units complying with the LAN Access Profile.

This profile is built upon the Serial Port Profile [10]. The requirements defined in the Serial Port Profile Section 5, “L2CAP Interoperability Requirements,” on page 186 apply to this profile.

- When reading [10], DevA (the connection initiator) is equivalent to DT and DevB is equivalent to the LAP.
- All the mandatory requirements defined in the Serial Port Profile section 5 on page 186 are mandatory for this profile.
- All the optional requirements defined in the Serial Port Profile Section 5 on page 186 are optional for this profile.

In addition:

1. The MTU used at the L2CAP layer is determined by the RFCOMM parameter ‘maximum frame size’ – see Section 6 on page 292.

9 LINK MANAGER

This section describes the requirements on Link Manager in units complying with the LAN Access Profile.

This profile is built upon the Serial Port Profile [10]. The requirements defined in the Serial Port Profile Section 7, “Link Manager (LM) Interoperability Requirements,” on page190 apply to this profile.

- When reading [10], DevA (the connection initiator) is equivalent to DT and DevB is equivalent to the LAP.
- All the mandatory requirements defined in the Serial Port Profile Section 7 on page 190 are mandatory for this profile.
- The following optional requirements defined in the Serial Port Profile Section 7 on page 190 are mandatory for this profile.

Procedure	Support in LAP	Support in DT
Authentication	M	M
Pairing	M	M
Encryption	M	M
Request master/slave switch	M	X
Perform master/slave switch	M	M

Table 9.1: LMP procedures

- All the remaining optional requirements defined in the Serial Port Profile Section 7 on page 190 are optional for this profile.

In addition:

- For bandwidth reasons, it is advisable but not mandatory for both devices to use multi-slot packets.
- When the LAP is configured in single-user mode (i.e. maximum number of users is 1), then the LAP may be either the master or the slave of the pico-net.
- When the LAP is configured in multi-user mode (i.e. maximum number of users is more than 1), then the LAP must be the master of the piconet.

9.1 PROFILE ERRORS

The LAP must deny access to the PPP service if the DT fails to comply with the requirements of this profile, as follows:

1. Failure to complete the pairing process.
2. Failure to comply with a request to enable encryption on the baseband connection.
3. Failure by the DT to comply with a request to perform a master/slave switch. The LAP only requests a master/slave switch when it is configured in multi-user mode. In this mode the LAP must be the master of the piconet.

The LAP must reject all attempts by the DT to perform the following operations (see [2] section 5.1.2 for the appropriate LMP rejection reasons):

4. Requesting that encryption be disabled. The error code “Host Rejected due to security reasons” is used.
5. Requesting that the LAP switch to be a slave when the LAP is configured to be in multi-user mode. The error code “Unspecified Error” is used.
6. Requesting that a new connection be made when the LAP already has its configured maximum number of users. The error code “Other End Terminated Connection: Low Resources” is used.

10 LINK CONTROL

This section describes the requirements on Link Control in units complying with the LAN Access Profile.

This profile is built upon the Serial Port Profile [10]. The requirements defined in the Serial Port Profile, Section 8, “Link Control (LC) Interoperability Requirements,” on page 191, apply to this profile.

- When reading [10], DevA (the connection initiator) is equivalent to DT and DevB is equivalent to the LAP.
- All the mandatory requirements defined in the Serial Port Profile, Section 8 on page 191, are mandatory for this profile.
- All the optional requirements defined in the Serial Port Profile, Section 8 on page 191, are optional for this profile.
- The timer definitions defined in the Serial Port Profile, Section 8 on page 191, are not used in this profile.

In addition:

1. The Non-discoverable and General Discoverable Modes of the LAP (i.e. how InquiryScan is used) are defined in the Generic Access Profile [13], Section 4 on page 30.
2. In order to discover the nearby LAPs, a DT must use the General Inquiry procedure defined in the Generic Access Profile [13], Section 6 on page 38.

A device taking on the role of a LAP must set the Networking bit (#17) in the CoD field in FHS packets sent. Additionally, it is recommended that devices whose primary function is to provide network access to other DTs should set the major device class to “LAN access point”.

11 MANAGEMENT ENTITY PROCEDURES

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

Section	Procedure	Support in LAP	Support in DT
11.1	Link Establishment	M	M
11.2	Single/Multi-user mode	M	N/A

Table 11.1: Management Entity Procedures

11.1 LINK ESTABLISHMENT

Link Establishment is required for communication between a LAP and a DT. The Link Establishment procedure is started as a direct consequence of the user operations described in "Establish LAN Connection" Section 4.3.

1. The DT first performs a General Inquiry to discover what LAPs are within radio range, see Generic Access Profile, Section 6 on page 38. Having performed the inquiry, the DT will have gathered a list of responses from nearby LAPs.
2. The DT sorts the list according to some product-specific criteria. The LAN Access Point CoD contains a field called 'Load Factor', see [11] section 1.2.6. It is recommended (but not mandated) that this field is used to sort the list.
3. The DT shall start with the LAP at the top of the list and try to establish a link with it, see Generic Access Profile, Section 7.1 on page 46. Any error or failure to establish a link shall cause the DT to skip thisLAP. The DT will attempt to establish a link the next LAP in the list.
4. If there are no more LAPs in the list, the DT shall not proceed with further link establishment procedures. Link establishment has to be re-initiated.

The following subsections apply.

11.1.1 No responses to inquiry

If the DT did not get any response during inquiry, the DT shall not proceed with further link establishment procedures. Link establishment has to be re-initiated by the user or an application.

11.1.2 No response to paging

If a LAP does not respond to paging attempts, the DT shall skip this LAP.

11.1.3 Pairing

During link establishment, the LAP and DT are paired, which means that the DT and LAP build a security wall towards other devices.

11.1.4 Errors

If any LM procedure or Service Discovery procedure fails, or if link is lost at any time during link establishment, then the DT shall skip this LAP.

11.2 MAXIMUM NUMBER OF USERS

When the LAP is configured to allow multiple users, then the LAP must be the master of the piconet, see 3.2. In this mode, the Management Entity on the LAP ensures that the LAP remains the master of the Bluetooth piconet.

While in multi-user mode, the LAP shall request that it become the master of any new baseband physical link. If, for any reason, the LAP cannot remain the master, then the baseband physical link shall fail. The LMP [2] allows a device to (a) request a master/slave switch, and also (b) to refuse to comply with a request to perform a master/slave switch, see [1] section 10.9.3.

12 APPENDIX A (NORMATIVE): TIMERS AND COUNTERS

No specific timers are required by this profile.

Timer name	Recommended value	Description	Comment

Table 12.1: Defined timers

No specific counters are required by this profile.

Counter name	Proposed value	Description	Comment

Table 12.2: Defined Counters

The following parameters are required by this profile.

Parameter	Description
Discoverability mode	Controls whether the DT can discover the LAP.
Connectability mode	Controls whether the DT can be connected to the LAP.
Pairing mode	Controls whether the DT can be pair with the LAP.
Maximum users	The maximum the number of simultaneous users/connections.

Table 12.3: Defined parameters

13 APPENDIX B (NORMATIVE): MICROSOFT WINDOWS

This section contains various bits of information relating to Microsoft Windows and how it can be used in this profile.

13.1 PC-2-PC CONFIGURATION

This section contains information for configuring two PCs to form a connection. This configuration is independent of Bluetooth. This configuration is the same whether a serial cable or Bluetooth is used to make the connection.

- It is known that Windows '98 comes with a PPP server and that this PPP Server can be used to achieve the PC-to-PC feature. Detailed configuration information is available at the following Web sites.

  Microsoft Direct Cable Connection & Dial-up networking:
  http://support.microsoft.com/support/windows/ServiceWare/
  Win95/33BKKC22.ASP
  http://www.wown.com/
  http://www.tecno.demon.co.uk/dcc.html
  http://www.cs.purdue.edu/homes/kime/directcc/directcc95.htm

- This application requires some exchange of text strings before the PPP connection will become operational. The client PC sends the string ‘CLIENT’ and the server must reply with ‘CLIENTSERVER’.
- The tools provided by Windows ’98 configure one PC as the server and the other as the client. The PC configured as the server can share its resources with the client, but not vice versa.

14 APPENDIX C (INFORMATIVE): INTERNET PROTOCOL (IP)

The use of IP in this profile is optional. This section is provided for information only.

This section contains various bits of IP information that relate to various parts of this profile.

14.1 IP INTERFACES

14.1.1 Interface Enabled

The PPP layer in the DT will enable an IP interface when the IPCP link has been established and a suitable IP address has been negotiated. Typically, the DT will only have one PPP session in operation and only need one IP interface.

The DT may also need to configure its default gateway – WINS, DNS, etc. This profile does not define how this configuration is achieved. Mechanisms exist within PPP for supplying this information, see [24]. Other mechanisms may be used as appropriate.

In the event a DT has multiple IP interfaces, we rely on the IP protocol layer within the DT to select the correct interface to use for transmitting packets.

14.1.2 Interface Disabled

When the PPP connection is terminated or aborted, then the IP interface is disabled. The IP protocol stack will then remove that IP address from its routing tables.

14.2 THE IPCP PROTOCOL

Optionally, a LAP may be configured to support the IP protocol. The use of this PPP protocol does not form part of this profile. It is mentioned here for information only.

If supported, the IPCP protocol must be fully supported as defined in [24].

The following sub-sections concerning IPCP are informational only. They briefly describe certain aspects of IPCP. See [24] for full details.

14.2.1 IPCP Connection

IPCP only starts to operate after (a) the PPP connection has been established using LCP and optionally (b) the user has been authenticated.

The IPCP protocol negotiates certain parameters between the LAP and the DT.

Once the IPCP connection is established, and a suitable IP address has been negotiated, then IP interface is enabled.

14.2.2 IP Address Allocation

The DT will require an IP address in order to operate on the LAN. Current PPP implementations allow only three possibilities:

1. The IPCP option is used to specify a pre-configured IP address. If this IP address is not suitable on the LAN, then the IPCP link will not be established.
2. The IPCP option is used to request a suitable IP configuration from a PPP Server.
3. The IPCP Mobile-IP options are used to request a specified IP configuration. When moving between access points on the same LAN, it may be advantageous for the DT to continue using the same IP configuration.

14.2.3 DNS and NBNS addresses

Optionally, the LAP support could include the IPCP extensions defined in RFC1877 (defined by Microsoft). These extensions define the negotiation of primary and secondary Domain Name System (DNS) and NetBIOS Name Server (NBNS) addresses.

14.2.4 NetBIOS over IP

The NetBIOS protocol is used by Microsoft Windows to implement many of its networking features. The NetBIOS protocol can be carried over IP packets as defined in [29] and [30].

15 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles....270
Figure 2.1: Protocol Stack....271
Figure 2.2: LAN Access by a single DT....272
Figure 2.3: LAN Access by multiple DTs....273
Figure 2.4: PC to PC connection....273

16 LIST OF TABLES

Table 3.1: Generic Mode requirements table 276
Table 4.1: Application-layer requirements table 278
Table 5.1: PPP capabilities 281
Table 9.1: LMP procedures 288
Table 11.1: Management Entity Procedures 291
Table 12.1: Defined timers 293
Table 12.2: Defined Counters 293
Table 12.3: Defined parameters 293

17 REFERENCES

17.1 NORMATIVE REFERENCES

[1] Bluetooth Baseband specification (See Volume 1, Part B)

[2] Bluetooth Link Manager Protocol (See Volume 1, Part C)

[3] Bluetooth Logical Link Control and Adaptation Protocol Specification (See Volume 1, Part D)

[4] RFCOMM with TS 07.10 (See Volume 1, Part F:1)

[5] TS 101 369 (GSM 07.10) version 6.2.0.

[6] Bluetooth Service Discovery Protocol (SDP) (See Volume 1, Part E)

[7] John Webb, "Bluetooth SIG MRD", version 1.0.

[8] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 50, RFC 1661, Daydreamer, July 1994.

[9] Simpson, W., Editor, "PPP in HDLC Framing", STD 51, RFC 1662, Daydreamer, July 1994.

[10] Serial Port Profile

[11] Bluetooth Assigned Numbers (http://www.bluetooth.org/assigned-numbers.htm)

[12] Thomas Miller, “Bluetooth Security Architecture”. Version 1.0.

[13] Generic Access Profile

17.2 INFORMATIVE REFERENCES

[20] Lloyd, B., and W. Simpson, "PPP Authentication Protocols", RFC 1334, Lloyd Internetworking, Daydreamer, October 1992.

[21] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.

[22] Zorn, G., " Microsoft PPP CHAP Extensions", RFC 2433, October 1998.

[23] L. Blunk., " PPP Extensible Authentication Protocol (EAP)", RFC 2433, March 1998.

[24] McGregor, G., " The PPP Internet Protocol Control Protocol (IPCP)", RFC 1332, May 1992.

[25] Simpson, W., " The PPP Internetwork Packet Exchange Control Protocol (IPXCP)", RFC 1552, December 1993.

[26] Pall, G., " The PPP NetBIOS Frames Control Protocol (NBFCP)”, RFC 2097, January 1997.

[27] " Mobile-IPv4 Configuration Option for PPP IPCP ", RFC 2290.

[28] Cobb, S., " PPP Internet Protocol Control Protocol Extensions for Name Server Addresses", RFC 1877, December 1995.

[29] NetBIOS Working Group, "PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT: CONCEPTS AND METHODS", RFC 1001, March 1987.

[30] NetBIOS Working Group, "PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT: DETAILED SPECIFICATIONS", RFC 1002, March 1987.

Part K:10

GENERIC OBJECT EXCHANGE PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the object exchange usage models. The requirements are expressed by defining the features and procedures that are required for interoperability between Bluetooth devices in the object exchange usage models.

CONTENTS

1 Introduction 306
1.1 Scope 306
1.2 Bluetooth Profile Structure 306
1.3 Bluetooth OBEX-Related Specifications 307
1.4 Symbols and conventions 308
1.4.1 Requirement status symbols 308
1.4.2 Signaling diagram conventions 309

2 Profile overview 310
2.1 Profile stack 310
2.2 Configurations and roles 310
2.3 User requirements and scenarios 311
2.4 Profile fundamentals 311

3 User interface aspects 312

4 Application layer 313
4.1 Feature Overview 313
4.2 Establishing an Object Exchange Session 313
4.3 Pushing a Data Object 313
4.4 Pulling a Data Object 313

5 OBEX Interoperability Requirements 314
5.1 OBEX Operations Used 314
5.2 OBEX Headers 314
5.3 Initialization of OBEX 315
5.4 Establishment of OBEX session 315
5.4.1 OBEX Session without Authentication 316
5.4.2 OBEX Session with Authentication 318
5.5 Pushing Data to Server 321
5.6 Pulling Data from Server 322
5.7 Disconnection 323

6 Serial Port Profile Interoperability Requirements 324
6.1 RFCOMM Interoperability Requirements 324
6.2 L2CAP Interoperability Requirements 324
6.3 SDP Interoperability Requirements 324
6.4 Link Manager (LM) Interoperability Requirements 324
6.5 Link Control (LC) Interoperability Requirements 324
6.5.1 Inquiry and Inquiry Scan 325

7 Generic Access Profile Interoperability Requirements 326

7.1 Modes 326

7.2 Security aspects 326

7.3 Idle mode procedures 327

7.3.1 Bonding 327

8 References 328

8.1 Normative references 328

FOREWORD

The purpose of this document is to work as a generic profile document for all application profiles using the OBEX protocol.

Interoperability between devices from different manufacturers is provided for a specific service and usage model if the devices conform to a Bluetooth SIG defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications and gives an unambiguous description of the air interface for specified service(s) and usage model(s).

All defined features are process-mandatory. This means that if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

The Generic Object Exchange profile defines the protocols and procedures that shall be used by the applications providing the usage models which need the object exchange capabilities. The usage model can be, for example, Synchronization, File Transfer, or Object Push model. The most common devices using these usage models can be notebook PCs, PDAs, smart phones, and mobile phones.

1.2 BLUETOOTH PROFILE STRUCTURE

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly. For example, the Object Push profile is dependent on Generic Object Exchange, Serial Port, and Generic Access profiles.

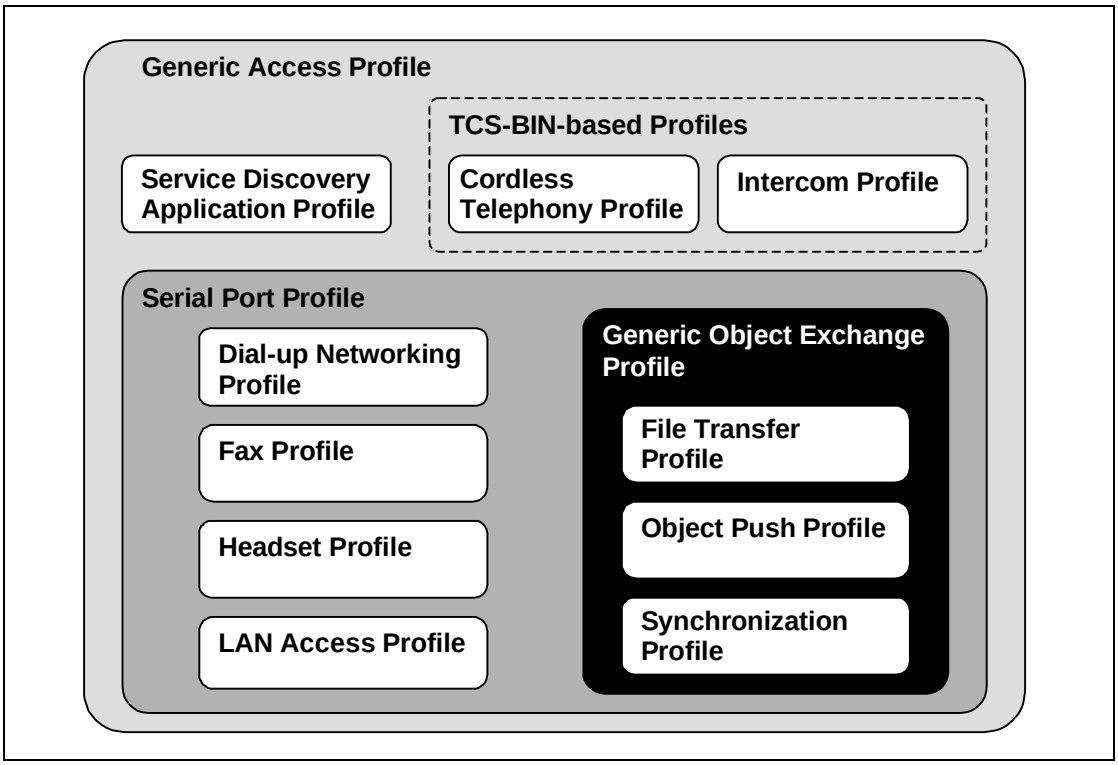


Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX-RELATED SPECIFICATIONS

Bluetooth Specification includes five separate specifications for OBEX and applications using it.

1. Bluetooth IrDA Interoperability Specification [1]

- Defines how the applications can function over both Bluetooth and IrDA.
- Specifies how OBEX is mapped over RFCOMM and TCP.
- Defines the application profiles using OBEX over Bluetooth.

2. Bluetooth Generic Object Exchange Profile Specification (This specification)

- Generic interoperability specification for the application profiles using OBEX.
- Defines the interoperability requirements of the lower protocol layers (e.g. Baseband and LMP) for the application profiles.

3. Bluetooth Synchronization Profile Specification [2]

- Application Profile for the Synchronization applications.
- Defines the interoperability requirements for the applications within the Synchronization application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

4. Bluetooth File Transfer Profile Specification [3]

- Application Profile for the File Transfer applications.
- Defines the interoperability requirements for the applications within the File Transfer application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

5. Bluetooth Object Push Profile Specification [4]

- Application Profile for the Object Push applications.
- Defines the interoperability requirements for the applications within the Object Push application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document, the following symbols are used:

‘M’ for mandatory to support (used for capabilities that shall be used in the profile);

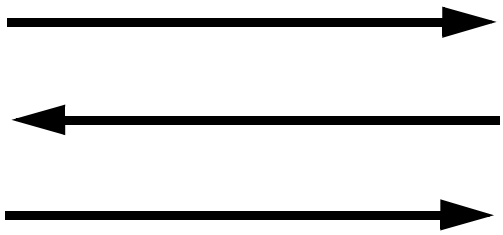
‘O’ for optional to support (used for capabilities that can be used in the profile);

‘C’ for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

‘X’ for capabilities that shall be supported by the unit but shall never be used in the profile);

‘N/A’ for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.



1.4.2 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

A		B
	PROC1	
	PROC2	
	PROC3	
	(PROC4)	
	(PROC5)	
	MSG1	
	MSG2	
	(MSG3)	
	(MSG4)	

Table 1.1: Arrows used in signaling diagrams

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A and B). PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates an optional message from B to A.

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

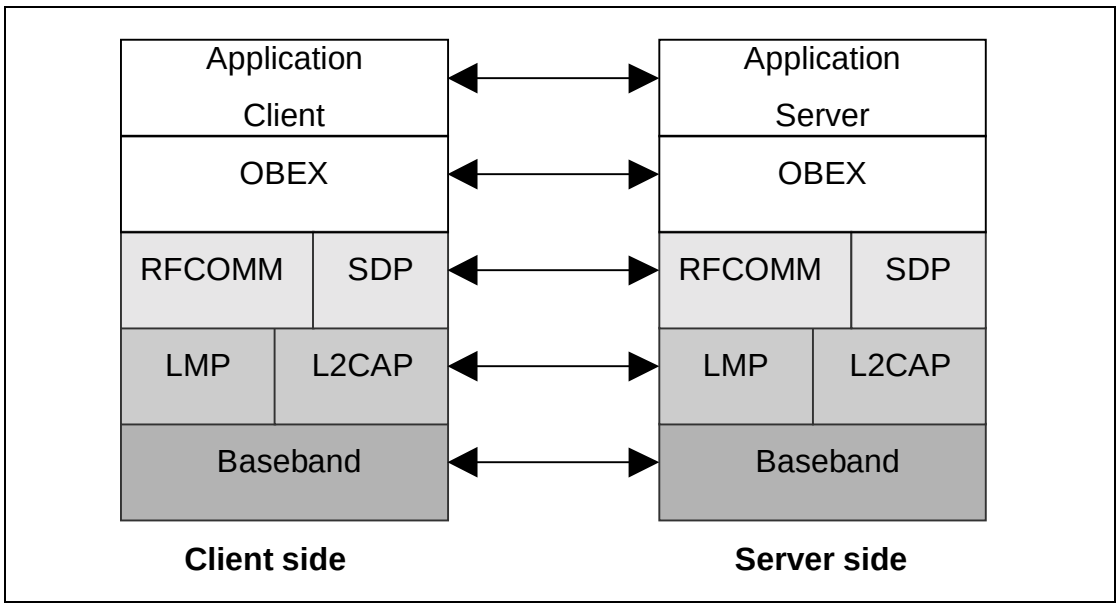


Figure 2.1: Protocol model

The Baseband [5], LMP [6] and L2CAP [7] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [8] is the Bluetooth adaptation of GSM TS 07.10 [9]. SDP is the Bluetooth Service Discovery Protocol [10]. OBEX [1] is the Bluetooth adaptation of IrOBEX [11].

The Application Client layer shown in Figure 2.1 is the entity sending and retrieving data object from the Server using the OBEX operations. The application Server is the data storage to and from which the data object can be sent or retrieved.

2.2 CONFIGURATIONS AND ROLES

The following roles are defined for this profile:

Server – This is the device that provides an object exchange server to and from which data objects can be pushed and pulled, respectively.

Client – This is the device that can push or/and pull data object(s) to and from the Server.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Usage of a Server by a Client to push data object(s)
- Usage of a Server by a Client to pull data object(s)

The following restrictions apply to this profile:

a) For the device containing the Server, it is assumed that the user may have to put it into the discoverable and connectable modes when the inquiry and link establishment procedures, respectively, are processed in the Client (see Generic Access Profile).

b) The profile only supports point-to-point configurations. As a result, the Server is assumed to offer services only for one Client at a time. However, the implementation may offer a possibility for multiple Clients at a time but this is not a requirement.

2.4 PROFILE FUNDAMENTALS

The profile fundamentals, with which all application profiles must comply, are the following:

1. Before a Server is used with a Client for the first time, a bonding procedure including the pairing may be performed (see Section 7.3.1). This procedure must be supported, but its usage is dependent on the application profiles. The bonding typically involves manually activating bonding support and entering a Bluetooth PIN code (see Section 7.3.1) on the keyboards of the Client and Server devices. This procedure may have to be repeated under certain circumstances; for example, if a common link key (as a bonding result) is removed on the device involved in the object exchange.
2. In addition to the link level bonding, an OBEX initialization procedure may be performed (see Section 5.3) before the Client can use the Server for the first time. The application profiles using GOEP must specify whether this procedure must be supported to provide the required security level.
3. Security can be provided by authenticating the other party upon connection establishment, and by encrypting all user data on the link level. The authentication and encryption must be supported by the devices; but whether they are used depends on the application profile usingGOEP.
4. Link and channel establishments must be done according to the procedures defined in GAP (see Section 7.1-7.2 in [14]). Link and channel establishment procedures in addition to the procedures in GAP must not defined by the application profiles using GOEP.
5. There are no fixed master/slave roles.
6. This profile does not require any lower power mode to be used.

3 USER INTERFACE ASPECTS

User interface aspects are not defined in this profile.They are instead defined in the application profiles, where necessary.

4 APPLICATION LAYER

This section describes the service capabilities which can be utilized by the application profiles using GOEP.

4.1 FEATURE OVERVIEW

Table 4.1 shows the features which the Generic Object Exchange profile provides for the application profiles. The usage of other features (e.g. setting the current directory) must be defined by the applications profiles needing them.

Feature no.	Feature
1	Establishing an Object Exchange session
2	Pushing a data object
3	Pulling a data object

Table 4.1: Features provided by GOEP

4.2 ESTABLISHING AN OBJECT EXCHANGE SESSION

This feature is used to establish the object exchange session between the Client and Server. Before a session is established, payload data cannot be exchanged between the Client and the Server. The usage of the OBEX operations for establishing an OBEX session is described in Section 5.4.

4.3 PUSHING A DATA OBJECT

If data needs to be transferred from the Client to the Server, then this feature is used. The usage of the OBEX operations for pushing the data object(s) is described in Section 5.5.

4.4 PULLING A DATA OBJECT

If data need to be transferred from the Server to the Client, then this feature is used. The usage of the OBEX operations for pulling the data object(s) is described in Section 5.6.

5 OBEX INTEROPERABILITY REQUIREMENTS

5.1 OBEX OPERATIONS USED

Table 5.1 shows the OBEX operations which are specified by the OBEX protocol. The application profiles using GOEP must specify which operations must be supported to provide the functionality defined in the application profiles.

Operation no.	OBEX Operation
1	Connect
2	Disconnect
3	Put
4	Get
5	Abort
6	SetPath

Table 5.1: OBEX Operations

The IrOBEX specification does not define how long a client should wait for a response to an OBEX request. However, implementations which do not provide a user interface for canceling an OBEX operation should wait a reasonable period between a request and response before automatically canceling the operation. A reasonable time period is 30 seconds or more.

5.2 OBEX HEADERS

Table 5.2 shows the specified OBEX headers.

Header no.	OBEX Headers
1	Count
2	Name
3	Type
4	Length
5	Time
6	Description
7	Target
8	HTTP

Table 5.2: OBEX Headers

Header no.	OBEX Headers
9	Body
10	End of Body
11	Who
12	Connection ID
13	Authenticate Challenge
14	Authenticate Response
15	Application Parameters
16	Object Class

Table 5.2: OBEX Headers

Applications profiles dedicated to specific usage models must specify which of these headers must be supported.

5.3 INITIALIZATION OF OBEX

If the OBEX authentication is supported and used by the Server and the Client, the initialization for this authentication (see also Section 5.4.2) must be done before the first OBEX connection can be established. The initialization can be done at any time before the first OBEX connection. The initialization of the OBEX authentication requires user intervention on both the Client device and the Server device.

Authentication is done using an OBEX password, which may be the same as a Bluetooth PIN code on the link level. Even if the user uses the same code for link authentication and OBEX authentication, the user must enter these codes separately. After entering the OBEX password in both the Client and Server, the OBEX password is stored in the Client and the Server, and it can be used in the future for authenticating the Client and the Server. When an OBEX connection is established, the devices must authenticate each other if the OBEX authentication is enabled.

5.4 ESTABLISHMENT OF OBEX SESSION

For the Object Exchange, the OBEX connection can be made with or without OBEX authentication. In the next two subsections, both of these cases are explained. All application profiles using GOEP must support an OBEX session without authentication.

5.4.1 OBEX Session without Authentication

Figure 5.1 depicts how an OBEX session is established using the CONNECT operation.

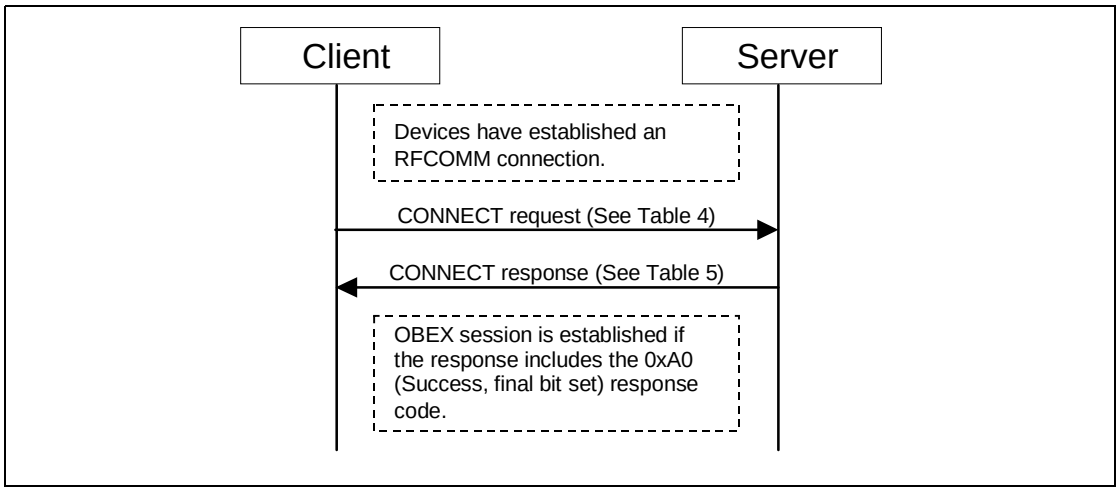


Figure 5.1: Establishment of OBEX Session without Authentication

The CONNECT request indicates a need for connection and may also indicate which service is used. The fields in the CONNECT request are listed below:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for CONNECT	0x80	M	-
Field	Connect Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	Target	Varies	C1	Used to indicate the specific Service.

Table 5.3: Fields and Headers in CONNECT Request

C1: The use of the Target header is mandatory for some application profiles. The application profiles define explicitly whether they use it or not. For the Target header, the example value could be ‘IRMC-SYNC’ to indicate the IrMC synchronization service. The target header is placed after the Maximum OBEX Packet Length field in the CONNECT request.

The response to the CONNECT request includes the fields listed below:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for CONNECT request	Varies	M	0xA0 for success
Field	Connect Response Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	ConnectionID	Varies	C2	The header value specifies the current connection to the specific service.
Header	Who	Varies	C2	The header value matches the Target header value.

Table 5.4: Fields and Headers in CONNECT Response

C2: The Who and Connection ID headers must be used if the Target header is used in the Connect request. These headers are placed after the Maximum OBEX Packet Length field in the response to the CONNECT request.

5.4.2 OBEX Session with Authentication

The OBEX authentication scheme is based on the HTTP scheme but does not have all the features and options. In GOEP, OBEX authentication is used to authenticate the Client and the Server. Figure 5.2 depicts establishment of an OBEX session with authentication.

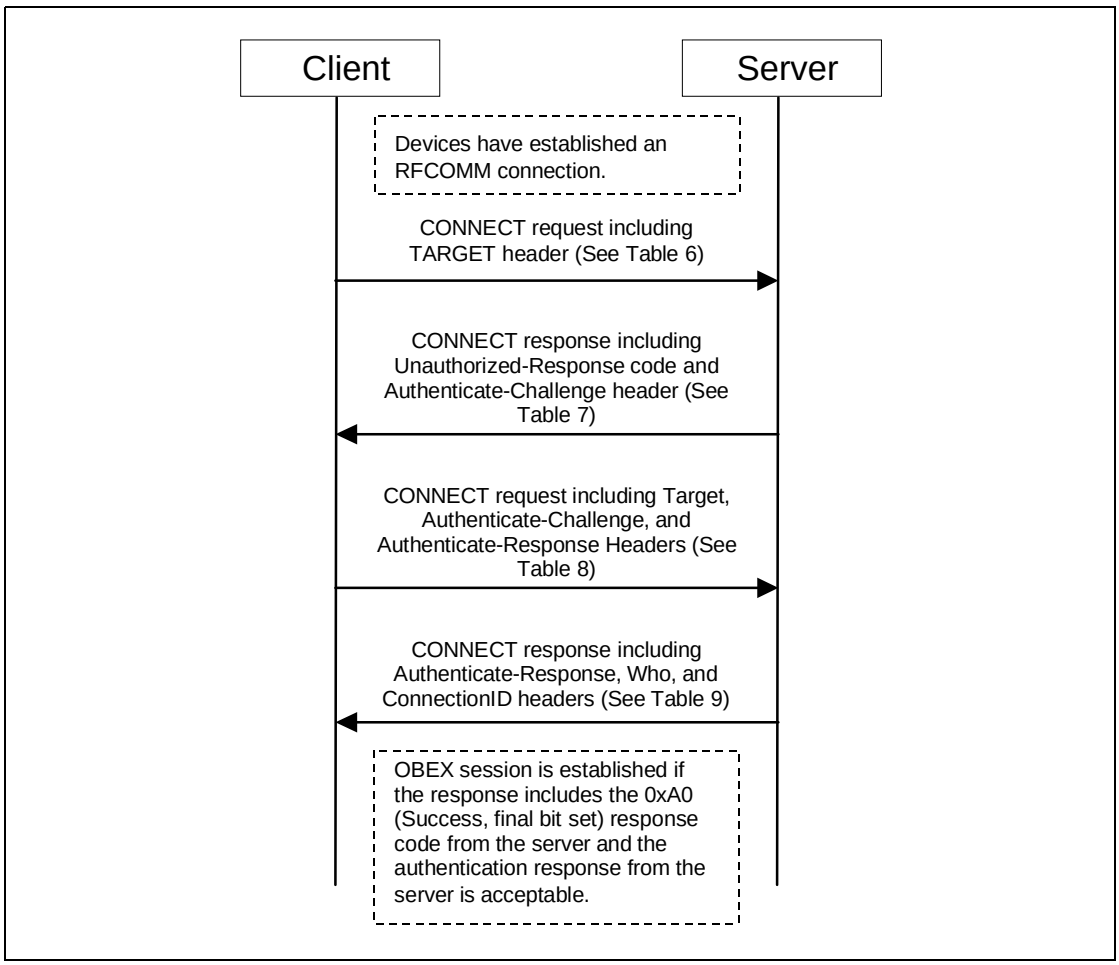


Figure 5.2: Establishment of OBEX Session with Authentication

The first CONNECT request indicates a need for connection and which service is used. The fields and the header in the CONNECT request are listed below:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for CONNECT	0x80	M	-
Field	Connect Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-

Table 5.5: Fields and Headers in CONNECT Request when Authentication Used

Field/ Header	Name	Value	Status	Explanation
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	Target	Varies	C1	Used to indicate the specific Service

Table 5.5: Fields and Headers in CONNECT Request when Authentication Used

C1: The usage ofthe Target header is dependent on the application profile utilizing GOEP. The example value for the Target header can be 'IRMC-SYNC' to indicate the IrMC synchronization service.

The first response to the CONNECT request from the Server, which authenticates the Client, includes the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for CONNECT request	Varies	M	0x41 for Unauthorized, because OBEX authentication is used.
Field	Connect Response Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	Authenticate Challenge	Varies	M	Carries the digest-challenge string.

Table 5.6: Fields and Headers in First CONNECT Response when Authenticating

The second CONNECT request has the following fields and headers in this order:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for CONNECT	0x80	M	-
Field	Connect Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-

Table 5.7: Fields and Headers in Second CONNECT Request when Authentication Used

Field/ Header	Name	Value	Status	Explanation
Header	Target	Varies	C1	-
Header	Authenticate Challenge	Varies	M	Carries the digest-challenge string.
Header	Authenticate Response	Varies	M	Carries the digest-response string. This is the response to the challenge from the Server.

Table 5.7: Fields and Headers in Second CONNECT Request when Authentication Used

C1: see Table 5.5

The second response to the CONNECT request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for CONNECT request	Varies	M	0xA0 for success
Field	Connect Response Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	ConnectionID	Varies	M	The header value specifies the current connection to the specific service.
Header	Who	Varies	M	The header value matches the Target header value.
Header	Authenticate Response	Varies	M	Carries the digest-response string. This is the response to the challenge from the Client.

Table 5.8: Fields and Headers in Second CONNECT Response when Authenticating

If the response code from the Server is successful, and the Client accepts the authentication response from the Server, the session is established and authenticated.

5.5 PUSHING DATA TO SERVER

The data object(s) is pushed to the Server using the PUT operation of the OBEX protocol. The data can be sent in one or more OBEX packets.
The PUT request must include the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for PUT	0x02 or 0x82	M	-
Field	Packet Length	Varies	M	-
Header	ConnectionID	Varies	C1	The header value specifies the current connection to the specific service.
Header	Name	Varies	M	The header value is the name of a single object, object store, or log information.
Header	Body/End of Body	Varies	M	End of Body identifies the last chunk of the object body.

Table 5.9: Fields and Headers in PUT Request

C1: The ConnectionID header is mandatory if the Target header is used when establishing the OBEX session.

Other headers, which can be optionally used, are specified in [11].

The response packet for the PUT request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for PUT	0x90 or 0xAO	M	0x90 for continue or 0xA0 for success
Field	Packet Length	Varies	M	-

Table 5.10: Fields and Headers in PUT Response

Other headers, which can be optionally used, are specified in [11].

5.6 PULLING DATA FROM SERVER

The data object(s) is pulled from the Server using the GET operation of the OBEX protocol. The data can be sent in one or more OBEX packets. The first GET request includes the following fields and headers.

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for GET	0x03	M	-
Field	Packet Length	Varies	M	-
Header	ConnectionID	Varies	C1	The header value specifies the current connection to the specific service.
Header	Type	Varies	C2	Indicates the type of the object to be pulled.
Header	Name	Varies	C2	The header value is the name of a single object, object store, or log information.

Table 5.11: Fields and Headers in GET Request

C1: The ConnectionID header is mandatory if the Target header is used when establishing the OBEX session.

C2: Either the Type header or the Name header must be included in the GET request when it is sent to the server.

Other headers, which can be optionally used, are specified in [11].

The response packet for the GET request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for Get	0x90 or 0xA0	M	0x90 or 0xA0 if the packet is the last the object
Field	Packet Length	Varies	M	-
Header	Name	Varies	O	The header value is the name of a single object, object store, or log information.
Header	Body/End of Body	Varies	M	End of Body identifies the last chunk of the object body.

Table 5.12: Fields and Headers in GET Response

Other headers, which can be optionally used, are specified in [11].

5.7 DISCONNECTION

see Chapter 2.2.2 in [1].

6 SERIAL PORT PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the protocol requirements of the Serial Port Profile (SeP) [12]. For the purposes of reading the SeP [12], the Server shall always be considered to be Device B and the Client shall always be considered to be Device A.

The following text, together with the associated sub-clauses, defines the requirements with regards to this profile – in addition to the requirements defined in [9].

6.1 RFCOMM INTEROPERABILITY REQUIREMENTS

For the RFCOMM layer, no additions to the requirements stated in Section 4 of Serial Port Profile apply.

6.2 L2CAP INTEROPERABILITY REQUIREMENTS

For the L2CAP layer, no additions to the requirements stated in Section 5 of Serial Port Profile apply.

6.3 SDP INTEROPERABILITY REQUIREMENTS

These requirements are defined by the application profiles. Thus, none of the requirements defined in the SeP profile (Section 6 in [12]) apply to this profile.

6.4 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

For the LM layer, no additions to the requirements stated in Section 7 of Serial Port Profile apply.

6.5 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

In the table below, LC capabilities differing from the capabilities required by the SeP profile (Section 8 in [12]) are listed.

	Capabilities	Support in baseband	Support in Server	Support in Client
5.	Packet types			
L	HV1 packet	M	X	X

Table 6.1: Baseband/LC capabilities

	Capabilities	Support in baseband	Support in Server	Support in Client
M	HV2 packet	O	X	X
N	HV3 packet	O	X	X
O	DV packet	M	X	X
7.	Voice codec			
A	A-law	O	X	X
B	μ-law	O	X	X
C	CVSD	O	X	X

Table 6.1: Baseband/LC capabilities

6.5.1 Inquiry and Inquiry Scan

For this profile, the Limited discoverable mode (see Section 7.1) should be used; but, if the Server device for some reason (e.g. lack of a sufficient user interface) wants to be visible at all times, the General discoverable mode (see Section 7.1) can be used instead. The client device must support the General inquiry procedure (see Section 7.3), and should also support the Limited inquiry procedure.

If the Limited inquiry procedure is supported, it should be used primarily. When this procedure is initiated in the Client, the client must perform this procedure for at least $T_{GAP}(100)$ (see Section 6.2.4 in GAP [14]). After the execution of the Limited inquiry procedure, the device may fall back to perform the General inquiry procedure. The device must support this fall-back functionality if the Limited inquiry procedure is supported. The fall-back procedure may or may not require user intervention. When general inquiry is initiated by the Client after limited inquiry, it shall be in this General limited procedure state for at least $T_{GAP}(100)$ (see Section 6.2.4 in GAP [14]).

For the inquiry, the returned CoD in the FHS packet must indicate that Object Transfer service is supported (see [13]). The major and minor device classes depend on the device supporting this profile. Therefore, usage of them is not defined in this profile.

The Limited Inquiry, Device Discovery and Name Discovery procedures are described in Section 6.2-6.4 in the Generic Access profile [14].

7 GENERIC ACCESS PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the Generic Access Profile. This section defines the support requirements with regards to procedures and capabilities defined in GAP.

7.1 MODES

Table 7.1 shows the support status for Modes within this profile.

	Procedure	Support in Client	Support in Server
1	Discoverability modes		
	Non-discoverable mode	N/A	M
	Limited discoverable mode	N/A	C1
	General discoverable mode	N/A	C1
2	Connectability modes		
	Non-connectable mode	N/A	O
	Connectable mode	N/A	M
3	Pairing modes		
	Non-pairable mode	N/A	M
	Pairable mode	N/A	M

Table 7.1: Modes

C1: The Limited discoverable mode should be used, but if the Server device for some reason (e.g. lack of a sufficient user interface) wants to be visible at all times, the General discoverable mode can be used instead.

7.2 SECURITY ASPECTS

Table 7.2 shows the support status for Security aspects within this profile.

	Procedure	Support in Client	Support in Server
1	Authentication	M	M
2	Security modes		

Table 7.2: Security aspects

	Procedure	Support in Client	Support in Server
	Security modes 1	M	M
	Security modes 2	C1	C1
	Security modes 3	C1	C1

Table 7.2: Security aspects

C1: Support for at least one of the security modes 2 and 3 is mandatory.

7.3 IDLE MODE PROCEDURES

Table 7.3 shows the support status for Idle mode procedures within this profile.

	Procedure	Support in Client	Support in Server
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	M	N/A
4	Device discovery	M	N/A
5	Bonding	M (Note 1)	M (Note 1)
Note 1: see section 7.3.1			

Table 7.3: Idle mode procedures

7.3.1 Bonding

It is mandatory for the Client and Server to support bonding. Bonding may be required before permitting communication between a Client and a Server. During bonding, the Client and Server are paired, which means that the Client and Server establish a security association (a common link key). This requires that an identical Bluetooth PIN code be entered on both the Client and Server devices.

The usage of bonding is optional for both Client and Server. The bonding procedures are defined in Section 6.5 in GAP [14].

8 REFERENCES

8.1 NORMATIVE REFERENCES

[1] Bluetooth Special Interest Group, IrDA Interoperability

[2] Bluetooth Special Interest Group, Synchronization Profile

[3] Bluetooth Special Interest Group, File Transfer Profile

[4] Bluetooth Special Interest Group, Object Push Profile

[5] Bluetooth Special Interest Group, Baseband Specification

[6] Bluetooth Special Interest Group, LMP Specification

[7] Bluetooth Special Interest Group, L2CAP Specification

[8] Bluetooth Special Interest Group, RFCOMM with TS 07.10", Specification of the Bluetooth System

[9] ETSI, TS 07.10, Version 6.3.0

[10] Bluetooth Special Interest Group, SDP Specification

[11] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX) with Published Errata, Version 1.2, April 1999

[12] Bluetooth Special Interest Group, Serial Port Profile

[13] Internet Assigned Numbers Authority, IANA Protocol/Number Assignments Directory (http://www.iana.org/numbers.html), May 1999.

[14] Bluetooth Special Interest Group, Generic Access Profile

Part K:11

OBJECT PUSH PROFILE

This application profile defines the application requirements for Bluetooth devices necessary for the support of the Object Push usage model. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Object Push usage model.

CONTENTS

1 Introduction 334
1.1 Scope 334
1.2 Bluetooth Profile Structure 334
1.3 Bluetooth OBEX-Related Specifications 335
1.4 Symbols and conventions 336
1.4.1 Requirement status symbols 336
1.4.2 Signaling diagram conventions 337

2 Profile overview 338
2.1 Profile stack 338
2.2 Configurations and roles 338
2.3 User requirements and scenarios 339
2.4 Profile fundamentals 339

3 User interface aspects 340
3.1 Mode selection, Push Servers 340
3.2 Function Selection, Push Clients 340
3.3 Application Usage Events 341
3.3.1 Object Push 341
3.3.2 Business Card Pull 341
3.3.3 Business Card Exchange 342

4 Application layer 344
4.1 Feature overview 344
4.2 Object Push Feature 344
4.2.1 Content Formats 344
4.2.2 Application Procedure 345
4.3 Business Card Pull Feature 345
4.3.1 Owner's Business Card 346
4.3.2 Application Procedure Business Card Pull 346
4.4 Business Card Exchange Feature 346
4.4.1 Owner's Business Card 346
4.4.2 Application Procedure Business Card Exchange 346

5 OBEX 348
5.1 OBEX Operations Used 348
5.2 OBEX Headers 348
5.2.1 OBEX Headers for the Object Push Feature 348
5.2.2 OBEX Headers for the Business Card Pull and Exchange Features 349
5.3 Initialization of OBEX 350
5.4 Establishment of OBEX session 350

5.5 Pushing Data 350
5.6 Pulling Data 350
5.7 Disconnection 350

6 Service Discovery 351
6.1 SD Service Records 351
6.2 SDP Protocol Data Units 352

7 References 353
7.1 Normative references 353

FOREWORD

This document, together with the Generic Object Exchange profile and the Generic Access profile, forms the Object Push usage model.

Interoperability between devices from different manufacturers is provided for a specific service and usage model if the devices conform to a Bluetooth SIG defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications and gives an unambiguous description of the air interface for specified service(s) and usage model(s).

All defined features are process-mandatory. This means that if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

The Object Push profile defines the requirements for the protocols and procedures that shall be used by the applications providing the Object Push usage model. This profile makes use of the Generic Object Exchange Profile (GOEP) [10] to define the interoperability requirements for the protocols needed by applications. The most common devices using these usage models can be notebook PCs, PDAs, and mobile phones.

The scenarios covered by this profile are the following:

- Usage of a Bluetooth device, e.g. a mobile phone to push an object to the inbox of another Bluetooth device. The object can for example be a business card or an appointment.
- Usage of a Bluetooth device; e.g. a mobile phone to pull a business card from another Bluetooth device.
- Usage of a Bluetooth device; e.g. a mobile phone to exchange business cards with another Bluetooth device. Exchange defined as a push of a business card followed by a pull of a business card.

1.2 BLUETOOTH PROFILE STRUCTURE

In Figure 1.1 Bluetooth Profiles, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly. For example, the Object Push profile is dependent on Generic Object Exchange, Serial Port, and Generic Access profiles.

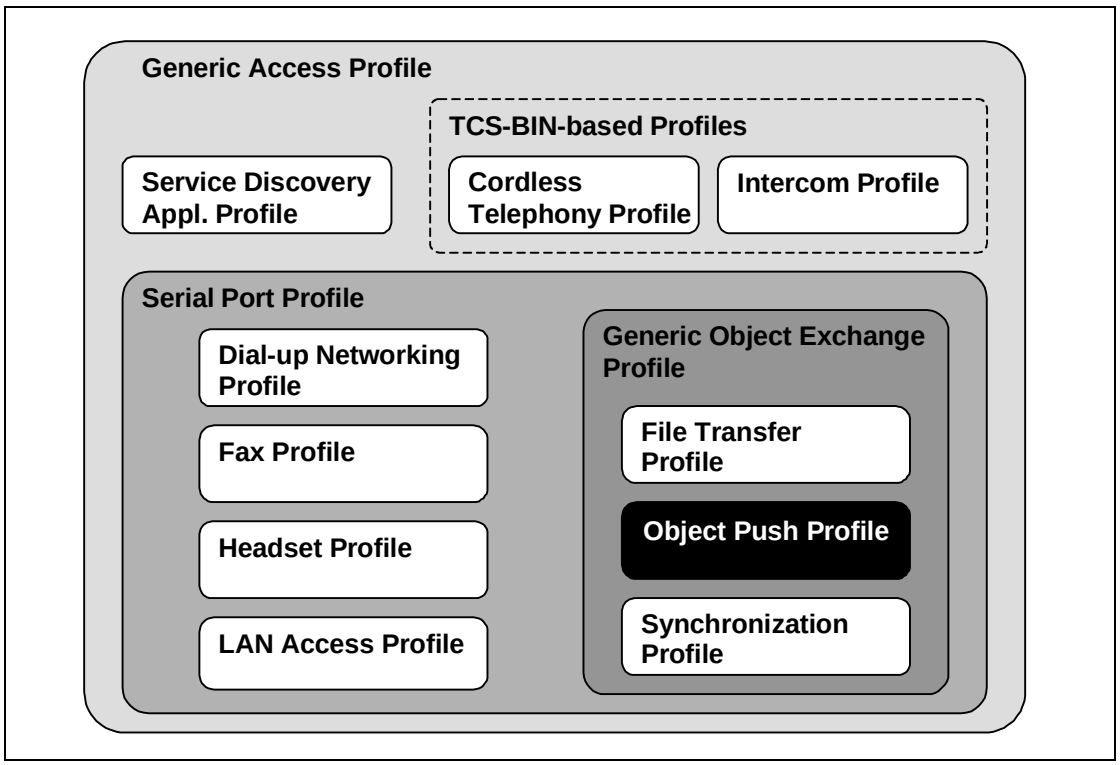


Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX-RELATED SPECIFICATIONS

Bluetooth Specification includes five separate specifications for OBEX and applications using OBEX.

1. Bluetooth IrDA Interoperability Specification [7].

- Defines how the applications can function over both Bluetooth and IrDA.
- Specifies how OBEX is mapped over RFCOMM and TCP.
- Defines the application profiles using OBEX over Bluetooth.

2. Bluetooth Generic Object Exchange Profile Specification [10]

- Generic interoperability specification for the application profiles using OBEX.
- Defines the interoperability requirements of the lower protocol layers (e.g. Baseband and LMP) for the application profiles.

3. Bluetooth Synchronization Profile Specification [15]

- Application Profile for Synchronization applications.
- Defines the interoperability requirements for the applications within the Synchronization application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

4. Bluetooth File Transfer Profile Specification [14]

- Application Profile for File Transfer applications.
- Defines the interoperability requirements for the applications within the File Transfer application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

5. **Bluetooth Object Push Profile Specification (this specification)**

- Application Profile for Object Push applications.
- Defines the interoperability requirements for the applications within the Object Push application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document, the following symbols are used:

'M' for mandatory to support (used for capabilities that shall be used in the profile);

'O' for optional to support (used for capabilities that can be used in the profile);

'C' for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

'X' for excluded (used for capabilities that may be supported by the unit but shall never be used in the profile);

'N/A' for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

1.4.2 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

A		B
	PROC1	
	PROC2	
	PROC3	
	(PROC4)	
	(PROC5)	
	MSG1	
	MSG2	
	(MSG3)	
	(MSG4)	

Table 1.1: Arrows used in signaling diagrams

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A and B). PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates an optional message from B to A.

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

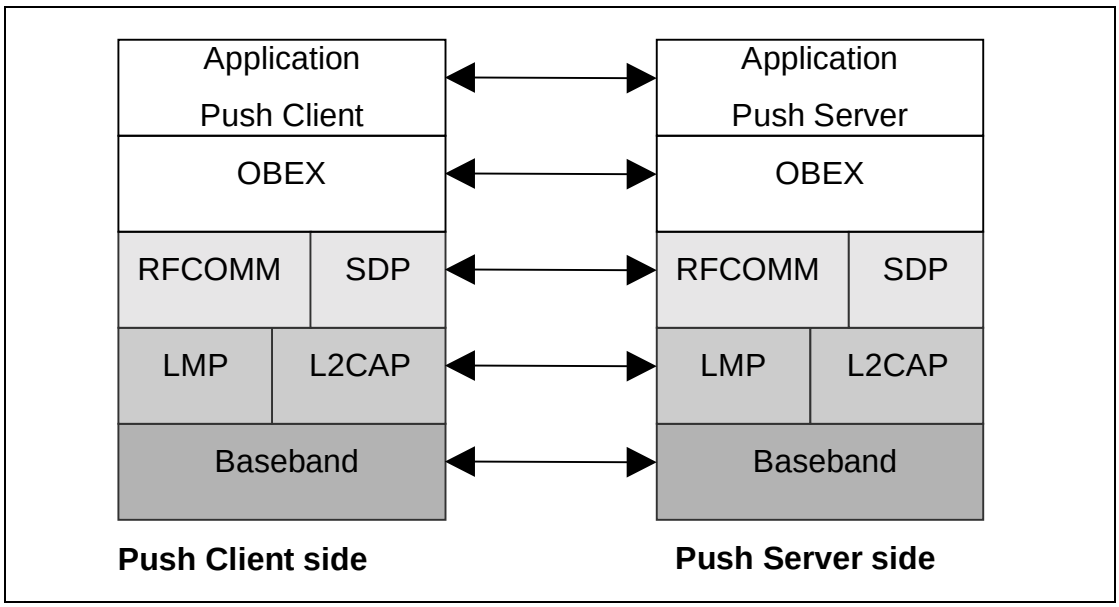


Figure 2.1: Protocol model

The Baseband [1], LMP [2] and L2CAP [3] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [4] is the Bluetooth adaptation of GSM TS 07.10 [5]. SDP is the Bluetooth Service Discovery Protocol [6]. OBEX [7] is the Bluetooth adaptation of IrOBEX [8].

The RFCOMM, L2CAP, LMP and Baseband interoperability requirements are defined in Section 6 in GOEP [10].

2.2 CONFIGURATIONS AND ROLES

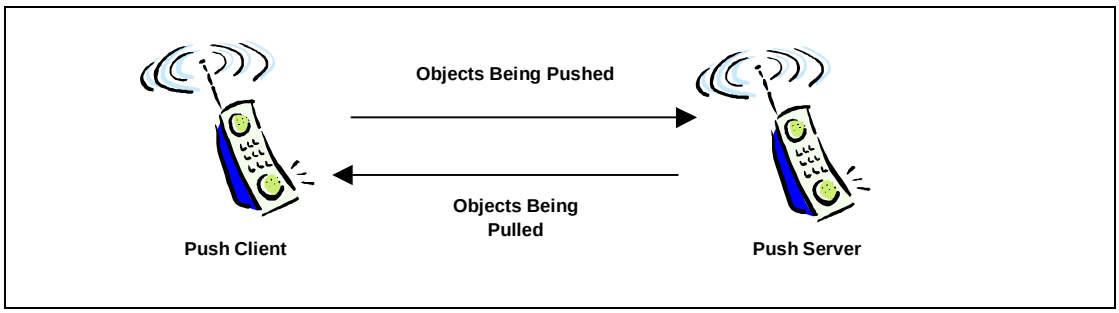


Figure 2.2: Push and Pull Example between two Mobile Phones

The following roles are defined for this profile:

Push Server – This is the server device that provides an object exchange server. In addition to the interoperability requirements defined in this profile, the Push Server must comply with the interoperability requirements for the server of the GOEP if not defined in the contrary.

Push Client – This is the client device that pushes and pulls objects to and from the Push Server. In addition to the interoperability requirements defined in this profile, the Push client must also comply with the interoperability requirements for the client of the GOEP, if not defined to the contrary.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are:

- Usage of a Push Client to push an object to a Push Server. The object can, for example, be a business card or an appointment.
- Usage of a Push Client to pull a business card from a Push Server.
- Usage of a Push Client to exchange business cards with a Push Server.

The restrictions applying to this profile are the same as in the GOEP.

The push operation described in this profile pushes objects from the Push Client to the inbox of the Push Server.

2.4 PROFILE FUNDAMENTALS

The profile fundamentals are the same as defined in the GOEP.

Link level authentication and encryption are mandatory to support and optional to use.

Bonding is mandatory to support and optional to use.

OBEX authentication is not used.

This profile does not mandate the server or client to enter any discoverable or connectable modes automatically, even if they are able to do so. On the Push Client side, end-user interaction is always needed to initiate the object push, business card pull or business card exchange.

3 USER INTERFACE ASPECTS

3.1 MODE SELECTION, PUSH SERVERS

Object Exchange mode affects the Push Server. It enables Push Clients to push and pull objects to and from the Push Server. The Push Clients can also try to pull objects from the Push Server in this mode. The Push Server does not have to support the pulling feature, but it must be able to respond with an appropriate error message.

When entering this mode, Push Servers should:
set the device in Limited Discoverable Mode (see Generic Access Profile),
must ensure that the Object Transfer bit is set in the CoD (see [16]),
and must ensure that a service record is registered in the SDDB (see Section 6).

Public devices, devices that want to be visible at all times, or devices that can not supply a user interface to enable Object Exchange mode shall use General Discoverable Mode (see [13]) instead of Limited Discoverable Mode.

It is recommended that this mode be set and unset by user interaction.

3.2 FUNCTION SELECTION, PUSH CLIENTS

- There are three different **functions** associated with the Object Push profile:
- Object Push function
- Business Card Pull function
- Business Card Exchange function

The **Object Push function** initiates the function that pushes one or more objects to a Push Server.

The **Business Card Pull function** initiates the function that pulls the business card from a Push Server.

The **Business Card Exchange function** initiates the function that exchanges business cards with a Push Server.

The three functions should be activated by the user. They should not be performed automatically without user interaction.

When the user selects one of these functions, an inquiry procedure will be performed to produce a list of available devices in the vicinity. Requirements on inquiry procedures are discussed in Section 6.5.1 of the GOEP [10].

3.3 APPLICATION USAGE EVENTS

In the following sections (3.3.1-3.3.3), the presented scenarios work as examples.Variations in the actual implementations are possible and allowed.

3.3.1 Object Push

When a Push Client wants to push an object to a Push Server, the following scenario can be followed. If authentication is used the user might have to enter a Bluetooth PIN at some point.

Push Client	Push Server
	The user sets the device **into Object Exchange mode**.
The user of the Push Client selects the **Object Push function** on the device.	
A list of Push Servers that may support the Object Push service is displayed to the user.	
The user selects a Push Server to push the object to. If the selected device does not support the Object Push service, the user is prompted to select another device.	
	When an object is received in the Push Server, it is recommended that the user of the Push Server be asked to accept or reject the object.
It is recommended that the user is notified of the result of the operation.	

3.3.2 Business Card Pull

When a Push Client wants to pull the business card from a Push Server the following user interaction can be followed.

If authentication is used, the user might have to enter a Bluetooth PIN at some point.

Push Client	Push Server
	The user sets the device into **Object Exchange mode**.
The user of the Push Client selects the **Business Card Pull function** on the device.	
A list of Push Servers that may support the Object Push service is displayed to the user.	
The user selects a Push Server to pull the business card from. If the selected device does not support the Object Push service, the user is prompted to select another device.	
	Some devices might ask the user whether or not to accept the request to pull the business card from his device.
It is recommended that the user is notified of the result of the operation.	

3.3.3 Business Card Exchange

When a Push Client wants to exchange business cards with a Push Server, the following user interaction can be followed.

If authentication is used, the user might have to enter a Bluetooth PIN at some point.

Push Client	Push Server
	The user sets the device into **Object Exchange mode**.
The user of the Push Client selects the **Business Card Exchange function** on the device.	
A list of Push Servers that may support the Object Push service is displayed to the user.	
The user selects a Push Server to exchange business cards with. If the selected device does not support the Object Push service, the user is prompted to select another device.	

Push Client	Push Server
	When a Push Client tries to exchange business cards with the Push Server , it is recommended that the user of the Push Server is asked to accept or reject the business card offered by the Push Client. Some devices might also ask the user whether or not to accept the request to pull the business card from his device.
It is recommended that the user is notified of the result of the operation.	

4 APPLICATION LAYER

This section describes the feature requirements on units active in the Object Push, Business Card Pull and Business Card Exchange use cases.

4.1 FEATURE OVERVIEW

Table 4.1 shows the features covered by the Object Push profile.

	Features	Support in Push Client	Support in Push Server
1.	Object Push	M*	M*.
2.	Business Card Pull	O	O†
3.	Business Card Exchange	O	O†.

Table 4.1: Application layer features

*. Support for vCard format is mandatory. Support for other formats is optional, although the server must be able to respond with an error code on a put request even if it doesn't support the requested format.

†. Optional, but the server must be able to respond with an error code on a pull request, even if it doesn't support this feature

4.2 OBJECT PUSH FEATURE

This feature lets a Push Client send one or more objects to a Push Server.

4.2.1 Content Formats

To achieve application level interoperability, content formats are defined for Object Push. For some applications content formats have been specified.

- Phone Book applications must support data exchange using the vCard 2.1 content format specified in [11]. The properties that are mandatory to support are listed in Chapter 7 of [9]. If a phone book application supports another content format it must still support the vCard 2.1 content format. If a device does not have a phone book application it does not have to support the vCard 2.1 content format.
- Calendar applications must support data exchange using the vCalendar 1.0 content format specified in [12]. The properties that are mandatory to support are listed in Chapter 8 of [9]. If a calendar application supports another content format it must still support the vCalendar 1.0 content format. If a device does not have a calendar application it does not have to support the vCalendar 1.0 content format.

- Messaging applications must support data exchange using the vMessage content format specified in Chapter 9 of [9]. If a messaging application supports another content format it must still support the vMessage content format as specified in Chapter 9 of [9]. If a device does not have a messaging application it does not have to support the vMessage content format.
- Notes applications must support data exchange using the vNote content format specified in Chapter 10 of [9]. If a notes application supports another content format it must still support the vNote content format as specified in Chapter 10 of [9]. If a device does not have a notes application it does not have to support the vNote content format.

It is highly recommended that a Push Client does not try to send objects of a format that the Push Server does not support. See Section 6 for information on how to find out which formats the Push Server supports.

The content formats supported by a Push Server must be identified in the SDDB.

4.2.2 Application Procedure

It is mandatory for Push Servers to be able to receive multiple objects within an OBEX connection. It is not mandatory for Push Clients to be able to send multiple objects during an OBEX connection. The Push Client uses one PUT operation for each object it wants to send. It is not mandatory to support sending or receiving of multiple objects within a single PUT operation.

Table 4.2 shows the application procedure required by the Push Client to push one or more objects to a Push Server.

Push Client	Details
OBEX CONNECT.	Target Header must not be used.
One or more OBEX PUTs for sending one or more objects.	
OBEX DISCONNECT	

Table 4.2: Application layer procedure for Object Push

For a detailed description of OBEX operations see Section 5.

4.3 BUSINESS CARD PULL FEATURE

A Push Client can optionally supply the functionality needed to pull a business card from a Push Server.

It is optional for the Push Server to support the business card pull feature. However, it must be able to respond to pull requests with an error message, see Section 5.6.

4.3.1 Owner's Business Card

Devices that support the business card pull and business card exchange services must store the owner's business card in the OBEX Default Get Object. Some devices (e.g. public devices) might hold information in the owner's business card that is relevant to the device rather than to the owner of the device.

The Default Get Object does not have a name; instead it is identified by its type. To achieve the ultimate application level interoperability, both the Push Client and the Push Server must support the vCard 2.1 content format specified in [11].

See [8] for a discussion on the Default Get Object.

4.3.2 Application Procedure Business Card Pull

Table 4.3 shows the application procedure required by the Push Client to perform a Business Card Pull from a Push Server.

Push Client	Details
OBEX CONNECT.	Target Header must not be used.
OBEX GET vCard of server's business card (default get object).	Type Header must be set to "text/x-vcard". Name Header must not be used.
OBEX DISCONNECT.	

Table 4.3: Application layer procedure for Business Card Pull

For a detailed description of OBEX operations see Section 5.

4.4 BUSINESS CARD EXCHANGE FEATURE

A Push Client can optionally supply the functionality needed to exchange business cards with a Push Server.

It is optional for the Push Server to support the business card exchange feature. It must, however, be able to respond to exchange requests with an error message, see Section 5.6.

4.4.1 Owner's Business Card

See Business Card Pull feature.

4.4.2 Application Procedure Business Card Exchange

Table 4.4 shows the application procedure required by the Push Client to perform a Business Card Exchange with a Push Server.

Push Client	Details
OBEX CONNECT.	Target Header must not be used.
OBEX PUT vCard with client's business card.	
OBEX GET vCard of server's business card (default get object).	Type Header must be set to "text/x-vcard". Name Header must not be used.
OBEX DISCONNECT.	

Table 4.4: Application layer procedure for Business Card Exchange

For a detailed description of OBEX operations see Section 5.

5 OBEX

5.1 OBEX OPERATIONS USED

Table 5.1 shows the OBEX operations, which are required in the Object Push profile.

Operation no.	OBEX Operation	Push Client	Push Server
1	Connect	M	M
2	Disconnect	M	M
3	Put	M	M
4	Get	O	M
5	Abort	M	M

Table 5.1: OBEX Operations

5.2 OBEX HEADERS

5.2.1 OBEX Headers for the Object Push Feature

Table 5.2 shows the specified OBEX headers which are required in the Object Push profile for the Object Push feature.

Header no.	OBEX Headers	Push Client	Push Server
1	Count	X	X
2	Name	M	M
3	Type	O	O
4	Length	M	M
5	Time	O	O
6	Description	O	O
7	Target	X	X
8	HTTP	O	O
9	Body	M	M
10	End of Body	M	M

Table 5.2: OBEX Headers used for the Object Push feature

Header no.	OBEX Headers	Push Client	Push Server
11	Who	X	X
12	Connection ID	X	X
13	Authenticate Challenge	X	X
14	Authenticate Response	X	X
15	Application Parameters	X	X
16	Object Class	X	X

Table 5.2: OBEX Headers used for the Object Push feature

5.2.2 OBEX Headers for the Business Card Pull and Exchange Features

Table 5.3 shows the specified OBEX headers which are required in the Object Push profile for the Business Card Pull and Exchange features.

Header no.	OBEX Headers	Push Client	Push Server
1	Count	X	X
2	Name	M	M
3	Type	M	M
4	Length	M	M
5	Time	O	O
6	Description	O	O
7	Target	X	X
8	HTTP	O	O
9	Body	M	M
10	End of Body	M	M
11	Who	X	X
12	Connection ID	X	X
13	Authenticate Challenge	X	X
14	Authenticate Response	X	X
15	Application Parameters	X	X
16	Object Class	X	X

Table 5.3: OBEX Headers used for the business card pull and business card exchange features

5.3 INITIALIZATION OF OBEX

Since OBEX authentication is not used by this profile, OBEX initialization is not applicable.

5.4 ESTABLISHMENT OF OBEX SESSION

See Section 5.4.1, in GOEP for a description of OBEX connection establishment without authentication.

The Push Client does not use the target header when establishing an OBEX connection.

5.5 PUSHING DATA

It is highly recommended that the Push Client use the Type Header when pushing objects to the Push Server.

See Section 5.5 in GOEP.

5.6 PULLING DATA

In the Object Push Profile, the Push Client only pulls data from the Push Server when it is getting the Default Get Object (owner's business card).

If there is no Default Get Object, the Push Server must respond with the error response code "NOT FOUND" [8]. The Push Client must be able to understand this error response code.

The Push Client must use the Type Header when getting the Default Get Object from the Push Server.

The Name Header is not used when getting the Default Get Object from the Push Server. If the Push Client sends a non-empty Name header, the Push Server should respond with the response code "FORBIDDEN"[8].

See Section 5.6 in GOEP .

5.7 DISCONNECTION

See Section 5.7 in GOEP.

6 SERVICE DISCOVERY

6.1 SD SERVICE RECORDS

The SD service record for the Object Push service is defined in Table 6.1.
A Push Client does not provide any SD service records.

Item	Definition	Type Size	Value*	AttrID	Status	Default Value
Service Class ID List				See [16]	M	
Service Class #0		UUID	OBEXObjectPush		M	
Protocol Descriptor List				See [16]	M	
Protocol ID #0		UUID	L2CAP		M	
Protocol ID #1		UUID	RFCOMM		M	
Param #0	Channel	Uint8	Varies		M	
Protocol ID #2		UUID	OBEX		M	
Service Name	Display-able Text name	String	Varies	See [16]	O	“OBEX Object Push”
BluetoothProfileDe-scriptorList				See [16]	O	
Profile ID #0	Supported profile	UUID	OBEXObjectPush			OBEX-Object-Push [16]
Version #0	Profile ver-sion	uint16	Varies			0x0100
Supported Formats List	Supported Formats List	Data Ele-ment Sequence of Uint8	Formats: **0x01** = vCard 2.1 **0x02** = vCard 3.0 **0x03** = vCal 1.0 **0x04** = iCal 2.0 **0x05** = vNote (as defined in [9]) **0x06** = vMessage (as defined in [9]) **0xFF** = any type of object.	See [16]	M	

Table 6.1: Object Push Service Record

*. Values that are of the type UUID are defined in the Assigned Numbers specification [16].

6.2 SDP PROTOCOL DATA UNITS

Table 6.2 shows the specified SDP PDUs (Protocol Data Units), which are required in the Object Push profile.

PDU no.	SDP PDU	Push Client	Push Server
1	SdpErrorResponse	M	M
2	SdpServiceSearchAttributeRequest	M	M
3	SdpServiceSearchAttributeResponse	M	M

Table 6.2: SDP PDUs

7 REFERENCES

7.1 NORMATIVE REFERENCES

[1] Bluetooth Special Interest Group, Baseband Specification

[2] Bluetooth Special Interest Group, LMP Specification

[3] Bluetooth Special Interest Group, L2CAP Specification

[4] Bluetooth Special Interest Group, RFCOMM with TS 07.10

[5] ETSI, TS 07.10, Version 6.3

[6] Bluetooth Special Interest Group, SDP Specification

[7] Bluetooth Special Interest Group, IrDA Interoperability

[8] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX), Version 1.2 with Published Errata, April 1999

[9] Infrared Data Association, IrMC (Ir Mobile Communications) Specification with Published Errata, Version 1.1, February 1999

[10] Bluetooth Special Interest Group, Generic Object Exchange Profile

[11] The Internet Mail Consortium, vCard – The Electronic Business Card Exchange Format, Version 2.1, September 1996

[12] The Internet Mail Consortium, vCalendar – The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996

[13] Bluetooth Special Interest Group, Generic Access Profile Specification

[14] Bluetooth Special Interest Group, File Transfer Profile Specification

[15] Bluetooth Special Interest Group, Synchronization Profile Specification

[16] Bluetooth Special Interest Group, Assigned Numbers specification http://www.bluetooth.org/assigned-numbers.htm

Part K:12

FILE TRANSFER PROFILE

This application profile defines the application requirements for Bluetooth devices necessary for the support of the File Transfer usage model. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the File Transfer usage model.

CONTENTS

1 Introduction 360
1.1 Scope 360
1.2 Bluetooth profile structure 360
1.3 Bluetooth OBEX-Related Specifications 361
1.4 Symbols and conventions 362
1.4.1 Requirement status symbols 362
1.4.2 Signaling diagram conventions 363

2 Profile overview 364
2.1 Profile stack 364
2.2 Configurations and roles 364
2.3 User requirements and scenarios 365
2.4 Profile fundamentals 365

3 User interface aspects 367
3.1 File Transfer Mode selection, Servers 367
3.2 Function Selection, Clients 367
3.3 Application usage 368

4 Application layer 370
4.1 Feature overview 370
4.2 Folder Browsing 370
4.3 Object Transfer 372
4.4 Object Manipulation 373

5 OBEX 374
5.1 OBEX Operations Used 374
5.2 OBEX Headers 375
5.3 Initialization of OBEX 375
5.4 Establishment of OBEX session 375
5.5 Browsing Folders 376
5.5.1 Pulling a Folder Listing Object 376
5.5.2 Setting the Current Folder (Forward) 376
5.5.3 Setting the Current Folder (Backward) 377
5.5.4 Setting the Current Folder (Root) 378
5.6 Pushing Objects 379
5.6.1 Pushing Files 379
5.6.2 Pushing Folders 379
5.6.2.1 Creating New Folders 380
5.7 Pulling Objects 381
5.7.1 Pulling Files 381
5.7.2 Pulling Folders 381

5.8 Manipulating Objects 381
5.8.1 Deleting Files 381
5.8.2 Deleting Folders 382
5.9 Disconnection 382
6 Service Discovery 383
6.1 SD service records 383
6.2 SDP protocol data units 384
7 References 385
7.1 Normative references 385

FOREWORD

This document, together with the Generic Object Exchange profile and the Generic Access profile form the File Transfer usage model.

Interoperability between devices from different manufacturers is provided for a specific service and usage model if the devices conform to a Bluetooth SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications, and gives an unambiguous description of the air interface for specified service(s) and usage model(s).

All defined features are process-mandatory. This means that if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

The File Transfer profile defines the requirements for the protocols and procedures that shall be used by the applications providing the File Transfer usage model. This profile uses the Generic Object Exchange profile (GOEP) as a base profile to define the interoperability requirements for the protocols needed by the applications. The most common devices using these usage models can be (but are not limited to) PCs, notebooks, and PDAs.

The scenarios covered by this profile are the following:

- Usage of a Bluetooth device (e.g. a notebook PC) to browse an object store (file system) of another Bluetooth device. Browsing involves viewing objects (files and folders) and navigating the folder hierarchy of another Bluetooth device. For example, one PC browsing the file system of another PC.
- A second usage is to transfer objects (files and folders) between two Bluetooth devices. For example, copying files from one PC to another PC.
- A third usage is for a Bluetooth device to manipulate objects (files and folders) on another Bluetooth device. This includes deleting objects, and creating new folders.

1.2 BLUETOOTH PROFILE STRUCTURE

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

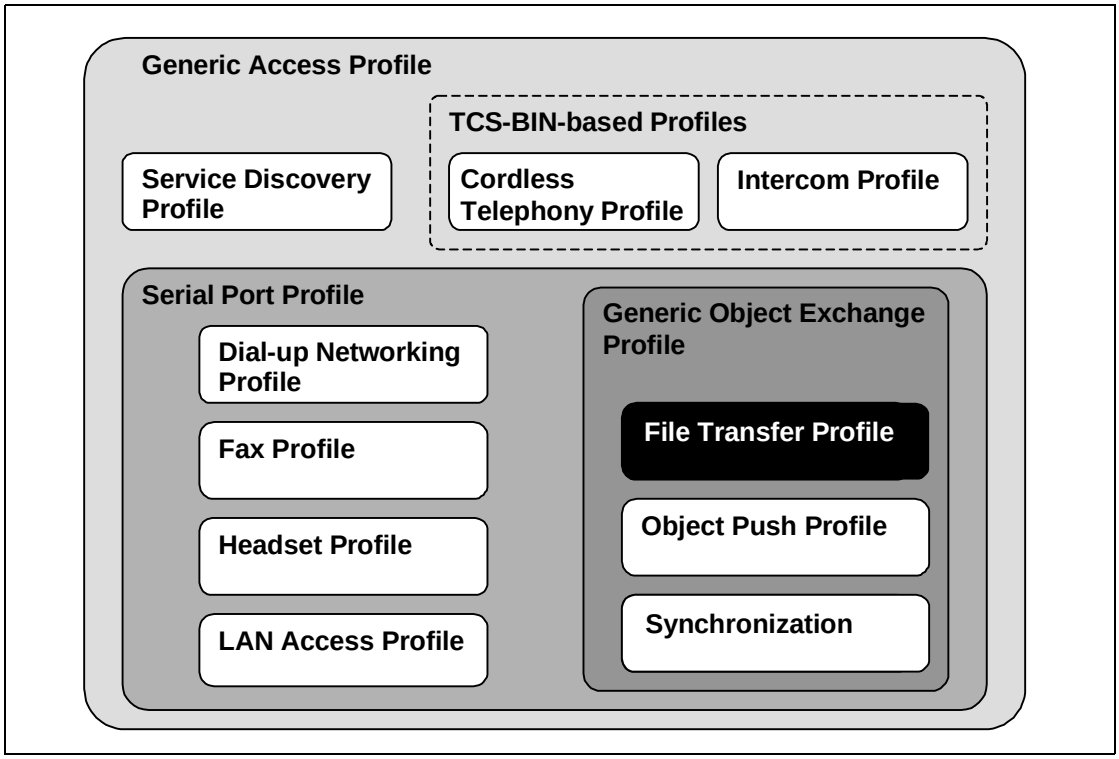


Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX-RELATED SPECIFICATIONS

Bluetooth Specification includes five separate specifications for OBEX and applications using OBEX.

1. Bluetooth IrDA Interoperability Specification [1].

- Defines how the applications can function over both Bluetooth and IrDA.
- Specifies how OBEX is mapped over RFCOMM and TCP.
- Defines the application profiles using OBEX over Bluetooth.

2. Bluetooth Generic Object Exchange Profile Specification [2]

- Generic interoperability specification for the application profiles using OBEX.
- Defines the interoperability requirements of the lower protocol layers (e.g. Baseband and LMP) for the application profiles.

3. Bluetooth Synchronization Profile Specification [3]

- Application Profile for Synchronization applications.
- Defines the interoperability requirements for the applications within the Synchronization application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

4. Bluetooth File Transfer Profile Specification (This Specification)

- Application Profile for File Transfer applications.
- Defines the interoperability requirements for the applications within the File Transfer application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

5. Bluetooth Object Push Profile Specification [4]

- Application Profile for Object Push applications.
- Defines the interoperability requirements for the applications within the Object Push application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document (especially in the profile requirements tables in Annex A), the following symbols are used:

'M' for mandatory to support (used for capabilities that shall be used in the profile);

'O' for optional to support (used for capabilities that can be used in the profile);

'C' for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

'X' for excluded (used for capabilities that may be supported by the unit but shall never be used in the profile);

'N/A' for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

1.4.2 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

A		B
	PROC1	
	PROC2	
	PROC3	
	(PROC4)	
	(PROC5)	
	MSG1	
	MSG2	
	(MSG3)	
	(MSG4)	

Table 1.1: Arrows used in signaling diagrams

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A and B). PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates an optional message from B to A.

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

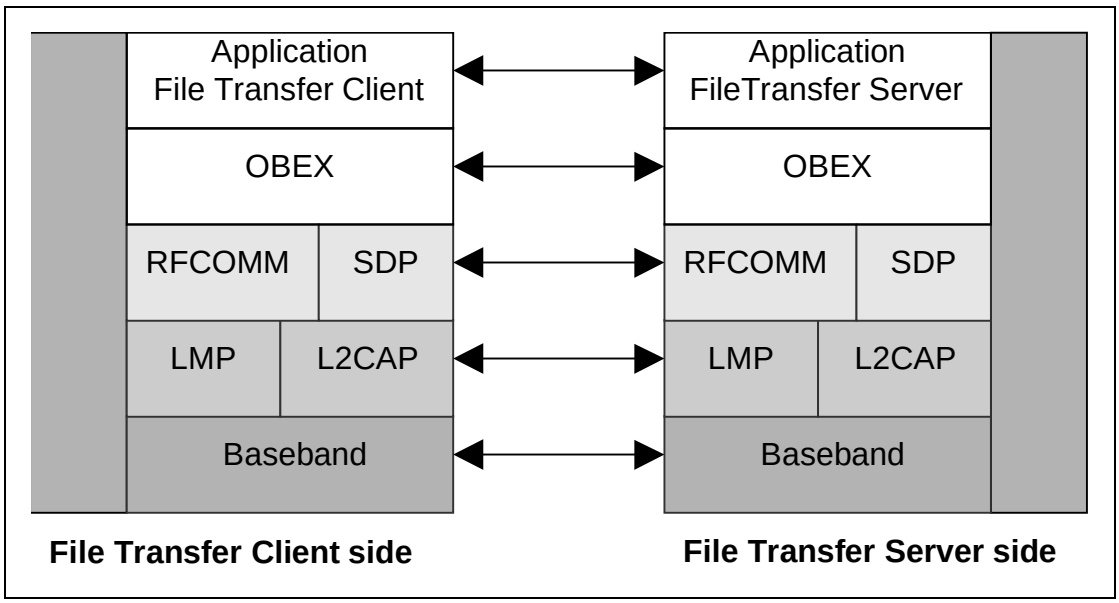


Figure 2.1: Protocol model

The Baseband [5], LMP [6] and L2CAP [7] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [8] is the Bluetooth adaptation of GSM TS 07.10 [9]. SDP is the Bluetooth Service Discovery Protocol [10]. OBEX [1] is the Bluetooth adaptation of IrOBEX [11].

The RFCOMM, L2CAP, LMP, and Baseband interoperability requirements are defined in GOEP.

2.2 CONFIGURATIONS AND ROLES

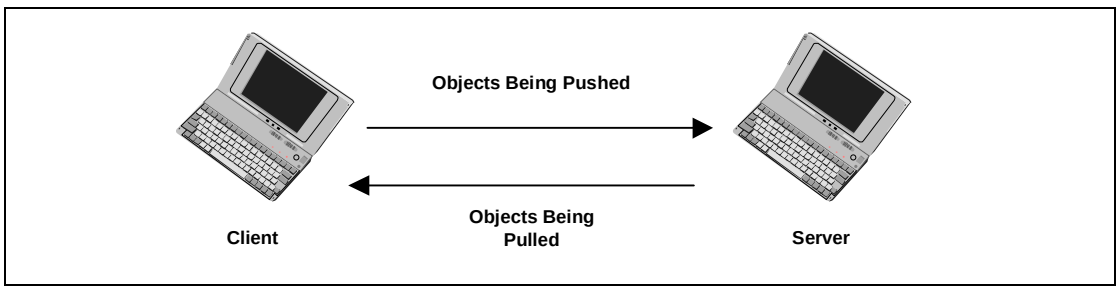


Figure 2.2: Bi-directional File Transfer Example between two Personal Computers

The following roles are defined for this profile:

Client – The Client device initiates the operation, which pushes and pulls objects to and from the *Server*. In addition to the interoperability requirements defined in this profile, the Client must also comply with the interoperability requirements for the Client of the GOEP if not defined in the contrary. The Client must be able to interpret the OBEX Folder Listing format and may display this information for the user.

Server – The Server device is the target remote Bluetooth device that provides an object exchange server and folder browsing capability using the OBEX Folder Listing format. In addition to the interoperability requirements defined in this profile, the Server must comply with the interoperability requirements for the Server of the GOEP if not defined in the contrary.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- If file browsing is supported on the client and the server, the client is able to browse the object store of the Server. Clients are required to pull and understand Folder Listing Objects. Servers are required to respond to requests for Folder Listing Objects. Servers should have a root folder. The root folder is typically not the root directory of the file system, but a designated public folder. A server may expose different root folders based on the user or device initiating the OBEX connection. Servers are not required expose a folder hierarchy.
- Usage of the Client to transfer objects to and from the Server. The transfer of objects includes folders and files. Clients must support the ability to push or pull files from the Server. Clients are not required to push or pull folders. Servers are required to support file push, pull, or both. Servers are allowed to have read-only folders and files, which means they can restrict object pushes. Thus, Servers are not required to support folder push or pull.
- Usage of the Client to create folders and delete objects (folders and files) on the Server. Clients are not required to support folder/file deletion or folder creation. Servers are allowed to support read-only folders and files, which means they can restrict folder/file deletion and creation.

A device adhering to this profile must support Client capability, Server capability or both. The restrictions applying to this profile are the same as in the GOEP.

2.4 PROFILE FUNDAMENTALS

The profile fundamentals are the same as defined in Section 2.4 in GOEP [2]. Support for link level authentication and encryption is required but their use is optional.

Support for OBEX authentication is required but its use is optional.

This profile does not mandate the server or client to enter any discoverable or connectable modes automatically, even if they are able to do so.

On the Client side, end-user intervention is typically needed to initiate file transfer (see Chapter 3).

Support of bonding is required but its use is optional.

3 USER INTERFACE ASPECTS

3.1 FILE TRANSFER MODE SELECTION, SERVERS

Servers must be placed in File Transfer mode. This mode enables a Client to initiate file transfer operations with the Server. When entering this mode, File Transfer Servers should set the device in *Limited Discoverable* mode (see Generic Access Profile), ensure that the Object Transfer Bit is set in the CoD (see [15]), and register a service record in the SDDB (see section 6 on page 394).

It is recommended that this mode be set and unset by user interaction, when possible. Public devices, devices that want to be visible at all times, or devices that can not supply a user interface to enable File Transfer mode shall use *General Discoverable* mode (see Generic Access Profile) instead of *Limited Discoverable* mode.

3.2 FUNCTION SELECTION, CLIENTS

Clients provide file transfer functions to the user via a user interface. An example of a file transfer user interface is a file-tree viewer to browse folders and files. Using such a system file-tree viewer, the user can browse and manipulate files on another PC, which appears in the network view.

File Transfer Applications may provide the following functions:

Select Server	Selecting the Server from a list of possible Servers, and setting up a connection to it.
Navigate Folders	Displaying the Server's folder hierarchy, including the files in the folders, and moving through the Server's folder hierarchy to select the current folder. The current folder is where items are pulled and/or pushed.
Pull Object	Copying a file or a folder from the Server to the Client.
Push Object	Copying a file or folder from the Client to the Server.
Delete Object	Deleting a file or folder on the Server.
Create Folder	Creating a new folder on the Server.

When the user selects the Select Server function, an inquiry procedure will be performed to produce a list of available devices in the vicinity. Requirements on inquiry procedures are discussed in Section 6.5.1 of the GOEP [2].

3.3 APPLICATION USAGE

In this section, the presented scenarios work as examples. Variations in the actual implementations are possible and allowed.

When the Client wants to select a Server the following user interaction can be followed:

Client	Server
	The user sets the device **into File Transfer mode**. A Server typically does not need to provide any other user interaction.
The user of the Client selects the **File Transfer Application** on the device.	
A list of Servers that may support the File Transfer service is displayed to the user.	
The user selects a Server in which to connect. The connection may require the user to enter a password for authentication. If both link level authentication and OBEX authentication is required, then the user will need to be prompted for two passwords.	If the Client requires authentication of the Server, then the Server will need to prompt the user for a password. If both link level authentication and OBEX authentication are required, then the user will need to be prompted for two passwords.
After the connection is complete, including any authentication, the contents of the Server's root folder are displayed.	

The following user interaction shows how the user of the Client performs file transfer functions. The operations assume a Server has already been selected as described above.

Client	Server
The user is presented with the folder hierarchy of the Server. The first presentation has the root folder selected as the current folder.	
The user chooses a folder to be the current folder. The contents of this folder are displayed.	
To push a file from the Client to the Server, the user selects a file on the Client and activates the **Push Object** function. The object is transferred to the current folder on the Server.	
To pull a file from the Server, the user selects a file in the current folder of the Server and activates the **Pull Object** function. The user is notified of the result of the operation.	
To delete a file on the Server, the user selects the file in the Server's current folder and activates the **Delete Object** function. The user is notified of the result of the operation.	
To create a new folder on the Server, the user activates the **Create Folder** function. This function requests a name from the user for the folder. When complete, a new folder is created in the Server's current folder.	

4 APPLICATION LAYER

This section describes the feature requirements on units active in the File Transfer use case.

4.1 FEATURE OVERVIEW

The File Transfer application is divided into three main features, as shown in the Table 4.1 below.

	Features	Support in File Transfer Client	Support in File Transfer Server
1.	Folder Browsing	M	M
2.	Object Transfer: File Transfer Folder Transfer	 M O	 M O*
3.	Object Manipulation	O	O*

Table 4.1: Application layer procedures

*. Optional, but the server must be able to respond with an appropriate error code, even if it doesn't support these capabilities.

4.2 FOLDER BROWSING

A folder browsing session may begin with the Client connecting to the Server and pulling the contents of the Server's root folder. When an OBEX connection is made, the Server starts out with its current folder set to the root folder. The Server may choose to expose different root folders to different users and/or devices. The Server has the right to refuse to disclose the contents of the root folder by replying to the GET folder object request with an Unauthorized or Forbidden response. If allowed, the contents of the folder must be transferred in the Folder Listing format specified in [11].

Table 4.2 shows the application procedure required by the Client to connect to the Server and pull the contents of the root folder.

Client	Details
OBEX CONNECT.	Target Header must be set to the Folder Browsing UUID: F9EC7BC4-953C-11D2-984E-525400DC9E09. This UUID is sent in binary (16 bytes) with most significant byte sent first (0xF9 is sent first).
Pull the contents of the Server's root folder using GET.	The Type Header must be set to the MIME-type of the Folder Listing Object (x-obex/folder-listing). The Connect ID header must be set to the value returned in the Connect operation. A Name header is not used.

Table 4.2: Application layer procedure for File Transfer Connect

Browsing an object store involves displaying folder contents and setting the 'current folder'. The OBEX SETPATH command is used to set the current folder. A Server must support the SETPATH command to the root folder (default directory). To display a folder hierarchy starting with the root folder, the Client must read the root folder contents using GET. It must then retrieve the contents of all sub-folders using GET. If the sub-folders contain folders, then the Client must retrieve the contents of these folders and so on. To retrieve the contents of a folder, the Client must set the current folder to the sub-folder using SETPATH, then pull the sub-folder contents using GET. Table 4.3 shows the application procedure required for retrieving the contents of a sub-folder.

Client	Details
Set the current folder to the sub-folder using OBEX SETPATH.	Name header is set to the name of the sub-folder. Connect ID header is required.
Pull the contents of the sub-folder using GET.	No Name is sent, since the sub-folder is the current folder. The Type Header must be set to the MIME-type of the Folder Listing Object (x-obex/folder-listing). Connect ID header is required.
Set the current folder back to the root folder using OBEX SETPATH.	Name header is empty. Connect ID header is required.
If the parent of the sub-folder is not the root folder, then set the current folder to the parent folder using SETPATH.	The Backup flag is set and no Name header is sent. Connect ID header is required.

Table 4.3: Application layer procedure for Folder Browsing

4.3 OBJECT TRANSFER

Objects are transferred from the Client to the Server using OBEX PUT, and objects are transferred from the Server to the Client using OBEX GET. Transferring files requires a single PUT or GET operation per file. Successful transfer of a file does not necessarily imply that file can be immediately retrieved due to the protection policies enforced by the Server. Transferring folders requires transferring all the items stored in a folder, including other folders. The process of transferring a folder may require that new folders be created. The SETPATH command is used to create folders.

Table 4.4 shows the application procedure for transferring a folder from the Client to the Server. If the folder contains other folders, then these other folders are transferred using the same method. The folder is transferred to the current folder on the Server.

Client	Details
Create a new folder (if it does not already exist) in the Server's current folder using SETPATH. The current folder is changed to this new folder.	Name header is set to the name of the new folder. Connect ID header is required.
Push all files to the new folder using a PUT command for each file.	The Name header is set to the name of the file. Connect ID header is required.
Folders are created using SETPATH.	Name header is set to folder name. This application procedure is applied recursively to each folder. Connect ID header is required.
Set the current folder back to the parent folder using SETPATH.	The Backup flag is set and no Name header is sent. Connect ID header is required.

Table 4.4: Application layer procedure for Pushing a Folder

Table 4.5 shows the application procedure for transferring a folder from the Server to the Client.

Client	Details
Set the current folder to the folder which is to be transferred using SETPATH.	The Name header is set to the name of the folder. Connect ID header is required.
Pull the contents of the folder using GET.	A Name header is not sent, and the Type Header must be set to the MIME-type of the Folder Listing Object (x-obex/folder-listing).
Pull all files to the new folder using a GET command for each file.	The Name header is set to the name of the file. Connect ID header is required.
Pull all Folders to the new folder using this application procedure.	This application procedure is applied recursively to each folder.

Table 4.5: Application layer procedure for Pulling a Folder

Set the current folder back to the parent folder, using SETPATH.	The Backup flag is set and no Name header is sent. Connect ID header is required.

Table 4.5: Application layer procedure for Pulling a Folder

4.4 OBJECT MANIPULATION

A Client can create and delete folders and files on a Server for which it has proper access privileges. A brief summary of these functions is shown below.

- A file is deleted by using a PUT command with the name of the file in a Name header and no Body header.
- An empty folder is deleted by using a PUT command with the name of the folder in a Name header and no Body header.
- A non-empty folder can be deleted in the same way as an empty folder but Servers may not allow this operation. If a Server refuses to delete a non-empty folder it must return the “Precondition Failed” (0xCC) response code. This response code tells the Client that it must first delete all the elements of the folder individually before deleting the folder.
- A new folder is created in the Server’s current folder by using the SETPATH command with the name of the folder in a Name header. If a folder with that name already exists, then a new folder is not created. In both cases the current folder is set to the new folder.

5 OBEX

5.1 OBEX OPERATIONS USED

Table 5.1 shows the OBEX operations that are used in the File Transfer profile.

Operation no.	OBEX Operation	Client	Server
1	Connect	M	M
2	Disconnect	M	M
3	Put	M	M
4	Get	M	M
5	Abort	M	M
6	SetPath	M	M

Table 5.1: OBEX Operations

5.2 OBEX HEADERS

Table 5.2 shows the specified OBEX headers that are used in the File Transfer profile.

Header no.	OBEX Headers	Client	Server
1	Count	O	O
2	Name	M	M
3	Type	M	M
4	Length	M	M
5	Time	O	O
6	Description	O	O
7	Target	M	M
8	HTTP	O	O
9	Body	M	M
10	End of Body	M	M
11	Who	M	M
12	Connection ID	M	M
13	Authenticate Challenge	M	M
14	Authenticate Response	M	M
15	Application Parameters	X	X
16	Object Class	X	X

Table 5.2: OBEX Headers

5.3 INITIALIZATION OF OBEX

Devices implementing the File Transfer profile can optionally use OBEX authentication. The initialization procedure is defined in Section 5.3 of GOEP [2].

5.4 ESTABLISHMENT OF OBEX SESSION

The OBEX connection must use a Target header set to the File Browsing UUID, F9EC7BC4-953C-11D2-984E-525400DC9E09. This UUID is sent in binary (16 bytes) with 0xF9 sent first. OBEX authentication can optionally be used. This profile follows the procedures described in Section 5.4 of GOEP [2] with the Target, Connection ID, and Who headers being mandatory.

5.5 BROWSING FOLDERS

Browsing folders involves pulling Folder Listing objects and setting the current folder. Navigating a folder hierarchy requires moving forward and backward by changing the current folder. Upon completion of the OBEX Connect operation the Server's current folder is the root folder. As noted previously, different root folders may be exported based on the Client device and/or user.

5.5.1 Pulling a Folder Listing Object

Pulling a Folder Listing object uses a GET operation and follows the procedure described in Section 5.6 of GOEP [2]. The Connection ID and Type headers are mandatory. A Name header containing the name of the folder is used to pull the listing of a folder. Sending the GET command without a name header is used to pull the contents of the current folder. Typically, a folder browsing application will pull the contents of the current folder, so a Name header is not used. The Type header must be set to 'x-obex/folder-listing'.

5.5.2 Setting the Current Folder (Forward)

Setting the current folder requires the SETPATH operation. The SETPATH request must include the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for SETPATH	0x85	M	-
Field	Packet Length	Varies	M	-
Field	Flags	0x02	M	'Backup level' flag is set to 0 and 'Don't Create' flag is set to 1.
Field	Constants	0x00	M	Constants are not used, and the field must be set to 0.
Header	Connection ID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.
Header	Name	Varies	M	Name of the folder.

Table 5.3: Fields and Headers in SETPATH Request for Setting Current Folder (Forward)

The response packet for the SETPATH request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for SETPATH	0xA0 or 0xC4	M	0xA0 for success or 0xC4 if the folder does not exist or 0xC1 if folder browsing is not permitted.
Field	Packet Length	Varies	M	-

Table 5.4: Fields and Headers in SETPATH Response for Setting Current Folder (Forward)

Other headers such as Description can optionally be used.

5.5.3 Setting the Current Folder (Backward)

Setting the current folder back to the parent folder requires the SETPATH operation. The SETPATH request must include the following fields and headers (note that a Name header is not used):

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for SETPATH	0x85	M	-
Field	Packet Length	Varies	M	-
Field	Flags	0x03	M	'Backup level' flag is set to 1 and 'Don't Create' flag is set to 1.
Field	Constants	0x00	M	Constants are not used, and the field must be set to 0.
Header	Connection ID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.

Table 5.5: Fields and Headers in SETPATH Request for Setting Current Folder (Backward)

The response packet for the SETPATH request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for SETPATH	0xA0 or 0xC4	M	0xA0 for success, or 0xC4 if the current folder is the root.
Field	Packet Length	Varies	M	-

Table 5.6: Fields and Headers in SETPATH Response for Setting Current Folder (Backward)

Other headers, such as Description, can optionally be used.

5.5.4 Setting the Current Folder (Root)

Setting the current folder to the root requires the SETPATH operation. The SETPATH request must include the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for SET-PATH	0x85	M	-
Field	Packet Length	Varies	M	-
Field	Flags	0x02	M	'Backup level' flag is set to 0 and 'Don't Create' flag is set to 1.
Field	Constants	0x00	M	Constants are not used, and the field must be set to 0.
Header	Connection ID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.
Header	Name	Empty	M	Name header is empty.

Table 5.7: Fields and Headers in SETPATH Request for Setting Current Folder (Root)

The response packet for the SETPATH request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for SETPATH	0xA0	M	0xA0 for success.
Field	Packet Length	Varies	M	-

Table 5.8: Fields and Headers in SETPATH Response for Setting Current Folder (Root)

Other headers, such as Description, can optionally be used.

5.6 PUSHING OBJECTS

Pushing object involves pushing files and folders.

5.6.1 Pushing Files

Pushing files follows the procedure described in Section 5.5 of GOEP [2]. The Connection ID header is mandatory.

5.6.2 Pushing Folders

Pushing folders involves creating new folders and pushing files. It may also involve navigating through the folder hierarchy. Navigation is described in Section 5.5 on page 387. Pushing files is described in Section 5.6.1 on page 390.

5.6.2.1 Creating New Folders

Creating a new folder requires the SETPATH operation. The SETPATH request must include the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for SET-PATH	0x85	M	-
Field	Packet Length	Varies	M	-
Field	Flags	0x00	M	'Backup level' flag is set to 0 and 'Don't Create' flag is set to 0.
Field	Constants	0x00	M	Constants are not used, and the field must be set to 0.
Header	Connection ID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.
Header	Name	Varies	M	Name of the folder.

Table 5.9: Fields and Headers in SETPATH Request for Creating a Folder.

The response packet for the SETPATH request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for SETPATH	0xA0 or 0xC1	M	0xA0 for success or 0xC1 if the current folder is read only and creation of a new folder is unauthorized.
Field	Packet Length	Varies	M	-

Table 5.10: Fields and Headers in SETPATH Response for Creating a Folder

Other headers such as Description can optionally be used.

5.7 PULLING OBJECTS

Pulling objects involves pulling files and folders.

5.7.1 Pulling Files

Pulling files follows the procedure described in Section 5.6 of GOEP [2]. The Connect ID header is mandatory.

5.7.2 Pulling Folders

Pulling folders involves navigating the folder hierarchy, pulling folder listing objects and pulling files. Navigating the folder hierarchy and pulling folder listing-objects is described in Section 5.5 on page 387. Pulling files is described in Section 5.7.1 on page 392.

5.8 MANIPULATING OBJECTS

Manipulating objects includes deleting objects and creating new folders. Creating new folders is described in Section 5.6.2.1 on page 391, Creating New Folders. Deleting objects involves deleting files and folders.

5.8.1 Deleting Files

Deleting a file requires the PUT operation. The PUT request must include the following fields and headers (note that no Body or End Body headers are sent):

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for PUT	0x82	M	-
Field	Packet Length	Varies	M	-
Header	ConnectionID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.
Header	Name	Varies	M	The header value is the name of the object to delete.

Table 5.11: Fields and Headers in PUT Request for Delete

The response packet for the PUT request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for PUT	0xA0, 0xC1 or 0xC4	M	0xA0 for success, 0xC1 for unauthorized (e.g. read only) or 0xC4 if the file does not exist.
Field	Packet Length	Varies	M	-

Table 5.12: Fields and Headers in PUT Response for Delete

Other headers such as Description can optionally be used.

5.8.2 Deleting Folders

A folder can be deleted using the same procedure used to delete a file (see Section 5.8.1 on page 392). Deleting a non-empty folder will delete all its contents, including other folders. Some Servers may not allow this operation and will return the “Precondition Failed” (0xCC) response code, indicating that the folder is not empty. In this case the Client will need to delete the contents before deleting the folder.

5.9 DISCONNECTION

See Section 5.7 in GOEP [2].

6 SERVICE DISCOVERY

6.1 SD SERVICE RECORDS

The service belonging to the File Transfer profile is a server, which enables bi-directional generic file transfer. OBEX is used as a session protocol for this service. The following service records must be put into the SDDB.

Item	Definition:	Type/ Size:	Value:*	AttrID	Status	Default Value
Service Class ID List				See [15]	M	
Service Class #0		UUID	OBEX-File Transfer		M	
Protocol Descriptor list				See [15]	M	
Protocol ID #0		UUID	L2CAP		M	
Protocol ID #1		UUID	RFCOMM		M	
Param #0	CHANNEL	Uint8	Varies		M	
Protocol ID #2		UUID	OBEX		M	
Service name	Display-able Text name	String	Varies	See [15]	O	“OBEX File Trans-fer”
BluetoothProfileDe-scriptorList				See [15]	O	
Profile ID #0	Supported profile	UUID	OBEX File-Transfer			OBEX File Transfer [15]
Param #0	Profile ver-sion	uint16	0x100			0x100

Table 6.1: File Transfer Service Record

* UUID values are defined in the *Assigned Numbers* document.

6.2 SDP PROTOCOL DATA UNITS

Table 19 shows the specified SDP PDUs (Protocol Data Units) which are required in the File Transfer profile.

PDU no.	SDP PDU	Server	Client
1	SdpErrorResponse	M	M
2	SdpServiceSearch AttributeRequest	M	M
3	SdpServiceSearch AttributeResponse	M	M

Table 6.2: SDP PDUs Minimal Requirements

7 REFERENCES

7.1 NORMATIVE REFERENCES

[1] Bluetooth Special Interest Group, IrDA Interoperability

[2] Bluetooth Special Interest Group, Generic Object Exchange Profile

[3] Bluetooth Special Interest Group, Synchronization Profile

[4] Bluetooth Special Interest Group, Object Push Profile

[5] Bluetooth Special Interest Group, Baseband Specification

[6] Bluetooth Special Interest Group, LMP Specification

[7] Bluetooth Special Interest Group, L2CAP Specification

[8] Bluetooth Special Interest Group, RFCOMM with TS 07.10

[9] ETSI, TS 07.10, Version 6.3.0

[10] Bluetooth Special Interest Group, SDP Specification

[11] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX) with Published Errata, Version 1.2, April 1999.

[12] Infrared Data Association, IrMC (Ir Mobile Communications) Specification with Published Errata, Version 1.1, February 1999.

[13] The Internet Mail Consortium, vCard – The Electronic Business Card Exchange Format, Version 2.1, September 1996.

[14] The Internet Mail Consortium, vCalendar – The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996.

[15] Bluetooth Special Interest Group, Assigned Numbers specification http://www.bluetooth.org/assigned-numbers.htm

[16] Bluetooth Special Interest Group, Bluetooth Generic Access Profile Specification

Part K:13

SYNCHRONIZATION PROFILE

This application profile defines the application requirements for Bluetooth devices necessary for the support of the Synchronization usage model. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Synchronization usage model.

CONTENTS

1 Introduction 392
1.1 Scope 392
1.2 Bluetooth Profile Structure 392
1.3 Bluetooth OBEX Related Specifications 393
1.4 Symbols and conventions 394
1.4.1 Requirement status symbols 394
1.4.2 Signaling diagram conventions 395

2 Profile overview 396
2.1 Profile stack 396
2.2 Configurations and roles 396
2.3 User requirements and scenarios 397
2.4 Profile fundamentals 398

3 User interface aspects 399
3.1 Mode selection 399
3.2 Application Usage Events 399
3.2.1 Synchronization Scenario 400
3.2.2 Sync Command Scenario 401
3.2.3 Automatic Synchronization Scenario 401

4 Application layer 402
4.1 Feature overview 402
4.2 Synchronization Feature 402
4.3 Sync Command Feature 403
4.4 Automatic Synchronization Feature 403

5 IrMC Synchronization Requirements 404

6 OBEX 406
6.1 OBEX Operations Used 406
6.2 OBEX Headers 406
6.3 Initialization of OBEX 407
6.4 Establishment of OBEX session 407
6.5 Pushing Data 407
6.6 Pulling Data 407
6.7 Disconnection 407

7 Service Discovery 408

7.1 SD Service Records 408

7.1.1 Synchronization Service 408

7.1.2 Sync Command Service 409

7.2 SDP Protocol Data Units 410

8 References 411

8.1 Normative references 411

FOREWORD

This document, together with the Generic Object Exchange profile and the Generic Access profile forms the Synchronization usage model.

Interoperability between devices from different manufacturers is provided for a specific service and usage model if the devices conform to a Bluetooth-SIG defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications and gives an unambiguous description of the air interface for specified service(s) and usage model(s).

All defined features are process-mandatory. This means that if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

The Synchronization profile defines the requirements for the protocols and procedures that shall be used by the applications providing the Synchronization usage model. This profile makes use of the Generic Object Exchange profile (GOEP) to define the interoperability requirements for the protocols needed by applications. The most common devices using these usage models might be notebook PCs, PDAs, and mobile phones.

The scenarios covered by this profile are:

- Usage of a mobile phone or PDA by a computer to exchange PIM (Personal Information Management) data, including a necessary log information to ensure that the data contained within their respective Object Stores is made identical. Types of the PIM data are, for example, phonebook and calendar items.
- Use of a computer by a mobile phone or PDA to initiate the previous scenario (Sync Command Feature).
- Use of a mobile phone or PDA by a computer to automatically start synchronization when a mobile phone or PDA enters the RF proximity of the computer

1.2 BLUETOOTH PROFILE STRUCTURE

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

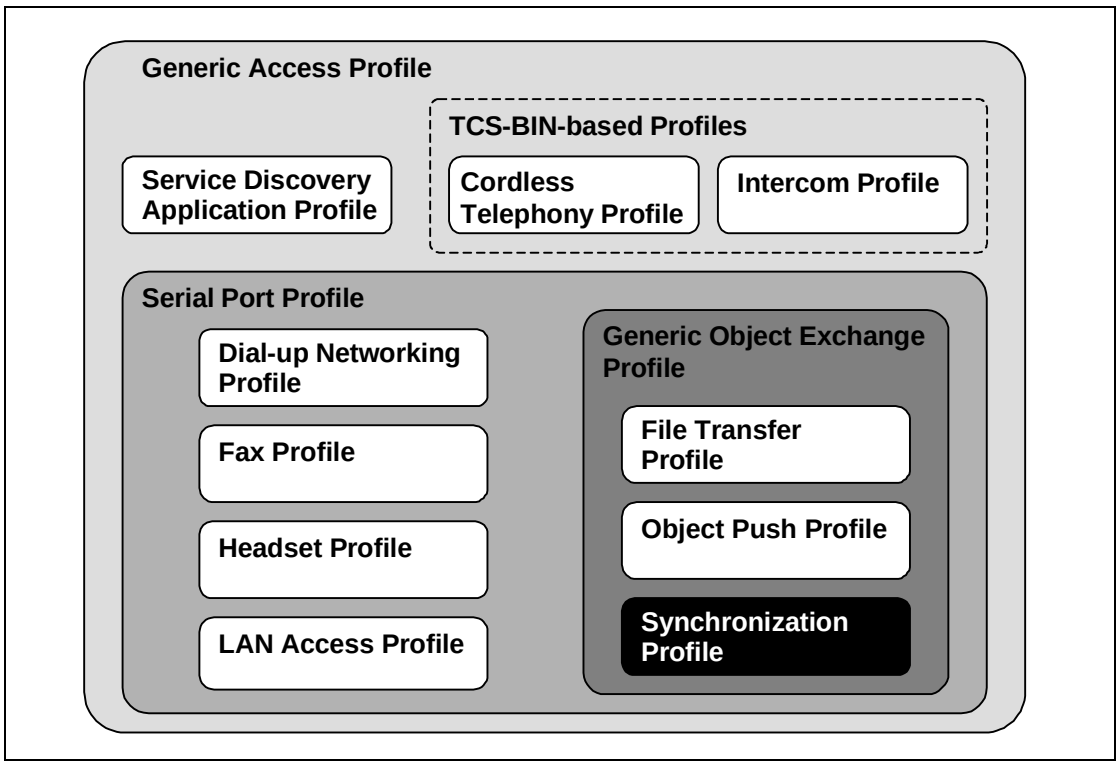


Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX RELATED SPECIFICATIONS

Bluetooth Specification includes five separate specifications for OBEX and applications using OBEX.

1. Bluetooth IrDA Interoperability Specification [1].

- Defines how the applications can function over both Bluetooth and IrDA.
- Specifies how OBEX is mapped over RFCOMM and TCP.
- Defines the application profiles using OBEX over Bluetooth.

2. Bluetooth Generic Object Exchange Profile Specification [2]

- Generic interoperability specification for the application profiles using OBEX.
- Defines the interoperability requirements of the lower protocol layers (e.g. Baseband and LMP) for the application profiles

3. Bluetooth Synchronization Profile Specification (This Specification)

- Application Profile for Synchronization applications.
- Defines the interoperability requirements for the applications within the Synchronization application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

4. Bluetooth File Transfer Profile Specification [3]

- Application Profile for File Transfer applications.
- Defines the interoperability requirements for the applications within the File Transfer application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

5. Bluetooth Object Push Profile Specification [4]

- Application Profile for Object Push applications.
- Defines the interoperability requirements for the applications within the Object Push application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document, the following symbols are used:

‘M’ for es that shall be used in the profile);

‘O’ for optional to support (used for capabilities that can be used in the profile);

‘C’ for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

‘X’ for excluded (used for capabilities that may be supported by the unit but shall never be used in the profile);

‘N/A’ for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

1.4.2 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

A		B
	PROC1	
	PROC2	
	PROC3	
	(PROC4)	
	(PROC5)	
	MSG1	
	MSG2	
	(MSG3)	
	(MSG4)	

Table 1.1: Arrows used in signaling diagrams

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A and B). PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates an optional message from B to A.

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

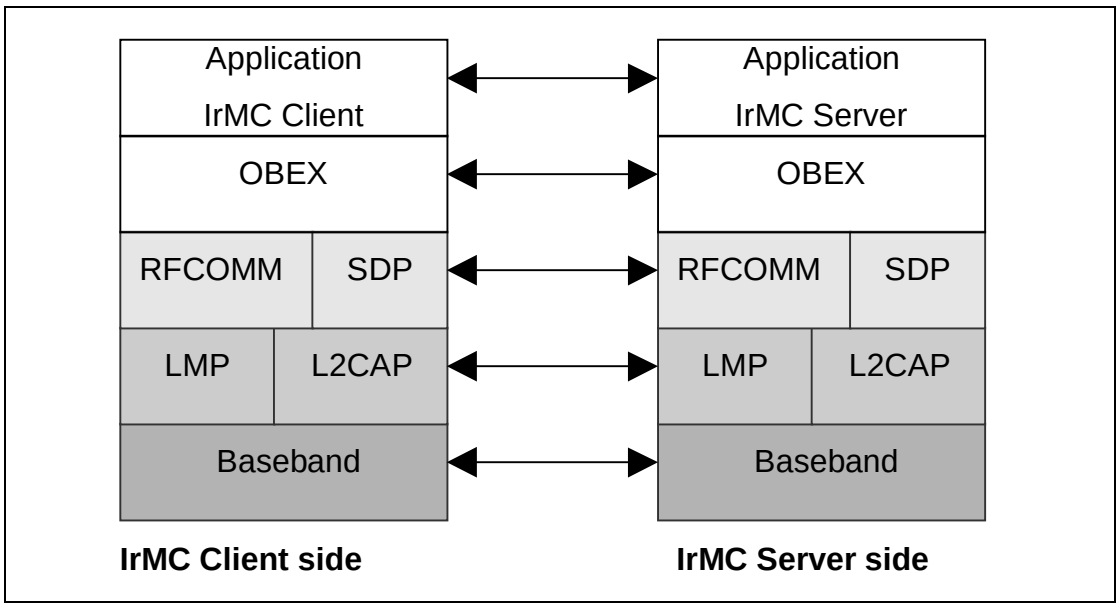


Figure 2.1: Protocol model

The Baseband [5], LMP [6] and L2CAP [7] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [8] is the Bluetooth adaptation of GSM TS 07.10 [9]. SDP is the Bluetooth Service Discovery Protocol [10]. OBEX [1] is the Bluetooth adaptation of IrOBEX [11].

The IrMC Client layer shown in Figure 2.1 is the entity processing the synchronization according to the IrMC specification [12], and the IrMC server is the server software compliant to the IrMC specification.

The RFCOMM, L2CAP, LMP, and Baseband interoperability requirements are defined in Section 6 in GOEP[2].

2.2 CONFIGURATIONS AND ROLES

Figure 2.2 depicts a synchronization example in which a mobile phone acts as an IrMC server and a PC notebook as an IrMC Client. The IrMC Client (PC) pulls the PIM data from the IrMC server and synchronizes this data with data stored in the IrMC client. After that, the IrMC client puts this synchronized data back to the IrMC server.

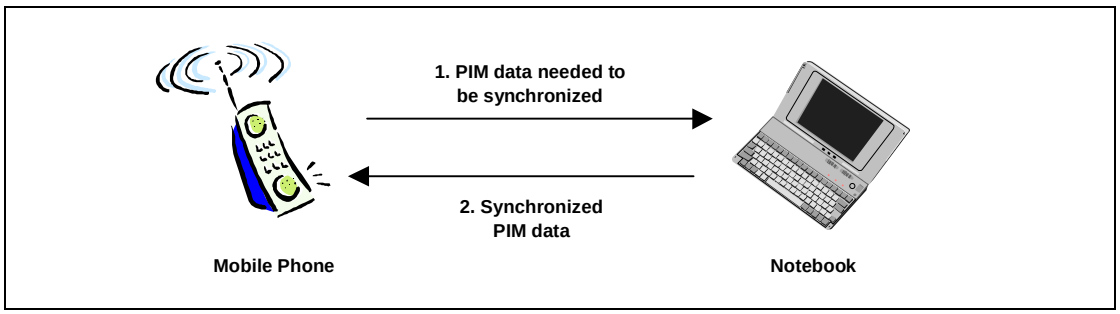


Figure 2.2: Synchronization Example with Mobile Phone and Computer

The following roles are defined for this profile:

IrMC Server – This is the IrMC server device that provides an object exchange server. Typically, this device is a mobile phone or PDA. In addition to the interoperability requirements defined in this profile, the IrMC server must comply with the interoperability requirements for the server of the GOEP, if not defined to the contrary.

If the IrMC Server also provides the functionality to initiate the synchronization, then it must act as a client temporarily. In this case, it must also comply with the requirements with the client of the GOEP if not defined in the contrary.

IrMC Client – This is the IrMC client device, which contains a sync engine and pulls and pushes the PIM data from and to the IrMC Server. Usually, the IrMC Client device is a PC. Because the IrMC Client must also provide functionality to receive the initialization command for synchronization, sometimes it must temporarily act as a server. In addition to the interoperability requirements defined in this profile, the IrMC server must also comply with the interoperability requirements for the server and client of the GOEP if not defined to the contrary.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are:

- Usage of an IrMC Server by an IrMC Client to pull the PIM data needed to be synchronized from the IrMC Server, to synchronize this data with the data on the IrMC Client, and to push this synchronized data back to the IrMC Server.
- Usage of an IrMC Client by an IrMC Server to initiate the previous scenario by sending a sync command to the IrMC Client.
- Automatic synchronization initiated by the IrMC client.

The restrictions applying to this profile are the same as in the GOEP. In addition to these restrictions, the peer-to-peer synchronization is not supported by the BT synchronization.

2.4 PROFILE FUNDAMENTALS

The profile fundamentals are the same as defined in Section 2.4 in GOEP [2], with the addition of the requirements that bonding, link level authentication, and encryption (Fundamentals 1 and 3 in GOEP) must always be used for this profile. The OBEX authentication (Fundamental 2 in GOEP) as an application-level security mechanism must be supported by the devices providing this profile, but this profile does not mandate that it must be used.

In this profile, because both the IrMC Client and IrMC Server can act as a client (IrMC Server temporarily), both can initiate link and channel establishments; i.e. create a physical link between these two devices.

This profile does not mandate the IrMC server or client to enter any discoverable or connectable modes automatically, even if they are able to do so. This means that the end-user intervention may be needed on both the devices when, for example, the synchronization is initiated on the IrMC client device.

3 USER INTERFACE ASPECTS

3.1 MODE SELECTION

There are two modes associated with the Synchronization profile.

- Initialization Sync mode
- General Sync mode

In the **Initialization Sync** mode, the IrMC Server is in the Limited discoverable (or the General discoverable mode, see Section 6.5.1 in GOEP [2]), Connectable, and Pairable modes (See Section 4 in GAP [16]). The IrMC Client does not enter this mode in this profile. It is recommended that the Limited Inquiry procedure (Section 6.2 in GAP[16]) is used by the IrMC Client when discovering the IrMC server. Requirements on inquiry procedures are discussed in Section 6.5.1 of the GOEP [2].

In the **General Sync** mode, the device is in the Connectable mode. Both the IrMC Client and Server can enter this mode. For the IrMC Server, this mode is used when the IrMC Client connects the server and starts the synchronization at the subsequent times after pairing. For the IrMC Client, the mode is used when the synchronization is initiated by the IrMC server.

The devices are not required to enter these modes automatically without user intervention, even if they can do so. When entering either of these modes, IrMC Server and Client must ensure that the Object Transfer bit is set in the CoD (See [15]), and register a service record in the SDDB (See Section 7).

3.2 APPLICATION USAGE EVENTS

In the following sections (Section 3.2.1-3.2.3), the presented scenarios work as examples and variations in the actual implementations are possible and allowed.

3.2.1 Synchronization Scenario

When an IrMC Client wants to synchronize with an IrMC Server for the first time, the following scenario (Table 3.1) can be followed:

Step	IrMC Client	IrMC Server
1		The IrMC server device must be in the General Sync mode. If the device is not in this mode, the user must activate this mode on the device.
2	The user activates an application for synchronization.	
3	A list of devices in the RF proximity of the IrMC client is displayed to the user.	
4	The user selects a device to be connected and synchronized.	
5	The user is alerted if the device does not support the Synchronization feature, and the user may select another possible device (Step 4).	
6	The Bluetooth PIN code is requested from the user and entered on both devices.	
7	If OBEX authentication is used, the user enters the password for the OBEX authentication on both devices.	
8	The first synchronization is processed.	
9	The user may be notified of the result of the operation.	

Table 3.1: Usage Events for First Time Synchronization

At subsequent times, when the bonding is done, the scenario below (Table 3.2) can be followed.:

Step	IrMC Client	IrMC Server
1		The IrMC server device must be in the General Sync mode. If the device is not in this mode, the user must activate this mode on the device.
2	The user of the IrMC Client selects the Synchronization feature on the device, or another event triggers the synchronization to start in the IrMC client.	
3	The synchronization is processed.	
4	The User may be notified of the result of the operation.	

Table 3.2: Usage Events after First Time Synchronization

3.2.2 Sync Command Scenario

When an IrMC Server wants to initiate synchronization, and when the bonding and the possible OBEX initialization are done, the scenario below (Table 3.3) can be followed:

Step	IrMC Client	IrMC Server
1	The IrMC Client should be in the General Sync mode, without user intervention. Otherwise this operation is not applicable.	
2		The user selects the Sync Command feature in the IrMC Server, and the synchronization is initiated with the IrMC client. On the IrMC Server device, the user has earlier configured the IrMC Client to which the sync command is sent.
3	The synchronization is processed.	
4		The User may be notified of the result of the operation.

Table 3.3: Usage Events of Sync Command Scenario

3.2.3 Automatic Synchronization Scenario

When it is desired that an IrMC Server and Client synchronize automatically, and when the bonding and (possible) OBEX initialization are done, the scenario below (Table 3.4) can be followed.

Step	IrMC Client	IrMC Server
1	The IrMC Server enters the RF proximity of the IrMC Client. The Client notices it, and starts the synchronization without any notification to the User. The IrMC Server must be constantly in the General Sync mode so that the IrMC Client can notice the presence of the server in its RF vicinity.	
2	The synchronization is processed.	
3	The User may be notified of the result of the operation on both the devices.	

Table 3.4: Usage Events of Automatic Synchronization Scenario

4 APPLICATION LAYER

This section describes the feature requirements on units active in the Synchronization use case.

4.1 FEATURE OVERVIEW

Table 4.1 shows the required services:

	Features	Support in IrMC Client	Support in IrMC Server
1.	Synchronization of one or more of the following cases:	M	M
	Synchronization of phonebooks		
	Synchronization of calendars		
	Synchronization of emails		
	Synchronization of notes		
2.	Sync Command	M	O
3.	Automatic Synchronization	O	M

Table 4.1: Application layer features

4.2 SYNCHRONIZATION FEATURE

The support of Synchronization with IrMC level 4 functionality is mandatory for both IrMC Clients and IrMC Servers. The requirements for IrMC Synchronization are defined in the IrMC spec (See also Section 5). Bluetooth Synchronization must support at least one of the following cases (i.e. the application classes):

1. Synchronization of phonebooks
2. Synchronization of calendars
3. Synchronization of messages
4. Synchronization of notes

To achieve application level interoperability, the content formats are defined for Bluetooth Synchronization. The content formats are dependent on the application classes, which are designed for the different purposes. The supported application classes must be identified in terms of the data stores in the SDDB of the IrMC Server (See Section 7.1.1). For the application classes the content format requirements are:

- Phone Book applications must support data exchange using the vCard 2.1 content format specified in [13]. Section 7 of IrMC Specification [12] includes extensions to vCard2.1, which must also be supported by the actual implementations.
- Calendar applications must support data exchange using the vCalendar 1.0 content format specified in [14].
- Messaging applications must support data exchange using the vMessage content format in Section 9 of [12].
- Notes applications must support data exchange using the vNote content format specified in Section 10 of [12].

The above requirements are the minimal requirements, and the application utilizing any of these classes may store its objects in any internal content format the implementer chooses.

The support for the various mandatory and optional fields of the content formats listed above shall be in accordance with the IrMC Specification [12].

4.3 SYNC COMMAND FEATURE

This feature means that the IrMC client device works temporarily as a server and is able to receive a Sync Command from the IrMC server, which in this case acts temporarily as a client. This Sync Command orders the IrMC client to start synchronization with the IrMC Server.

After sending the sync command and getting the response for it, the IrMC Server must terminate the OBEX session and the RFCOMM data link connection.

This feature must be supported by the IrMC Client and it can optionally be supported by the IrMC Server. The formal requirements for this feature are defined in Section 5.8 in [12].

4.4 AUTOMATIC SYNCHRONIZATION FEATURE

In this feature, the IrMC Client can start the synchronization when the IrMC Server enters the RF proximity of the IrMC Client. Basically, this means that, on the Baseband level, the IrMC Client pages the IrMC Server at intervals and, when it finds that the IrMC Server is in the range, the IrMC Client can begin synchronization.

The support of this feature is optional for the IrMC Client but mandatory for the IrMC Server. This means that the IrMC Server must offer a capability to put the server device into the General Sync mode so that it does not leave this mode automatically.

5 IRMC SYNCHRONIZATION REQUIREMENTS

The IrMC specification [12] specifies IrMC Synchronization, which is utilized by this profile. The sections of the IrMC specification, with which this profile complies, are defined in Table 5.1.

Chapter	Name	Informative Sections	Mandatory Sections	Optional Sections	Not Applicable Sections
1	Introduction	All	-	-	-
2	IrMC Framework	2.1-3, 2.5.1, and 2.6-7	2.8.1-2, 2.8.4, and 2.9 (except 2.9.2)	2.8.3, and 2.9.2	2.4 and 2.5.2-3
3	Data Transmis-sions Services	3.3	3.1	-	3.2
4	OBEX Informa-tion Access and Indexing	4.1, 4.4.2, and 4.7	4.1.2, 4.2-3, 4.6, and 4.8	4.1.1 and 4.5	4.4.1
5	Synchronization	5.1 and 5.7	5.2-6 (except 5.5.3), and 5.8	5.5.3	-
6	Device Informa-tion	-	6.1-2	-	
7	Phone Book	7.1	7.3, 7.5, 7.7.1, 7.7.3, 7.7.5, 7.8.1, and 7.8.2	7.4, 7.6, 7.7.4, 7.7.6, and 7.8.3-5	7.2 and 7.7.2
8	Calendar	8.1	8.3, 8.5, 8.6.1, 8.6.3, 8.6.5, and 8.7	8.4 and 8.6.4	8.2, and 8.6.2
9	Messaging	9.1	9.3, 9.5, 9.8.1, 9.8.3, 9.8.6, and 9.9-10	9.4, 9.6-7, 9.8.4, and 9.8.5	9.2, and 9.8.2
10	Notes	10.1	10.3, 10.5, 10.6.1, 10.6.3, 10.6.5, and 10.7	10.4 and 10.6.4	10.2, and 10.6.2

Table 5.1: IrMC Specification Dependencies

Chapter	Name	Informative Sections	Mandatory Sections*	Optional Sections	Not Applicable Sections
11	Call Control	-	-	-	ALL
12	Audio	-	-	-	ALL
13	IrMC Applications IAS Entry and Service Hint Bit	-	-	-	ALL

Table 5.1: IrMC Specification Dependencies

*. Some of these sections may not be mandatory if the applications do not support all of the applications classes

This profile does not mandate that the functionality of IrMC level 1 must be supported for the different personal data objects (vcard, vcal, vmessage and vnote), although the IrMC specification requires its support. However, it is worth mentioning that the Push command of IrMC requires the level1 functionality for a text message. Thus, the IrMC client must be able to receive this command into its Inbox and the IrMC server must be able to send this command, if support for the Sync Command feature is claimed. For Bluetooth, the object push functionality and requirements are defined in the Object Push profile.

6 OBEX

6.1 OBEX OPERATIONS USED

Table 6.1 shows the OBEX operations which are required in the Synchronization profile.

Operation no.	OBEX Operation	Ability to Send		Ability to Respond	
		IrMC Client	IrMC Server*	IrMC Client*	IrMC Server
1	Connect	M	O	M	M
2	Disconnect	M	O	M	M
3	Put	M	O	M	M
4	Get	M	X	X	M
5	Abort	M	O	M	M
6	SetPath	X	X	X	X

Table 6.1: OBEX Operations

The columns marked with '*' refer to the Sync Command feature for which support in the IrMC Server is optional.

6.2 OBEX HEADERS

Table 6.2 shows the specified OBEX headers which are required in the Synchronization profile.

Header No.	OBEX Headers	IrMC Client	IrMC Server
1	Count	X	X
2	Name	M	M
3	Type	X	X
4	Length	M	M
5	Time	O	O
6	Description	O	O
7	Target	M	M

Table 6.2: OBEX Headers

Header No.	OBEX Headers	IrMC Client	IrMC Server
8	HTTP	O	O
9	Body	M	M
10	End of Body	M	M
11	Who	M	M
12	Connection ID	M	M
13	Authenticate Challenge	M	M
14	Authenticate Response	M	M
15	Application Parameters	M	M
16	Object Class	X	X

Table 6.2: OBEX Headers

6.3 INITIALIZATION OF OBEX

OBEX authentication must be supported by the devices implementing the Synchronization profile. The initialization procedure for OBEX is defined in Section 5.3 in GOEP [2].

6.4 ESTABLISHMENT OF OBEX SESSION

The Target header must be used when the IrMC client establishes the connection (See Section 5.4 in GOEP [2]). The Target header value is 'IRMC-SYNC'.

6.5 PUSHING DATA

See Section 5.5 in GOEP [2].

6.6 PULLING DATA

See Section 5.6 in GOEP [2].

6.7 DISCONNECTION

See Section 5.7 in GOEP [2].

7 SERVICE DISCOVERY

7.1 SD SERVICE RECORDS

There are two separate services related to the Synchronization profile. The first is the actual synchronization server (i.e. IrMC server), and the second is the sync command server (i.e. IrMC Client).

7.1.1 Synchronization Service

In this case, the service is the IrMC server. The following information (i.e. service records) must be put into the SDDB.

Item	Definition:	Type/ Size:	Value:*	AttrID:	Status:	Default Value:
Service Class ID List				See [15]	M	
Service Class #0		UUID	IrMCSync		M	
Protocol Descriptor list				See [15]	M	
Protocol ID #0		UUID	L2CAP		M	
Protocol ID #1		UUID	RFCOMM		M	
Param #0	CHANNEL	Uint8	Varies		M	
Protocol ID #2		UUID	OBEX		M	
Service name	Displayable Text name	String	Varies	See [15]	O	'IrMC Synchro-nization'
BluetoothProfileDe-scriptorList	Supported profiles and versions			See [15]	O	
Profile #0		UUID	IrMCSync			IrMC-Sync
Version #0		Uint16	Varies			0x0100
Supported Data Stores List	Data stores may be phonebook, calendar, notes, and messages.	Data Ele-ment Sequence of UInt8	Data stores: **0x01** = Phonebook **0x03** = Cal-endar **0x05** = Notes **0x06** = Mes-sages	See [15]	M	

Table 7.1: Synchronization Service Record

*. Values that are of the type UUID are defined in the Assigned Numbers specification [15].

7.1.2 Sync Command Service

The Sync Command service is used for initiating the synchronization from the IrMC server device. The following service records must be put into the SDDB by the application which provides this service.

Item	Definition:	Type/ Size:	Value:*	AttrID:	Status:	Default Value:
Service Class ID List				See [15]	M	
Service Class #0		UUID	IrMCSync-Command		M	
Protocol Descriptor list				See [15]	M	
Protocol ID #0		UUID	L2CAP		M	
Protocol ID #1		UUID	RFCOMM	See [15]	M	
Param #0	CHANNEL	Uint8	Varies		M	
Protocol ID #2		UUID	OBEX		M	
Service name	Displayable Text name	String	Varies		O	'Sync Com-mand Service'
BluetoothProfileDe-scriptorList	Supported profiles and versions			See [15]	O	
Profile #0		UUID	IrMCSync			IrMC-Sync
Version #0		Uint16	Varies			0x0100

Table 7.2: Sync Command Service Record

*. Values that are of the type UUID are defined in the Assigned Numbers specification [15].

7.2 SDP PROTOCOL DATA UNITS

Table 7.3 shows the specified SDP PDUs (Protocol Data Units) which are required in the Synchronization profile.

PDU no.	SDP PDU	Ability to Send		Ability to Retrieve	
		IrMC Client	IrMC Server	IrMC Client	IrMC Server
1	SdpErrorResponse	M*	M	M	O*
2	SdpServiceSearchAttribute-Request	M	O*	M*	M
3	SdpServiceSearchAttribute-Response	M*	M	M	O*

Table 7.3: SDP PDUs

The PDUs marked with '*' refer to the Sync Command feature, of which the support in the IrMC Server is optional.

8 REFERENCES

8.1 NORMATIVE REFERENCES

[1] Bluetooth Special Interest Group, IrDA Interoperability.

[2] Bluetooth Special Interest Group, Generic Object Exchange Profile.

[3] Bluetooth Special Interest Group, File Transfer Profile.

[4] Bluetooth Special Interest Group, Object Push Profile.

[5] Bluetooth Special Interest Group, Baseband Specification.

[6] Bluetooth Special Interest Group, LMP Specification.

[7] Bluetooth Special Interest Group, L2CAP Specification.

[8] Bluetooth Special Interest Group, RFCOMM with TS 07.10.

[9] ETSI, TS 07.10, Version 6.3.0.

[10] Bluetooth Special Interest Group, SDP Specification.

[11] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX) with Published Errata, Version 1.2, April 1999.

[12] Infrared Data Association, IrMC (Ir Mobile Communications) Specification with Published Errata, Version 1.1, February 1999.

[13] The Internet Mail Consortium, vCard – The Electronic Business Card Exchange Format, Version 2.1, September 1996.

[14] The Internet Mail Consortium, vCalendar – The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996.

[15] Bluetooth Special Interest Group, Bluetooth Assigned Numbers. http://www.bluetooth.org/assigned-numbers.htm

[16] Bluetooth Special Interest Group, Bluetooth Generic Access Profile Specification.

Appendix I

REVISION HISTORY

REVISION HISTORY

Part K:1
Generic Access Profile

Rev	Date	Comments
1.0	June 20th 1999	• Released for final review. • Release for sign-off. • Updated based on received comments. • Final updated version. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars. Changes are also made in the following Figures: 5.1, 5.1, 7.1, 7.2 and 10.2
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:2
Service Discovery Application Profile

Rev	Date	Comments
1.0	June 20th 1999	• Aligned with GAP whenever necessary. • Emphasized that SDAP can be used as the basis for the service discovery portion of other profiles. • Added section 5.1 with SDP PDU exchange examples. • Emphasized that normal operation requires a LocDev to initiate and terminate L2CAP connections for SDP. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:3
Cordless Telephony Profile

Rev	Date	Comments
1.0	June 20th 1999	• Ftf preversion, some editorial updates and minor content changes - number of TLs 8 -> 7, Master-Slave Switch made conditional, restrictions in digits for Called&Calling party IEs, updates to CoD and SDP sections. • Updates after ftf. Added feature "Register recall", removed feature "service call" and redefined "Multi-terminal support" to reflect decisions on WUG status. Added description of Register recall to section 4.3. Removed emergency, service and ad-hoc call classes. Added description of piconet handling to 4.1.2. Updated and reworked SDP record. Additions to contributor list. Figure in section 8.2 removed. "Status" chapter removed. Added remark on security with respect to L2CAP connectionless. Editorial updates to section 4.4. • Updates to incorporate GAP and editorial guidelines for the specification • Errors in tables 3 and 4 and section 4.2. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:4
Intercom Profile

Rev	Date	Comments
1.0	June 20th 1999	• Update after F2F, incorporating technical issues only. • Editorial improvements. • Replaced bonding with authentication in Section 2.4. • Corrected references to LMP. • Removed PSM field from service record, and rephrased opening statement of SDP section. • Added chapter on GAP interoperability requirements. • Final GAP alignment. • Mandated call confirmation as SETUP confirmation. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:5
Serial Port Profile

Rev	Date	Comments
1.0	June 20th 1999	• Added more details on application layer procedures (chapter 3). First alignment with Generic Access Profile. • Added requirements on SDP procedures. More alignment with GAP. • Corrected some typos. Removed section 5.3.3 (Link Power Mode in L2CAP). Removed “Management entity” throughout document. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:6
Headset Profile

Rev	Date	Comments
1.0	June 20th 1999	• Update after F2F, incorporating outstanding issues as discussed (volume control and synchronisation, added AT command +VGS and +VGM, extended audio connection transfer description, authentication/ encryption optional to use, status change of outgoing audio connection, service record updated) and various editorial issues (amongst others update of contributors list). Removed status and history section. • Remote audio volume control: replaced may’s with shall’s to make it more consistent (if Remote audio volume control is supported, the entire procedure shall be supported as specified). • SDP - removed PSM for RFCOMM, added misplaced server channel. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:7
Dial-up Networking Profile

Rev	Date	Comments
1.0	June 20th 1999	• Some SDP values filled in, CoD updated after assigned numbers doc. • Updates after Tampere ftf: SDP record updated and reworked. Removed table from chapter 5.1 (now in RFCOMM). Updated contributors list. Figure removed from section 5.5.1. • Added profile structure section. Alignment with GAP (section 6) added. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:8
Fax Profile

Rev	Date	Comments
1.0	June 20th 1999	• Replaced bonding with authentication in Section 2.4. • Corrected references to LMP. • removed PSM field from service record, and rephrased opening statement of SDP section. • Added chapter on GAP interoperability requirements. • Updated Figure 1, Service discovery Profile to Service Discovery Application Profile. • Removed “ME” block from both sides of figure 2. • Removed paragraph discussing “ME” in section 2.1. • Renamed Section Heading 4 From Dialling and Control to Dialling and Control Interoperability requirements. • Re-worded section 4.1.2. • Removed the words “the” and “section” from the last sentence in section 5.3, paragraph 2. • Removed section 5.6. • Aligned section 6 with new changes from Dialup networking profile. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:9
LAN Access Profile

Rev	Date	Comments
1.0	June 20th 1999	• Updated Service records in line with “best practice”. • Removed Security section. • Editorial changes in Section 4.1. • Editorial changes in Section 5.1. • Editorial changes in Section 11.2. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.
1.1	Dec 1st 2000	• Errata items previously published on the web have been included and are marked with a change bar.

Part K:10
Generic Object Exchange Profile

Rev	Date	Comments
1.0	June 20th 1999	• Updated Chapter 1.2 and added reference to GAP regarding link and channel establishment. • Removed the security statement from the Profile fundamentals chapter, clarified the use of the limited discoverable mode in the Inquiry and Inquiry Scan chapter, and added the GAP requirement chapter. • Changed the 'initialization' wording to 'bonding', added some cross-references, and included the errata for IrOBEX in the reference list. • Management entity removed and the fall back procedure added if the Limited Inquiry procedure is supported. • Clarified that the fall back to the General inquiry is mandatory if Limited Inquiry is used. • Editorial changes and Chapter 7.3.1 (Bonding) updated to describe the result of Bonding. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.

Part K:11
Object Push Profile

Rev	Date	Comments
1.0	June 20th 1999	• Removed PSM from SDP record. Updated text in Profile Structure 1.2. GAP alignment in Profile Fundamentals. • Editorial. • Renamed Initialization to Bonding. • Removed the ME section and references to ME. Stated in profile fundamentals that bonding is mandatory to support and optional to use. Removed “Notation for timers and counters”. Changed wording in application procedure for object push feature. Minor update of SDP record. • Changed recommended inquiry procedure in chapter 3 to reference to the GOEP. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:12
File Transfer Profile

Rev	Date	Comments
1.0	June 20th 1999	• SDP table changes, addition of references to doc [16]. • More SDP table changes, alignment with GAP, contributors update, and copyright notice. • Editorial and reference corrections. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.

Revision History

Part K:13
Synchronization Profile

Rev	Date	Comments
1.0	June 20th 1999	• Updated service records, IrMC chapter updated. • Chapter 1.2 updated, profile fundamentals clarified, recommended inquiry procedure added into Chapter 3 and service records updated. • Security issues clarified in Profile Fundamentals chapter and some editorial changes. • Change the 'Initialization' wording to 'Bonding', updated cross-references, and editorial changes. • Remove Management entity, removed statement that IrMC client must initiate the link establishment when bonding is not performed, and added a reference to the inquiry procedures of GOEP. • Editorial changes. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Appendix II

CONTRIBUTORS

CONTRIBUTORS

Part K:1
Generic Access Profile

Ken Morley	3Com Corporation
Chatschik Bisdikian	IBM Corporation
Jon Inouye	Intel Corporation
Brian Redding	Motorola, Inc.
David E. Cypher	NIST
Stephane Bouet	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Martin Roter	Nokia Mobile Phones
Johannes Elg	Telefonaktiebolaget LM Ericsson
Patric Lind (Section Owner)	Telefonaktiebolaget LM Ericsson
Erik Slotboom	Telefonaktiebolaget LM Ericsson
Johan Sörensen	Telefonaktiebolaget LM Ericsson

Part K:2
Service Discovery Application Profile

Chatschik Bisdikian (Section Owner)	IBM Corporation
Brent Miller	IBM Corporation
Thomas Müller	Nokia Mobile Phones
Johannes Elg	Telefonaktiebolaget LM Ericsson
Robert A. Pascoe	XtraWorX

Contributors

Part K:3
Cordless Telephony Profile

Ken Morley	3Com Corporation
Richard Shaw	3Com Corporation
Sridhar Rajagopal	Intel Corporation
Ramu Ramakesavan	Intel Corporation
Alex Feinman	Motorola, Inc.
Brian Redding	Motorola, Inc.
Martin Roter	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Olof Dellien	Telefonaktiebolaget LM Ericsson
Erik Slotboom (Section Owner)	Telefonaktiebolaget LM Ericsson
Gert-jan van Lieshout	Telefonaktiebolaget LM Ericsson
Jun'ichi Yoshizawa	Toshiba Corporation

Part K:4
Intercom Profile

Ken Morley	3Com Corporation
Richard Shaw	3Com Corporation
Sridhar Rajagopal	Intel Corporation
Ramu Ramakesavan	Intel Corporation
Brian Redding	Motorola, Inc.
Thomas Müller	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Olof Dellien	Telefonaktiebolaget LM Ericsson
Erik Slotboom (Section Owner)	Telefonaktiebolaget LM Ericsson
Gert-jan van Lieshout	Telefonaktiebolaget LM Ericsson
Jun’ichi Yoshizawa	Toshiba Corporation

Part K:5
Serial Port Profile

Riku Mettälä	Nokia Mobile Phones
Olof Dellien	Telefonaktiebolaget LM Ericsson
Johan Sörensen (Section Owner)	Telefonaktiebolaget LM Ericsson

Contributors

Part K:6
Headset Profile

Ken Morley	3Com Corporation
Richard Shaw	3Com Corporation
John Bromell	Cambridge Consultants
Bailey Cross	Intel Corporation
Sridhar Rajagopal	Intel Corporation
Alex Feinman	Motorola, Inc.
Brian Redding	Motorola, Inc.
Bjorn Bunte	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Martin Roter	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Olof Dellien	Telefonaktiebolaget LM Ericsson
Erik Slotboom (Section Owner)	Telefonaktiebolaget LM Ericsson
Jun'ichi Yoshizawa	Toshiba Corporation

Part K:7
Dial-up Networking Profile

Ken Morley	3Com Corporation
Richard Shaw	3Com Corporation
Sridhar Rajagopal	Intel Corporation
Alex Feinman	Motorola, Inc.
Brian Redding	Motorola, Inc.
Riku Mettälä	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Martin Roter	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Olof Dellien	Telefonaktiebolaget LM Ericsson
Erik Slotboom (Section Owner)	Telefonaktiebolaget LM Ericsson
Jun'ichi Yoshizawa	Toshiba Corporation

Part K:8
Fax Profile

Ken Morley	3Com Corporation
Richard Shaw	3Com Corporation
Sridhar Rajagopal	Intel Corporation
Ramu Ramakesavan	Intel Corporation
Alex Feinman	Motorola, Inc.
Brian Redding	Motorola, Inc.
Riku Mettälä	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Martin Roter	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Olof Dellien	Telefonaktiebolaget LM Ericsson
Erik Slotboom (Section Owner)	Telefonaktiebolaget LM Ericsson
Gert-jan van Lieshout	Telefonaktiebolaget LM Ericsson
Jun'ichi Yoshizawa	Toshiba Corporation

Part K:9
LAN Access Profile

Jon Burgess	3Com Corporation
Phil Crooks	3Com Corporation
Dean Gratton (Section Owner)	3Com Corporation
Paul J. Moran	3Com Corporation
Pravin Bhagwat	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Marcia Peters	IBM Corporation
Kris Fleming	Intel Corporation
Von Mock	Motorola, Inc.
Stephane Bouet	Nokia Mobile Phones
Michael Camp	Nokia Mobile Phones
Johan Sörensen	Telefonaktiebolaget LM Ericsson
Shaun Astarabadi	Toshiba Corporation
Yosuke Tajika	Toshiba Corporation

Contributors

Part K:10
Generic Object Exchange Profile

David Kammer	3Com Corporation
David Suvak	Extended Systems
Apratim Purakayastha	IBM Corporation
Aron Walker	IBM Corporation
Jon Inouye	Intel Corporation
Stephane Bouet	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phones
James Scales	Nokia Mobile Phones
Steve Rybicki	PumaTech
Patrik Olsson (Section Owner)	Telefonaktiebolaget LM Ericsson
Shaun Astarabadi	Toshiba Corporation
Katsuhiro Kinoshita	Toshiba Corporation

Part K:11
Object Push Profile

David Suvak	Extended Systems
Apratim Purakayastha	IBM Corporation
Aron Walker	IBM Corporation
Jon Inouye	Intel Corporation
Stephane Bouet	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phones
James Scales	Nokia Mobile Phones
David Kammer	Palm
Steve Rybicki	PumaTech
Patrik Olsson (Section Owner)	Telefonaktiebolaget LM Ericsson
Shaun Astarabadi	Toshiba Corporation
Katsuhiro Kinoshita	Toshiba Corporation

Part K:12
File Transfer Profile

David Suvak	Extended Systems
Apratim Purakayastha	IBM Corporation
Aron Walker	IBM Corporation
Jon Inouye	Intel Corporation
Mike Foley	Microsoft Corporation
Stephane Bouet	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phones
James Scales	Nokia Mobile Phones
Steve Rybicki	PumaTech
Patrik Olsson	Telefonaktiebolaget LM Ericsson
Shaun Astarabadi (Section Owner)	Toshiba Corporation
Katsuhiro Kinoshita	Toshiba Corporation

Part K:13
Synchronization Profile

David Kammer	3Com Corporation
David Suvak	Extended Systems
Brent Miller	IBM Corporation
Apratim Purakayastha	IBM Corporation
Aron Walker	IBM Corporation
Jon Inouye	Intel Corporation
Stephane Bouet (Section Owner)	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phones
James Scales	Nokia Mobile Phones
Steve Rybicki	PumaTech
Patrik Olsson	Telefonaktiebolaget LM Ericsson
Shaun Astarabadi	Toshiba Corporation
Katsuhiro Kinoshita	Toshiba Corporation

Appendix III

ACRONYMS AND ABBREVIATIONS

LIST OF ACRONYMS AND ABBREVIATIONS

Abbreviation or Acronym	Meaning
ACL	Asynchronous Connectionless
AG	Audio Gateway
AP	Access Point
B	
BB	Baseband
BD_ADDR	Bluetooth Device Address
C	
CC	Call Control
CL	Connectionless
CO	Connection-oriented
CoD	Class Of Device
CTP	Cordless Telephony Profile
D	
DAC	Device Access Code
DIAC	Dedicated Inquiry Access Code
DT	Data Terminal
DT	Data Terminal
F	
FHS	Frequency Hopping Synchronization
G	
GAP	Generic Access Profile
GIAC	General Inquiry Access Code
GM	Group Management
GOEP	Generic Object Exchange Profile
GW	Gateway
H	
HCI	Host Controller Interface
HS	Headset

I	
IP	Internet Protocol
IPX	Internet Protocol eXchange
IrDA	Infrared Data Association
IrMC	Ir Mobile Communications
L	
L2CA	Logical Link Control And Adaptation
L2CAP	Logical Link Control And Adaptation Protocol
LAN	Local Area Network
LAP	LAN Access Point
LC	Link Controller
LIAC	Limited Inquiry Access Code
LM	Link Manager
LMP	Link Manager Protocol
LocDev	Local Device
M	
ME	Management Entity
MM	Mobility Management
MSC	Message Sequence Chart
MTU	Maximum Transmission Unit
O	
OBEX	Object Exchange Protocol
P	
PC	Personal Computer
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PIM	Personal Information Management
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
Q	
QoS	Quality Of Service

R	
RemDev	Remote Device
RFCOMM	Serial Cable Emulation Protocol
S	
SD	Service Discovery
SDDB	Service Discovery Database
SDP	Service Discovery Protocol
SeP	Serial Port
SIG	Special Interest Group
SrvDscApp	Service Discovery Application
T	
TCP	Transport Control Protocol
TCS	Telephony Control Specification
TL	Terminal
TL_O	Terminal Originating A Call
TL_T	Terminal Terminating A Call
U	
UDP	User Datagram Protocol
UI	User Interface
UIAC	Unlimited Inquiry Access Code
UUID	Universally Unique Identifier
W	
WUG	Wireless User Group

INDEX

Page numbers are hypertext links.

Numerics

3-in-1 phone 100

A

adaptation layer 178
Assigned Numbers 383
AT command set 231, 254
Audio connection transfer 204
Audio Gateway 197
Authentication 54
authentication 33, 173, 175
Automatic Synchronization 402

B

Bluetooth Bonding 42
Bluetooth channel establishment 48
Bluetooth connection establishment 50
Bluetooth Device Address 25
Bluetooth Device Class 27
Bluetooth Device Discovery 41
Bluetooth Device Inquiry 37, 39
Bluetooth Device Name 26
Bluetooth Device Name Discovery 40
Bluetooth Device Type 27
Bluetooth link establishment 45
Bluetooth Passkey 26
Bluetooth Service Type 27
bondable mode 32
Bonding 55, 173
bonding 42
browsing 174
Business Card Exchange 346
Business Card Exchange function 340
Business Card Pull 345
Business Card Pull function 340

C

Calendar 344
Calendar application 403
Call information 106, 147
Calling line identification presentation 106
Channel establishment 54
channel establishment 48
Class of Device 185
Client 310
CLIP 106
Connectable device 53
connectable mode 31
Connection establishment 54
connection establishment 50
Connection ID 375, 376
Connection Management 106
Content Formats 344
content formats 344
Cordless Telephony profile 100

D

Data Access Points 246
Data calls 230
data link connection 175
Data Terminal 227, 250, 272
DCE 172
Default Get Object 346
Device discovery 54
Dial-up Networking 223
Direct Cable Connection 294
Discoverable device 53
discoverable mode 29
DT 433
DTE 172
DTMF signalling 106

E

encryption 173, 175, 365

F

Fax profile 246
Fax service 253
File Browsing UUID 375
File Transfer 359
File Transfer mode 367
File Transfer profile 360
flow control 178
flush timeout 180

G

Gateway 104, 227, 250
General Discoverable 367
general discoverable mod 31
general inquir 37
General Inquiry procedure 185

General Sync mode 399
Generic Object Exchange profile 306, 360, 392
GET-operation 322
GOEP 360, 392
GW 104

H

Headset 193, 197
Headset Control 196

I

Incoming audio connection 201
Incoming external call 106
Initialisation 106
Initialization Sync mode 399
inquiry 185
inquiry scan 185
Intercom 143
Intercom call 107, 147
Internet Bridge 223
IP 269
IPX 269
IrMC Client 397
IrMC Server 397
IrMC specification 404
IRMC_SYNC 407

L

LAN Access 269
LAN Access Point 271
latency 180
legacy applications 171, 175
Limited Discoverable 367
Limited discoverable mode 185
limited discoverable mode 30
limited inquiry 38
Link 52
Link establishment 54
link establishment 45
link level authentication 365
low power mode 176

M

ME 434
Messaging 344
Messaging application 403
Modem Status Command 178
MTU sizes 180
Multi-terminal support 107

N

Name discovery 54
name discovery 40
non-bondable mode 32
non-connectable mode 31
non-discoverable mode 29
non-pairable mode 32
Notes 345
Notes application 403

O

OBEX 305
OBEX authentication 315, 365, 375, 407
OBEX Connect 376
OBEX connection 375
OBEX Headers 348
OBEX headers 348, 349
OBEX operation 314
OBEX Operations 348
OBEX operations 348
Object Exchange mode 340
Object Push 334, 344
Object Push function 340
On hook 107, 147
Outgoing audio connection 202
Outgoing external call 107
owner’s business card 346

P

paging 185
pairable mode 32
Phone Book 344
Phone Book application 403
PIM 392
Post-dialling 107
PPP 269, 281
Push Client 339
Push Server 339
PUT-operation 321

Q

Quality of Service 180

R
Register recall 107
Remote audio volume control 205
Remote Line Status indication command 178
Remote Port Negotiation Command 178
RFCOMM Server channel 174
RFCOMM session 174
RS232 control signalling 172
RS232 control signals 178

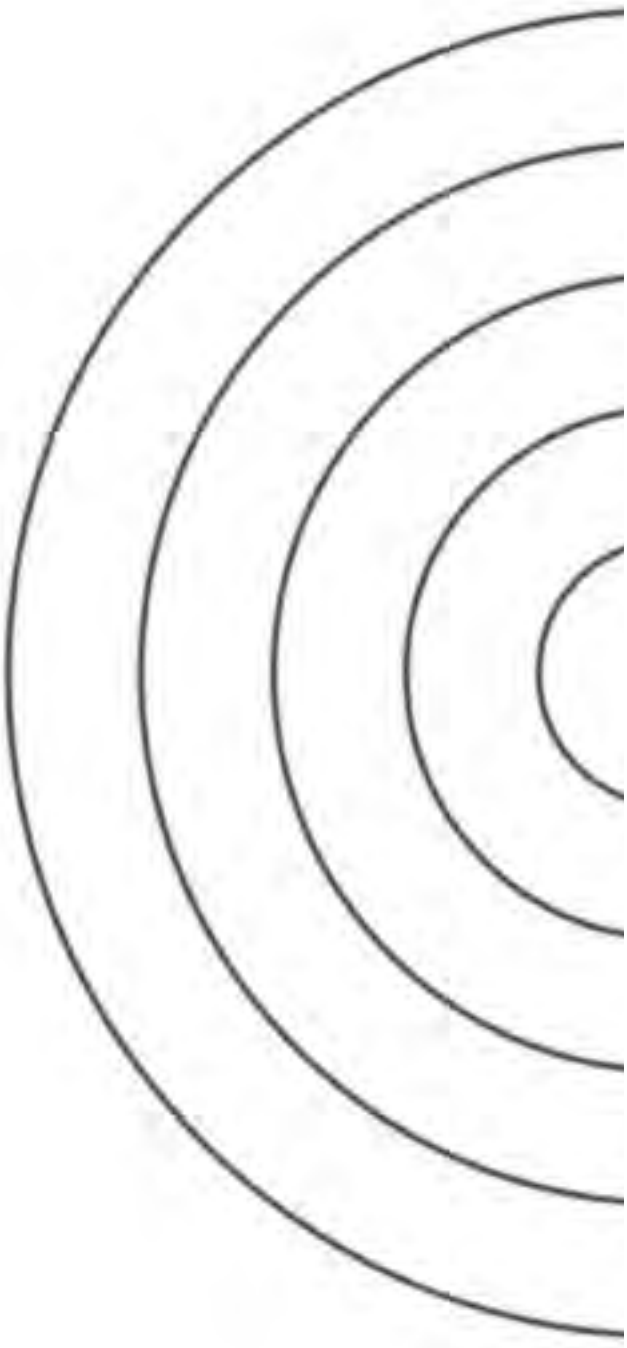
S
SDP database 175
security features 173
security mode 34
serial port 171
Server 310
Service Class ID 174
service record 181
Sync Command 402
Synchronization 391
Synchronization profile 392

T
Target header 375
Terminal 104
TL 104

U
UUID 383

V
virtual serial cable 175
virtual serial port 172

W
walkie-talkie 143
Wide Area Networks 246



https://web.archive.org/web/20030419213004/https://www.irda.org/standards/pubs/OBEX13.pdf

Infrared Data Association®

(IrDA®) Object Exchange Protocol

OBEX™

Extended Systems, Inc
Microsoft Corporation

January 3, 2003

Version 1.3

Copyright ©, 2002 Infrared Data Association ©

Authors:
Pat Megowan, Dave Suvak, Doug Kogan (Extended Systems, Inc.)

Contributors:
Wassef Haroun, Bei-jing Guo, Cliff Strom (Microsoft)
Kerry Lynn (Apple)
Brian McBride, Stuart Williams (Hewlett Packard)
Petri Nykanen (Nokia)
Deepak Amin (Indicus)
Kevin Hendrix (Extended Systems)

Editors:
Doug Kogan, Kevin Hendrix (Extended Systems)

Document Status: Version 1.3
Major changes from Version 1.2 to 1.3 :

- Incorporate Errata from October 1999, January, April and July 2000 IrDA meetings.
- Incorporate Errata from January 2001 IrDA meeting.
- Incorporate Errata from November and December 2002.
- Incorporate *Session Capabilities in OBEX*™ version 0.14 document.
- Removal of the OBEX Test Specification section in favor of the complete *OBEX Test Specification* version 1.0 document.

Major changes from Version 1.1 draft to 1.2:

- Incorporate the OBEX Errata approved at the January 1999 IrDA meeting.
- Incorporate the OBEX Errata approved in March of 1999.

Major changes from Version 1.0 to 1.1 draft:

- Incorporate the “OBEX Errata v.3” approved at the July 1997 IrDA meeting.
- Incorporate the OBEX Errata approved at the October 1997 IrDA meeting.
- Incorporate the “IrOBEX Test Guidelines” approved at the October 1998 IrDA meeting.

INFRARED DATA ASSOCIATION (IrDA) - NOTICE TO THE TRADE -

SUMMARY:

Following is the notice of conditions and understandings upon which this document is made available to members and non-members of the Infrared Data Association.

- Availability of Publications, Updates and Notices
- Full Copyright Claims Must be Honored
- Controlled Distribution Privileges for IrDA Members Only
- Trademarks of IrDA - Prohibitions and Authorized Use
- No Representation of Third Party Rights
- Limitation of Liability
- Disclaimer of Warranty
- Certification of Products Requires Specific Authorization from IrDA after Product Testing for IrDA Specification Conformance

IrDA PUBLICATIONS and UPDATES:

IrDA publications, including notifications, updates, and revisions, are accessed electronically by IrDA members in good standing during the course of each year as a benefit of annual IrDA membership. Electronic copies are available to the public on the IrDA web site located at irda.org. IrDA publications are available to non-IrDA members for a pre-paid fee. Requests for publications, membership applications or more information should be addressed to: Infrared Data Association, P.O. Box 3883, Walnut Creek, California, U.S.A. 94598; or e-mail address: info@irda.org; or by calling Lawrence Faulkner at (925) 944-2930 or faxing requests to (925) 943-5600.

COPYRIGHT:

1. Prohibitions: IrDA claims copyright in all IrDA publications. Any unauthorized reproduction, distribution, display or modification, in whole or in part, is strictly prohibited.

2. Authorized Use: Any authorized use of IrDA publications (in whole or in part) is under NONEXCLUSIVE USE LICENSE ONLY. No rights to sublicense, assign or transfer the license are granted and any attempt to do so is void.

DISTRIBUTION PRIVILEGES for IrDA MEMBERS ONLY:

IrDA Members Limited Reproduction and Distribution Privilege: A limited privilege of reproduction and distribution of IrDA copyrighted publications is granted to IrDA members in good standing and for sole purpose of reasonable reproduction and distribution to non-IrDA members who are engaged by contract with an IrDA member for the development of IrDA certified products. Reproduction and distribution by the non-IrDA member is strictly prohibited.

TRANSACTION NOTICE to IrDA MEMBERS ONLY:

Each and every copy made for distribution under the limited reproduction and distribution privilege shall be conspicuously marked with the name of the IrDA member and the name of the receiving party. Upon reproduction for distribution, the distributing IrDA member shall promptly notify IrDA (in writing or by e-mail) of the identity of the receiving party.

A failure to comply with the notification requirement to IrDA shall render the reproduction and distribution unauthorized and IrDA may take appropriate action to enforce its copyright, including but not limited to, the termination of the limited reproduction and distribution privilege and IrDA membership of the non-complying member.

TRADEMARKS:

1. Prohibitions: IrDA claims exclusive rights in its trade names, trademarks, service marks, collective membership marks and certification marks (hereinafter collectively "trademarks"), including but not limited to the following trademarks: INFRARED DATA ASSOCIATION (wordmark alone and with IR logo), IrDA (acronym mark alone and with IR logo), IR logo, IR DATA CERTIFIED (composite mark), and MEMBER IrDA (wordmark alone and with IR logo). Any unauthorized use of IrDA trademarks is strictly prohibited.

2. Authorized Use: Any authorized use of a IrDA collective membership mark or certification mark is by NONEXCLUSIVE USE LICENSE ONLY. No rights to sublicense, assign or transfer the license are granted and any attempt to do so is void.

NO REPRESENTATION of THIRD PARTY RIGHTS:

IrDA makes no representation or warranty whatsoever with regard to IrDA member or third party ownership, licensing or infringement/non-infringement of intellectual property rights. Each recipient of IrDA publications, whether or not an IrDA member, should seek the independent advice of legal counsel with regard to any possible violation of third party rights arising out of the use, attempted use, reproduction, distribution or public display of IrDA publications.

IrDA assumes no obligation or responsibility whatsoever to advise its members or non-members who receive or are about to receive IrDA publications of the chance of infringement or violation of any right of an IrDA member or third party arising out of the use, attempted use, reproduction, distribution or display of IrDA publications.

LIMITATION of LIABILITY:

BY ANY ACTUAL OR ATTEMPTED USE, REPRODUCTION, DISTRIBUTION OR PUBLIC DISPLAY OF ANY IrDA PUBLICATION, ANY PARTICIPANT IN SUCH REAL OR ATTEMPTED ACTS, WHETHER OR NOT A MEMBER OF IrDA, AGREES TO ASSUME ANY AND ALL RISK ASSOCIATED WITH SUCH ACTS, INCLUDING BUT NOT LIMITED TO LOST PROFITS, LOST SAVINGS, OR OTHER CONSEQUENTIAL, SPECIAL, INCIDENTAL OR PUNITIVE DAMAGES. IrDA SHALL HAVE NO LIABILITY WHATSOEVER FOR SUCH ACTS NOR FOR THE CONTENT, ACCURACY OR LEVEL OF ISSUE OF AN IrDA PUBLICATION.

DISCLAIMER of WARRANTY:

All IrDA publications are provided "AS IS" and without warranty of any kind. IrDA (and each of its members, wholly and collectively, hereinafter "IrDA") EXPRESSLY DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND WARRANTY OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. IrDA DOES NOT WARRANT THAT ITS PUBLICATIONS WILL MEET YOUR REQUIREMENTS OR THAT ANY USE OF A PUBLICATION WILL BE UN-INTERRUPTED OR ERROR FREE, OR THAT DEFECTS WILL BE CORRECTED. FURTHERMORE, IrDA DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING USE OR THE RESULTS OR THE USE OF IrDA PUBLICATIONS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN PUBLICATION OR ADVICE OF A REPRESENTATIVE (OR MEMBER) OF IrDA SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY.

LIMITED MEDIA WARRANTY:

IrDA warrants ONLY the media upon which any publication is recorded to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of distribution as evidenced by the distribution records of IrDA. IrDA's entire liability and recipient's exclusive remedy will be replacement of the media not meeting this limited warranty and which is returned to IrDA. IrDA shall have no responsibility to replace media damaged by accident, abuse or misapplication. ANY IMPLIED WARRANTIES ON THE MEDIA, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF DELIVERY. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM PLACE TO PLACE.

CERTIFICATION and GENERAL:

Membership in IrDA or use of IrDA publications does NOT constitute IrDA compliance. It is the sole responsibility of each manufacturer, whether or not an IrDA member, to obtain product compliance in accordance with IrDA rules for compliance.

All rights, prohibitions of right, agreements and terms and conditions regarding use of IrDA publications and IrDA rules for compliance of products are governed by the laws and regulations of the United States. However, each manufacturer is solely responsible for compliance with the import/export laws of the countries in which they conduct business. The information contained in this document is provided as is and is subject to change without notice.

Contents

1. INTRODUCTION 8

1.1 Tasks, Platforms, and Goals 8
1.2 OBEX components 9
1.2.1 OBEX Session Protocol 9
1.2.2 OBEX Application Framework 10
1.3 Relation to other IrDA protocols 11
1.4 Specification versus Implementation 11
1.4.1 Mapping OBEX packets to TinyTP/IrLMP packets 11
1.5 References 12

2. OBEX OBJECT MODEL 13

2.1 OBEX Headers 13
2.2 Header descriptions 14
2.2.1 Count 14
2.2.2 Name 14
2.2.3 Type 15
2.2.4 Length 15
2.2.5 Time 15
2.2.6 Description 16
2.2.7 Target 16
2.2.8 HTTP 16
2.2.9 Body, End-of-Body 16
2.2.10 Who 17
2.2.11 Connection Identifier 17
2.2.12 Application Request-Response Parameters 17
2.2.13 Authenticate Challenge 18
2.2.14 Authenticate Response 18
2.2.15 Creator ID 18
2.2.16 WAN UUID 18
2.2.17 Object Class 19
2.2.18 Session-Parameters 19
2.2.19 Session-Sequence-Number 20
2.2.20 User Defined Headers 21

3. SESSION PROTOCOL 22

3.1 Request format 23
3.2 Response format 23
3.2.1 Response Code values 24
3.3 OBEX Operations and Opcode definitions 25
3.3.1 Connect 25
3.3.2 Disconnect 28
3.3.3 Put 29
3.3.4 Get 31
3.3.5 Abort 32
3.3.6 SetPath 32
3.3.7 Session 33
3.4 Packet Timeouts 37
3.5 Authentication Procedure 37
3.5.1 Digest Challenge 38
3.5.2 Digest Response 40
3.5.3 Hashing Function 40
3.5.4 Authentication Examples 41
3.6 Multiplexing with OBEX 42

3.6.1 Connections multiplexed at the OBEX Command level 42
3.6.2 Connections multiplexed at the OBEX Transport layer 43

4. OBEX APPLICATION FRAMEWORK 44

4.1 The Default OBEX Server 44
4.2 The Inbox Service 44
4.3 Inbox Connection 45
4.4 Capability Service 45
4.5 Custom OBEX applications 45
4.6 Directed Operations and Connections 46
4.6.1 Directed Connection Mechanics 46
4.6.2 Target Header Processing 46
4.7 OBEX Access Methods 47
4.7.1 File Access 47
4.7.2 Database Access 47
4.7.3 Process / RPC 47

5. USING OBEX OVER IRDA ULTRA-LITE (CONNECTIONLESS USE) 48

6. IRDA OBEX IAS ENTRIES, SERVICE HINT BIT AND TCP PORT NUMBER 49

6.1 IAS entry 49
6.1.1 IrDA:TinyTP:LsapSel 49
6.2 Service Hint bits 49
6.3 TCP port number 49

7. OBEX EXAMPLES 50

7.1 Simple Put - file/note/ecard transfer 50
7.2 Simple Get - field data collection 51
7.3 Example Get of the Capability Object 52
7.4 Connect using Target, Who and Connection Id headers 54
7.5 Combined Get and Put - paying for the groceries 55
7.6 Create Session Followed by a Directed Connection 56
7.7 Suspend a Session 57
7.8 Resume a Session 57
7.9 Close a Session 58
7.10 Negotiate a Timeout 59

8. OBEX SERVICES AND PROCEDURES 60

8.1 Folder Browsing Service 60
8.1.1 Exchanging Folder Listings 60
8.1.2 Navigating Folders 61
8.1.3 Security 61
8.2 Simple OBEX Put file transfer (+ SetPath) 62
8.3 Telecom/IrMC Synchronization Service 62
8.4 OBEX Get Default Object 62
8.4.1 Get default vCard example 63
8.5 Capability Service 63
8.5.1 The Capability Object/Database 63
8.5.2 The Object Profile Database 64
8.5.3 Locating the Capability Service 64

9. OBEX OBJECTS 67

9.1 The Folder Listing Object 67
9.1.1 Element Specification 67
9.1.2 Folder Listing Details 70

9.1.3 Encoding Folder Listing Objects 71
9.1.4 XML Document Definition 72
9.2 Generic File Object 73
9.2.1 Introduction 73
9.2.2 Commonly Used Headers 73
9.2.3 Response Codes Commonly Used in File Exchange 73
9.2.4 Example Put Exchange 74
9.3 The Capability Object 74
9.3.1 The Capability Container (Capability) 75
9.3.2 General Information Container (General) 75
9.3.3 Inbox Container (Inbox) 77
9.3.4 Service/Application List 77
9.3.5 Requesting the Capability Object 78
9.3.6 Capability Object DTD 78
9.3.7 Capability Object Example 79
9.4 The Object Profile Object 81
9.4.1 Creating an Object Profile 81
9.4.2 Object Profiles 81
9.4.3 Object Profile Example 82

10. SESSION CAPABILITIES IN OBEX 83

10.1 OBEX Session Overview 83
10.2 Reliable OBEX Session Overview 84
10.3 Session Context 86
10.4 Persistence 86
10.5 Multiple Transport Support Proposal 86

11. OBEX TESTING REQUIREMENTS 88

11.1 Symbols and Conventions 88
11.2 OBEX Roles 88
11.3 IrDA-Specific Features 88
11.4 OBEX Features 88
11.5 Required Behavior 88

12. APPENDICES 90

12.1 Minimum level of service 90
12.2 Extending OBEX 90
12.3 Proposed Additions to OBEX 90
12.4 Known Target Identifiers 90
12.5 MD5 Algorithm for Authentication 90

1. Introduction

1.1 Tasks, Platforms, and Goals

One of the most basic and desirable uses of the IrDA infrared communication protocols is simply to send an arbitrary “thing”, or data object, from one device to another, and to make it easy for both application developers and users to do so. We refer to this as object exchange (un-capitalized), and it is the subject of the protocol described in this document.

This document describes the current status of the protocol OBEX (for IrDA Object Exchange, OBEX for short). OBEX is a compact, efficient, binary protocol that enables a wide range of devices to exchange data in a simple and spontaneous manner. OBEX is being defined by members of the Infrared Data Association to interconnect the full range of devices that support IrDA protocols. It is not, however, limited to use in an IrDA environment.

OBEX performs a function similar to HTTP, a major protocol underlying the World Wide Web. However, OBEX works for the many very useful devices that cannot afford the substantial resources required for an HTTP server, and it also targets devices with different usage models from the Web. OBEX is enough like HTTP to serve as a compact final hop to a device “not quite” on the Web.

A major use of OBEX is a “Push” or “Pull” application, allowing rapid and ubiquitous communications among portable devices or in dynamic environments. For instance, a laptop user pushes a file to another laptop or PDA; an industrial computer pulls status and diagnostic information from a piece of factory floor machinery; a digital camera pushes its pictures into a film development kiosk, or if lost can be queried (pulled) for the electronic business card of its owner. However, OBEX is not limited to quick connect-transfer-disconnect scenarios - it also allows sessions in which transfers take place over a period of time, maintaining the connection even when it is idle.

PCs, pagers, PDAs, phones, printers, cameras, auto-tellers, information kiosks, calculators, data collection devices, watches, home electronics, industrial machinery, medical instruments, automobiles, and office equipment are all candidates for using OBEX. To support this wide variety of platforms, OBEX is designed to transfer flexibly defined “objects”; for example, files, diagnostic information, electronic business cards, bank account balances, electrocardiogram strips, or itemized receipts at the grocery store. “Object” has no lofty technical meaning here; it is intended to convey flexibility in what information can be transferred. OBEX can also be used for Command and Control functions - directives to TVs, VCRs, overhead projectors, computers, and machinery. Finally, OBEX can be used to perform complex tasks such as database transactions and synchronization.

OBEX is designed to fulfill the following major goals:

1. Application friendly - provide the key tools for rapid development of applications.
2. Compact - minimum strain on resources of small devices.
3. Cross platform.
4. Flexible data handling, including data typing and support for standardized types - this will allow devices to be simpler to use via more intelligent handling of data inside.
5. Maps easily into Internet data transfer protocols.
6. Extensible - provide growth path to future needs like security, compression, and other extended features without burdening more constrained implementations.
7. Testable and Debuggable.

1.2 OBEX components

The OBEX specification consists of two major parts: a protocol and an application framework. The OBEX protocol is a session level protocol that specifies the structure for the conversation between devices. It also contains a model for representing objects. The OBEX application framework is built on top of the OBEX protocol. Its main purpose is to facilitate interoperability between devices using the OBEX protocol. Both of these are discussed in more detail below.

1.2.1 OBEX Session Protocol

The OBEX protocol consists of two major elements: a model for representing objects (and information that describes the objects), and a session protocol to provide a structure for the “conversation” between devices. OBEX is a protocol for sending or exchanging objects and control information. In its simplest form, it is quite compact and requires a small amount of code to implement. It can reside on top of any reliable transport, such as that provided by IrDA Tiny TP [IRDATTP] (including IrDA Lite implementations), or TCP/IP stream sockets. OBEX consists of the following pieces:

- An object model that carries information *about* the objects being sent, as well as containing the objects themselves. The object model is built entirely with parsable headers, similar in concept to the headers in HTTP.

- A session protocol, which structures the dialogue between two devices. The session protocol uses a binary packet-based client/server request-response model.

- An IAS definition and hint bits for the service.

1.2.2 OBEX Application Framework

The OBEX application framework is necessary to ensure interoperability between devices using OBEX. It puts a structure on top of the OBEX protocol. The application framework is the foundation for a set of standard OBEX services that satisfy many object exchange requirements. OBEX implementations are not required to follow the conventions specified by the application framework but doing so will ensure interoperability with other devices. The table below outlines the elements of the OBEX application framework.

Element	Description
OBEX Client	An OBEX Client is the entity that initiates the underlying transport connection to an OBEX server and initiates OBEX operations.
OBEX Server	An OBEX Server is the entity that responds to OBEX operations. The OBEX server waits for the OBEX client to initiate the underlying transport connection.
Default OBEX Server	The Default OBEX server is the server that resides at the LSAP-Sel specified in the OBEX IAS definition. Other OBEX servers can exist but the Default OBEX server is the “well known” server. This is analogous to the HTTP server located at TCP port number 80.
OBEX Transport Connection	The OBEX transport connection is the underlying transport connection, which carries the OBEX protocol.
OBEX Connection	An OBEX Connection is a virtual binding between two applications or services. An OBEX connection is initiated by sending an OBEX **CONNECT** packet. Once a connection is established all operations sent over the connection are interpreted in a continuous context.
Directed Connection	A directed connection is one where the OBEX **CONNECT** packet contains targeting information which the OBEX protocol uses to connect the client to its intended service or application.
The Inbox Connection	The inbox connection is the OBEX connection made to the default OBEX server, where the OBEX **CONNECT** packet does not contain targeting information. A number of services can be accessed via the inbox connection. These services are described later.
Inbox	The inbox is the intended recipient of a client push operation over the Inbox Connection. The inbox does not have to be an actual storage location. It is really a method for encapsulating the concept that the client pushes an object to a recipient without the need to understand the details of how the recipient stores the object.
Application	An OBEX application communicates using a proprietary method known only by the manufacturer. Such applications can only expect to be understood by exact peers. Alternatively, an application may be a service with proprietary extensions. In this case the application must know if it is communicating with a service or application peer.
Service	An OBEX service communicates using procedures specified in a publicly available standard. Such as in IrMC or this specification.
Capability Service	The capability service is used to find information about the OBEX server including device information, types of objects supported, object profiles and supported applications.

1.3 Relation to other IrDA protocols

The following figure illustrates where OBEX fits into the overall scheme of IrDA software.

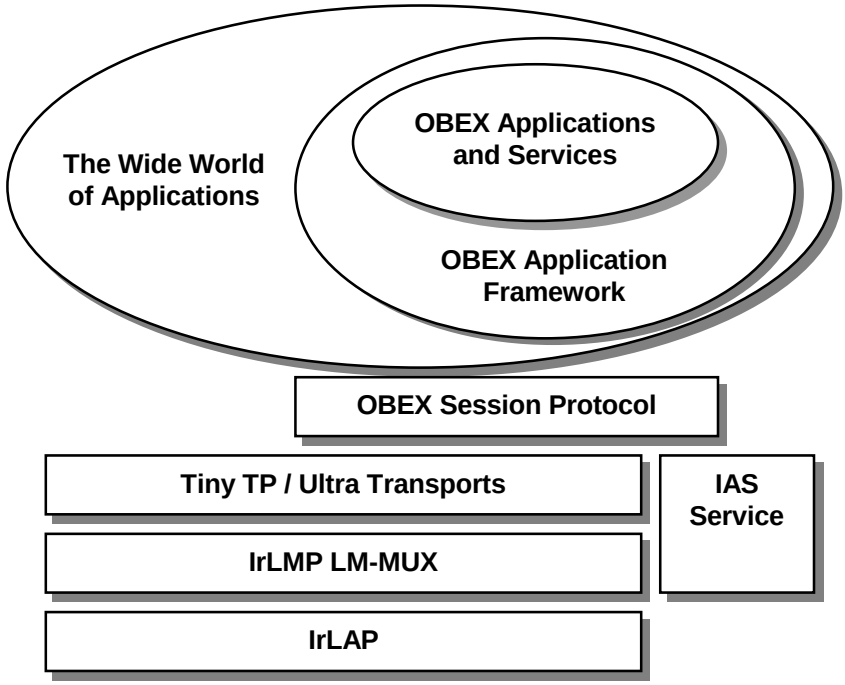


Figure 1. OBEX in the IrDA architecture

The above figure places OBEX within the IrDA protocol hierarchy, but it could just as well appear above some other transport layer that provides reliable flow-controlled connections. Connection oriented operation over IrDA protocols uses Tiny TP flow control to allow for multiple logical connections, including simultaneous OBEX client/server sessions in both directions.

1.4 Specification versus Implementation

This description does not specify any implementation of the protocol; only the requirements that the implementation must embody. In particular, this specification does not specify APIs at either the top or bottom boundaries. The primary OBEX context assumes a lower protocol layer that supports reliable data interchange with other devices (as provided by [IRDATTP]).

1.4.1 Mapping OBEX packets to TinyTP/IrLMP packets

There is no requirement on the alignment of OBEX packets within TinyTP or IrLMP PDUs (packets). Except that all OBEX data shall be carried in data packets, not in TinyTP/IrLMP Connect or Disconnect packets.

OBEX specifically does not use Tiny TP segmentation and reassembly. There is no relationship between OBEX packets and TTP-SDUs. Therefore, the TinyTP Connect packet should not use the MaxSduSize parameter.

1.5 References

IRDALAP	Serial Infrared Link Access Protocol, IrLAP, Version 1.1, Infrared Data Association
IRDALMP	Link Management Protocol, IrLMP, Version 1.1, Infrared Data Association
IRDACOM	Serial and parallel port emulation, IrCOMM, Version 1.0, Infrared Data Association
IRDATTP	Tiny Transport Protocol, TinyTP, Version 1.1, Infrared Data Association
IRDAIAS	IrLMP Hint Bit Assignments and Known IAS Definitions, Ver 1.0, IrDA
HTTP1.1	HTTP v1.1, HTTP 1.1 working group
IANAREG	IANA media type registry
MIME	Multipurpose Internet Mail Extensions
IRMC	Infrared Data Association Specifications for Ir Mobile Communications (IrMC) Version 1.1
DATES	ISO 8601:1988, Data elements and interchange formats - Information interchange - Representation of dates and times
LANG	ISO 639:1988 (E/F), Code for the representation of names of languages

2. OBEX Object Model

The object model addresses the question of how objects are represented by OBEX. The model must deal with both the object being transferred and information *about* the object. It does this by putting the pieces together in a sequence of **headers**. A header is an entity that describes some aspect of the object, such as name, length, descriptive text, or the object body itself. For instance, headers for the file jumar.txt might contain the name, a type identifier of “text”, a description saying “How to use jumars to grow better tomatoes”, the file length, and the file itself.

2.1 OBEX Headers

Headers have the general form:

<HI, the header ID>
<HV, the header value>

HI, the header ID, is an unsigned one-byte quantity that identifies what the header contains and how it is formatted. HV consists of one or more bytes in the format and meaning specified by HI. All headers are optional - depending on the type of device and the nature of the transaction, you may use all of the headers, some, or none at all. IDs make headers parseable and order independent, and allow unrecognized headers to be skipped easily. Unrecognized headers should be skipped by the receiving device.

OBEX defines a set of headers that are used quite frequently and therefore benefit from a compact representation. It also provides a mechanism to use any HTTP header, as well as user defined headers. This small set will serve the needs of file transfer on PCs as well as the needs of many other devices, while facilitating a clean mapping to HTTP when OBEX is used as a compact final hop in a web communication.

The low order 6 bits of the header identifier are used to indicate the meaning of the header, while the upper 2 bits are used to indicate the header encoding. This encoding provides a way to interpret unrecognized headers just well enough to discard them cleanly. The length prefixed header encodings send the length in network byte order, and the length includes the 3 bytes of the identifier and length. For Unicode text, the length field (immediately following the header ID) includes the 2 bytes of the null terminator (0x00, 0x00). Therefore the length of the string ”Jumar” would be 12 bytes; 5 visible characters plus the null terminator, each two bytes in length.

The 2 high order bits of HI have the following meanings (shown both as bits and as a byte value):

Bits 8 and 7 of HI	Interpretation
00 (0X00)	null terminated Unicode text, length prefixed with 2 byte unsigned integer
01 (0X40)	byte sequence, length prefixed with 2 byte unsigned integer
10 (0X80)	1 byte quantity
11 (0XC0)	4 byte quantity – transmitted in network byte order (high byte first)

The Header identifiers are:

HI - identifier	Header name	Description
0xC0	Count	Number of objects (used by Connect)
0x01	Name	name of the object (often a file name)
0x42	Type	type of object - e.g. text, html, binary, manufacturer specific
0xC3	Length	the length of the object in bytes
0x44 0xC4	Time	date/time stamp – ISO 8601 version - preferred date/time stamp – 4 byte version (for compatibility only)
0x05	Description	text description of the object
0x46	Target	name of service that operation is targeted to
0x47	HTTP	an HTTP 1.x header
0x48 0x49	Body End of Body	a chunk of the object body. the final chunk of the object body
0x4A	Who	identifies the OBEX application, used to tell if talking to a peer
0xCB	Connection Id	an identifier used for OBEX connection multiplexing
0x4C	App. Parameters	extended application request & response information
0x4D	Auth. Challenge	authentication digest-challenge
0x4E	Auth. Response	authentication digest-response
0xCF	Creator ID	indicates the creator of an object
0x50	WAN UUID	uniquely identifies the network client (OBEX server)
0x51	Object Class	OBEX Object class of object
0x52	Session-Parameters	Parameters used in session commands/responses
0x93	Session-Sequence-Number	Sequence number used in each OBEX packet for reliability
0x14 to 0x2F	Reserved for future use	this range includes all combinations of the upper 2 bits
0x30 to 0x3F	User defined	this range includes all combinations of the upper 2 bits

All allowable headers and formats thereof are listed by this table. Applications must not change the upper bits on OBEX defined headers and expect anyone else to recognize them. Note that the header identifiers are numbered in order, starting with zero. The high order bits which specify the encoding obscure this linear sequence of header numbering.

Certain headers like **Body** are expected to be present repeatedly, however headers like **Name** and **Time** do not necessarily make sense when sent multiple times. The behavior by the recipient of multiple non-**Body** headers is not defined by the protocol.

All Unicode headers are encoded using UTF-16 format with the bytes sent in network order.

2.2 Header descriptions

2.2.1 Count

Count is a four byte unsigned integer to indicate the number of objects involved in the operation.

2.2.2 Name

Name is a null terminated Unicode text string describing the name of the object.

Example: JUMAR.TXT

Though the **Name** header is very useful for operations like file transfer, it is optional - the receiving application may know what to do with the object based on context, **Type**, or other factors. If the object is being sent to a general purpose device such as a PC or PDA, this will normally be used as the filename of the received object, so act accordingly. Receivers that interpret this header as a file name must be prepared to handle Names that are not legitimate filenames on their system. In some cases an empty **Name** header is used to specify a particular behavior; see the **GET** and **SETPATH** Operations. An empty **Name** header is defined as a **Name** header of length 3 (one byte opcode + two byte length).

2.2.3 Type

Type is a byte sequence consisting of null terminated ASCII text describing the type of the object, such as text, binary, or vCard. **Type** is used by the receiving side to aid in intelligent handling of the object. This header corresponds to the content-type header in HTTP.

Whenever possible, OBEX (like HTTP) uses IANA registered media types to promote interoperability based on open standards. When a registered type is used, the HTTP canonical form for the object body must also be used. In other words, if you say a thing is of type “text/html”, it must meet all the rules for representing items of type “text/html”. OBEX follows RFC 1521 which defines the media type format and handles **Type** header values are case insensitive values. See the following URL for a list of MIME defined media types: http://www.isi.edu/in-notes/iana/assignments/media-types.

If no **Type** is specified, the assumed type is binary, and it is up to the receiving software to deal with it as best it can. This may involve simply storing it without modification of any kind under the stated name, and/or trying to recognize it by the extension on the name. For instance, a Microsoft Word file could be sent with no type, and the receiving software, seeing the .doc suffix could choose to interpret it as a Word file.

Though the **Type** header is very useful for transfer of non-file object types, it is optional - the receiving application may know what to do with the object based on context, name, or other factors.

2.2.4 Length

Length is a four byte unsigned integer quantity giving the total length in bytes of the object. If the Length is known in advance, this header should be used. This allows the receiver to quickly terminate transfers requiring too much space, and also makes progress reporting easier.

If a single object exceeds 4 gigabytes - 1 in length, its size cannot be represented by this header. Instead an **HTTP** content-length header should be used, which is ASCII encoded decimal and can represent arbitrarily large values. However, implementations that cannot handle such large objects are not required to recognize the **HTTP** header.

The **Length** header is optional, because in some cases, the length is not known in advance, and the **End-of-Body** header will signal when the end of the object is reached.

2.2.5 Time

Time is a byte sequence that gives the object’s UTC date/time of last modification in ISO 8601 format. Local times should be represented in the format YYYYMMDDTHHMMSS and UTC time in the format YYYYMMDDTHHMMSSZ. The letter “T” delimits the date from the time. UTC time is identified by concatenating a “Z” to the end of the sequence. When possible UTC times should be used. This header is encoded as a US-ASCII string. The Date/**Time** header is optional.

Note: One notable OBEX application was released before the standard became final, and uses an unsigned 4 byte integer giving the date/time of the object’s last modification in seconds since January 1, 1970. Implementers may wish to accept or send both formats for backward compatibility, but it is not required. The preferred ISO 8601 format and this format can be distinguished by the high two bits of the

Header Identifier—ISO 8601 uses the text HI encoding 0x44, while this one uses the 4 byte integer HI encoding 0xC4.

2.2.6 Description

Description is a null terminated Unicode text string used to provide additional description of the object or operation. The **Description** header is optional. The **Description** header is not limited to describing objects. For example, it may accompany a response code to provide additional information about the response.

2.2.7 Target

Target is a byte sequence that identifies the intended target of the operation. On the receiving end, object name and type information provide one way of performing dispatching - this header offers an alternate way of directing an operation to the intended recipient.

The **Target** headers most common use, is when sent in an OBEX **CONNECT** packet to initiate a directed connection to an OBEX server (see section 4.6). A list of well-known **Target** header values is contained in the appendix of this specification. The **Target** header is commonly used in conjunction with the **Who** and **Connection Id** headers when establishing a directed connection.

When used with the **PUT** operation, it allows for behavior analogous to the HTTP POST operation. Wherein the **Target** specifies the service that should process the object contained in the **PUT** request.

The sending device must provide this header in a form meaningful to the destination device. If this header is received but not recognized, it is up to the implementation to decide whether to accept or reject the accompanying object. When used, the **Target** header must be the first header in the operation.

To work effectively in a broad application environment it is necessary that the **Target** header identify a universally unique service or client. It is recommended that 128-bit UUID's be used to fulfill this requirement. Since the **Target** header is a binary header type, values are compared in a case sensitive manner. Therefore, care must be taken to maintain the proper case when using ASCII values.

It is illegal to send a **Connection Id** and a **Target** header in the same request. It is legal to send both headers in a response, as discussed in section 3.3.1.6.

2.2.8 HTTP

HTTP is a byte sequence containing an HTTP 1.x header. This can be used to include many advanced features already defined in HTTP without re-inventing the wheel. HTTP terminates lines using CRLF, which will be preserved in this header so it can be passed directly to standard HTTP parsing routines. This header is optional.

2.2.9 Body, End-of-Body

The body of an object (the contents of a file being transferred, for instance) is sent in one or more **Body** headers. A "chunked" encoding helps make abort handling easier, allows for operations to be interleaved, and handles situations where the length is not known in advance, as with process generated data and on-the-fly encoding.

A **Body** header consists of the HI (identifying it as an object body), a two byte header length, and all or part of the contents of the object itself.

A distinct HI value (**End-of-Body**) is used to identify the last chunk of the object body. In some cases, the object body data is generated on the fly and the end cannot be anticipated, so it is legal to send a zero

length **End-of-Body** header. The **End-of-Body** header signals the end of an object, and must be the final header of any type associated with that object.

2.2.10 Who

Who is a length prefixed byte sequence used so that peer applications may identify each other, typically to allow special additional capabilities unique to that application or class of device to come into play.

The **Who** header is typically used in an OBEX **CONNECT** response packet to indicate the UUID of the service which has accepted the directed connection. The value of the **Who** header matches the value of the **Target** header sent in the **CONNECT** command.

To work effectively in a broad application environment it is necessary that the **Who** header identify a universally unique service or client. It is recommended that 128-bit UUID's be used to fulfill this requirement.

2.2.11 Connection Identifier

Connection Id is a 4-byte value that tells the recipient of the request which OBEX connection this request belongs to. The **Connection Id** header is optional. When in use, the **Connection Id** header *must* be the first header in the request.

When it is desirable to provide concurrent access to multiple OBEX services over the same TinyTP connection, the **Connection Id** is used to differentiate between multiple clients. This is often the case when multiple services are accessed via the default OBEX server. The server can identify the need for a connection to be assigned, by the presence of a **Target** header in the OBEX **CONNECT** packet. The connection identifier is returned to the client in the OBEX **CONNECT** response packet. This connection identifier must be unique so that services can be uniquely identified. Once a logical OBEX Connection has been established, all further client requests to that service must include the **Connection Id** header. Only the first packet in the request needs to contain the **Connection Id** header. As a result, only one request can be processed at a time.

If a **Connection Id** header is received with an invalid connection identifier, it is recommended that the operation be rejected with the response code (0xD3) "Service Unavailable". Since not all servers will parse this header, there is no guarantee that this response code will be returned in all such cases. For convenience the connection identifier value 0xFFFFFFFF is reserved and is considered invalid.

It does not make sense to send a **Connection Id** header in an OBEX **CONNECT** operation and is therefore forbidden. It is optional to send a **Connection Id** header in an OBEX **ABORT** operation, as discussed in section 3.3.5. It is also illegal to send a **Connection Id** and a **Target** header in the same request.

2.2.12 Application Request-Response Parameters

The **Application Parameters** header is used by applications (and protocols) layered above OBEX to convey additional information in a request or response packet. In a request, this header conveys request parameters or modifiers. In a response, it is used in cases where the simple pass/fail status returned by OBEX is insufficient. In order to support an unbounded set of values, from integer values to whole structures (used in RPC style requests) the Application Parameter header is based on the OBEX Byte-Sequence Header format.

A Tag-Length-Value encoding scheme is used to support a variety of request/response types and levels. An application parameters header may contain more than one tag-length-value triplet. The header format is shown below:

Parameter Triplet 1	Parameter Triplet 2	Parameter Triplet . . .

Tag1	Length	Value	Tag2	Length	Value	Tag	Length	Value

The tag and length fields are each one byte in length. The value field can be from zero to *n* bytes long. The value *n* is constrained by the maximum size of an OBEX header, the length field maximum of 255 bytes and the size of other TLV-triplets encoded in the header.

TAG values are defined by the applications or upper protocol layer that uses them and only need to be unique within the scope of the application or protocol layer.

2.2.13 Authenticate Challenge

The **Authenticate Challenge** header is used by both clients and servers to initiate the authentication of the remote device. This header carries the digest-challenge string. See the Authentication chapter for a detailed explanation of the authentication procedure.

A Tag-Length-Value encoding scheme is used to support the variety of options available for authentication. An **Authenticate Challenge** header may contain more than one tag-length-value triplet. The header format is shown below:

Authentication Triplet 1			**Authentication Triplet 2**			**Authentication Triplet . . .**		
Tag1	Length	Value	Tag2	Length	Value	Tag	Length	Value

The tag and length fields are each one byte in length. The value field can be from zero to *n* bytes long. The value *n* is constrained by the maximum size of an OBEX header, the length field maximum of 255 bytes and the size of other TLV-triplets encoded in the header.

2.2.14 Authenticate Response

The **Authenticate Response** header is used by both clients and servers to respond to an authentication request. This header carries the digest-response string. See the Authentication chapter for a detailed explanation of the authentication procedure.

A Tag-Length-Value encoding scheme is used to support the variety of options available for authentication. An **Authenticate Response** header may contain more than one tag-length-value triplet. The tag format is the same as that defined for the **Authenticate Challenge** header shown above.

2.2.15 Creator ID

The **Creator ID** header is a 4-byte unsigned integer value that identifies the creator of the object. The Creator ID can be used by the receiving side to aid in intelligent handling of the object.

2.2.16 WAN UUID

The **WAN UUID** header is for use when layering OBEX over stateless networks where the roles of network client and server are opposite to those of OBEX client and server (such as in WAP). In such an environment, the network client (the mobile device) is also the OBEX server. Here, the **WAN UUID** is used to provide state because the network client is only recognized by the server for one request/response pair at a time. The **WAN UUID** header is a 16 byte UUID that uniquely identifies the network client (OBEX server) to the network server. The network server provides the UUID to the OBEX server in an OBEX Connect request packet. The OBEX Server then sends the header to the network server in all future OBEX responses. The header type is byte-sequence (0x20).

2.2.17 Object Class

The **Object Class** header is used to reference the object class and properties. It is based on the byte sequence header type. At its lowest level the **Object Class** header works similarly to the OBEX **Type** header except that its namespace is that of OBEX not MIME. For sub-object level access, the header can be detailed enough to express specific fields or subsets of the objects' contents. The values used in **Object Class** headers are limited in scope to the application or services that define them. However, the sharing of Object Classes is encouraged.

2.2.18 Session-Parameters

Session-Parameters is a byte sequence that is required for the **CREATESESSION**, **CLOSESESSION**, **SUSPENDSESSION**, **RESUMESESSION** and **SETTIMEOUT** commands and the responses to these commands. It contains a set of parameters, which are specific to the command in which the header is used. If required, the **Session-Parameters** must be the first header in a **SESSION** command or response. In addition, for **SESSION** commands only, the Session Opcode must be the first tag-length-value triplet within the **Session-Parameters** header.

A Tag-Length-Value encoding scheme is used to support a variety of parameters. A **Session-Parameters** header may contain more than one tag-length-value triplet. The header format is shown below:

Parameter Triplet 1			Parameter Triplet 2			Parameter Triplet . . .		
Tag1	Length	Value	Tag2	Length	Value	Tag	Length	Value

The tag and length fields are each one byte in length. The value field can be from zero to *n* bytes long. The value *n* is constrained by the maximum size of an OBEX Packet, which at this point is 255 bytes, the length field maximum of 255 bytes and the size of other TLV-triplets encoded in the header.

The following TAG values are defined.

Tag Value	Name	Meaning
0x00	Device Address	The *device address* of the device sending the header. If running over IrDA this is the 32-bit device address. For Bluetooth this is the 48-bit Bluetooth address. If Running over TCP/IP this is the IP address.
0x01	Nonce	The nonce is a value provided by the device, which will be used in the creation of the session ID. This number must be unique for each session created by the device. One method for creating the nonce is to start with a random number then increment the value each time a new session is created. The Nonce should be at least 4 bytes and at most 16 bytes in size.
0x02	Session ID	Session ID. This is a 16-byte value, which is formed by taking the device address and nonce from the client and server and running the MD5 algorithm over the resulting string. The Session ID is created as follows: MD5("*Client Device Address*" "*Client Nonce*" "*Server Device Address*" "*Server Nonce*")
0x03	Next Sequence Number	This is a one-byte value sent by the server, which indicates the next sequence number expected when the session is resumed.
0x04	Timeout	This is a 4-byte value that contains the number of seconds a session can be in suspend mode before it is considered closed. The value of 0xffffffff indicates a timeout of infinity. This is the default timeout. If a device does not send a timeout field then it can be assumed that the desired timeout is infinity. The timeout in affect is the smallest timeout sent by the client or server.
0x05	Session Opcode	The session opcode is a 1-byte value sent by the client to indicate the Session command that is to be performed. This tag-length-value is only sent in **SESSION** commands and must be the first tag in the **Session-Parameters** header. The session opcode definitions are defined in the bulleted list below.
0x06 - 0xff		Reserved

Session Opcode Definitions:

- 0x00 Create Session
- 0x01 Close Session
- 0x02 Suspend Session
- 0x03 Resume Session
- 0x04 Set Timeout
- 0x05 – 0xFF Reserved

2.2.19 Session-Sequence-Number

Session-Sequence-Number is a 1-byte quantity containing the current sequence number. Even though OBEX is a command/response protocol and only one outstanding command can exist at any given time, the counting capacity of the sequence number is 256 using digits 0 – 255 (0xFF). At 255 the sequence number wraps around to 0. This adds another level of verification that the correct session is being resumed. If the sequence number is invalid then the session should be aborted. This procedure should be followed both when the session is being resumed and during an active session.

When the session is first created the **Session-Sequence-Number** number is set to 0. The first command packet sent by the client after the session is established will contain a **Session-Sequence-Number** header with value of 0. The response to this packet from the server contains the next packet number expected. If the packet was received successfully then the value of the **Session-Sequence-Number** header in the response would be 1.

The **Session-Sequence-Number** must be the first header. The OBEX specification indicates that the **Connection-ID** header must be the first header so there appears to be conflict. Since the new reliable session features sit below the standard OBEX parser, the reliable session could be considered a lower layer protocol. Thus, session information should be stripped off before being sent up to the OBEX parser. The standard OBEX parser will not see the **Session-Sequence-Number** header and the **Connection ID** header will appear to be the first header when it exists.

Session-Sequence-Number headers are not used in Session commands or responses. Thus, session numbers are not advanced when sending session commands/responses when a reliable session is active.

2.2.20 User Defined Headers

User defined headers allow complete flexibility for the application developer. Observe the use of the high order two bits to specify encoding, so that implementations can skip unrecognized headers. Obviously you cannot count on user defined headers being interpreted correctly except by strict peers of your application. So exercise due care at connect time before relying on these - the **Who** header can be used at connect time to identify strict peers.

3. Session Protocol

The session protocol describes the basic structure of an OBEX conversation. It consists of a format for the “conversation” between devices and a set of opcodes that define specific actions. The OBEX conversation occurs within the context of an OBEX connection. The connection oriented session allows capabilities information to be exchanged just once at the start of the connection, and allows state information to be kept (such as a target path for **PUT** or **GET** operations).

OBEX follows a client/server **request-response** paradigm for the conversation format. The terms client and server refer to the originator/receiver of the OBEX connection, not necessarily who originated the low level IrLAP connection. Requests are issued by the client (the party that initiates the OBEX connection). Once a request is issued, the client waits for a response from the server before issuing another request. The request/response pair is referred to as an **operation**.

In order to maintain good synchronization and make implementation simpler, requests and responses may be broken into multiple OBEX packets that are limited to a size specified at connection time. Each request packet is acknowledged by the server with a response packet. Therefore, an operation normally looks like a sequence of request/response packet pairs, with the final pair specially marked. In general, an operation should not be broken into multiple packets unless it will not fit into a single OBEX packet.

Each Request packet consists of an opcode (such as **PUT** or **GET**), a packet length, and one or more headers, as defined in the object model chapter. A header must entirely fit within a packet - it may not be split over multiple packets. It is strongly recommended that headers containing the information *describing* the object (name, length, date, etc.) be sent before the object body itself. For instance, if a file is being sent, the file name should be sent first so that the file can be created and the receiver can be ready to store the contents as soon as they show up.

However, This does not mean that *all* descriptive headers must precede *any* **Body** header. **Description** headers could come at any time with information to be presented in the user interface, and in the future intermediate headers may be used to distinguish among multiple parts in the object body.

The orderly sequence of **request** (from a client) followed by **response** (from a server) has one exception. An **ABORT** operation may come in the middle of a request/response sequence. It cancels the current operation.

Each side of a communication link may have both client and server if desired, and thereby create a peer to peer relationship between applications by using a pair of OBEX sessions, one in each direction. However, it is not a requirement that a device have a both client and server, or that more than one session be possible at once. For example, a data collection device (say a gas meter on a house) might be a server only, and support only the **GET** operation, allowing the device to deliver its information (the meter reading) on demand. A simple file send applet on a PC might support only **PUT**.

3.1 Request format

Requests consist of one or more packets, each packet consisting of a one byte opcode, a two byte packet length, and required or optional data depending on the operation. Each request packet must be acknowledged by a response. Each opcode is discussed in detail later in this chapter, including the number and composition of packets used in the operation. The general form of a request packet is:

Byte 0	Bytes 1, 2	Bytes 3 to n
opcode	packet length	headers or request data

Every request packet in an operation has the opcode of that operation. The high order bit of the opcode is called the Final bit. It is set to indicate the last packet for the request. For example, a **PUT** operation sending a multi-megabyte object will typically require many **PUT** packets to complete, but only the last packet will have the Final bit set in the **PUT** opcode.

If the operation requires multiple response packets to complete after the Final bit is set in the request (as in the case of a **GET** operation returning an object that is too big to fit in one response packet). The server will signal this with the “Continue” response code, telling the client that it should ask for more. The client (requestor) should send another request packet with the same opcode, Final bit set, and no headers if it wishes to continue receiving the object. If the client does not want to continue the operation, it should send an **ABORT** packet.

As with header lengths, the packet length is transmitted in network byte order (high order byte first), and represents the entire length of the packet including the opcode and length bytes. The maximum packet length is 64K bytes - 1.

3.2 Response format

Responses consist of one or more packets - one per request packet in the operation. Each packet consists of a one byte response code, a two byte packet length, and required or optional data depending on the operation. Each response code is listed later in this chapter, and commonly used codes are discussed in the individual opcode sections. The general form of a request packet is:

Byte 0	Bytes 1,2	Bytes 3 to n
response code	response length	response data

The high order bit of the response code is called the Final bit. In OBEX 1.0 it is always set for response packets. Its meaning is slightly different from the request packet final bit—the response packet final bit tells the other side (the client) that it is once again the client’s turn to send a packet. If the response packet carries a success or failure response code, the client is free to begin a new operation. However, if the response code is “Continue” as is often the case in a lengthy **GET** operation, then the client next issues a continuation of the **GET** request. See the **GET** operation description below for examples.

As with header and request lengths, the response length is transmitted in network byte order (high order byte first), and represents the entire length of the packet including the opcode and length bytes. The maximum response packet length is 64K bytes - 1, and the actual length in a connection is subject to the limitations negotiated in the OBEX **CONNECT** operation.

The (optional) response data may include objects and headers, or other data specific to the request that was made. If a **Description** header follows the response code before any headers with object specific information, it is interpreted as descriptive text expanding on the meaning of the response code. Detailed responses are discussed in the opcode sections below.

The response code contains the HTTP status code (a 3 digit ASCII encoded positive integer) encoded in the low order 7 bits as an unsigned integer (the code in parentheses has the Final bit set). See the HTTP document for complete descriptions of each of these codes. The most commonly used response codes are 0x90 (0x10 Continue with Final bit set, used in responding to non-final request packets), and 0xA0 (0x20 Success w/Final bit set, used at end of successful operation).

3.2.1 Response Code values

OBEX response code	HTTP status code	Definition
0x00 to 0x0F	None	reserved
0x10 (0x90)	100	Continue
0x20 (0xA0)	200	OK, Success
0x21 (0xA1)	201	Created
0x22 (0xA2)	202	Accepted
0x23 (0xA3)	203	Non-Authoritative Information
0x24 (0xA4)	204	No Content
0x25 (0xA5)	205	Reset Content
0x26 (0xA6)	206	Partial Content
0x30 (0xB0)	300	Multiple Choices
0x31 (0xB1)	301	Moved Permanently
0x32 (0xB2)	302	Moved temporarily
0x33 (0xB3)	303	See Other
0x34 (0xB4)	304	Not modified
0x35 (0xB5)	305	Use Proxy
0x40 (0xC0)	400	Bad Request - server couldn't understand request
0x41 (0xC1)	401	Unauthorized
0x42 (0xC2)	402	Payment required
0x43 (0xC3)	403	Forbidden - operation is understood but refused
0x44 (0xC4)	404	Not Found
0x45 (0xC5)	405	Method not allowed
0x46 (0xC6)	406	Not Acceptable
0x47 (0xC7)	407	Proxy Authentication required
0x48 (0xC8)	408	Request Time Out
0x49 (0xC9)	409	Conflict
0x4A (0xCA)	410	Gone
0x4B (0xCB)	411	Length Required
0x4C (0xCC)	412	Precondition failed
0x4D (0xCD)	413	Requested entity too large
0x4E (0xCE)	414	Request URL too large
0x4F (0xCF)	415	Unsupported media type
0x50 (0xD0)	500	Internal Server Error
0x51 (0xD1)	501	Not Implemented
0x52 (0xD2)	502	Bad Gateway
0x53 (0xD3)	503	Service Unavailable
0x54 (0xD4)	504	Gateway Timeout
0x55 (0xD5)	505	HTTP version not supported
0x60 (0xE0)	- - -	Database Full
0x61 (0xE1)	- - -	Database Locked

3.3 OBEX Operations and Opcode definitions

OBEX operations consist of the following:

Opcode (w/high bit set)	Definition	Meaning
0x80 *high bit always set	Connect	choose your partner, negotiate capabilities
0x81 *high bit always set	Disconnect	signal the end of the session
0x02 (0x82)	Put	send an object
0x03 (0x83)	Get	get an object
0x04 (0x84)	Reserved	not to be used w/out extension to this specification
0x85 *high bit always set	SetPath	modifies the current path on the receiving side
0x06 (0x86)	Reserved	not to be used w/out extension to this specification
0x87 *high bit always set	Session	used for reliable session support
0xFF *high bit always set	Abort	abort the current operation
0x08 to 0x0F	Reserved	not to be used w/out extension to this specification
0x10 to 0x1F	User definable	use as you please with peer application
Bits 5 and 6 are reserved and should be set to zero.		
Bit 7 of the opcode means Final packet of request.		

The high bit of the opcode is used as a Final bit, described in the previous sections of this chapter. Bits 5 and 6 of the opcode are reserved for future use and should be set to zero by sender and ignored by receiver. However, one notable exemption from this rule is the **ABORT** opcode, which currently sets bits 5 and 6.

If a server receives an unrecognized opcode, it should return 0xD1 response code (Not Implemented, with Final bit set) and ignore the operation. It may send the request to the bit bucket, save it for later analysis, or whatever it chooses.

Each operation is described in detail in the following sections.

3.3.1 Connect

This operation initiates the connection and sets up the basic expectations of each side of the link. The request format is:

Byte 0	Bytes 1 and 2	Byte 3	Byte 4	Bytes 5 and 6	Byte 7 to n
0x80	connect packet length	OBEX version number	flags	maximum OBEX packet length	optional headers

The response format is:

Byte 0	Bytes 1 and 2	Byte 3	Byte 4	Bytes 5 and 6	Byte 7 to n
response code	connect response packet length	OBEX version number	flags	maximum OBEX packet length	optional headers

The **CONNECT** request and response must each fit in a single packet. Implementations are not *required* to recognize more than the first 7 bytes of these packets, though this may restrict their usage.

3.3.1.1 OBEX version number

The version number is the version of the OBEX protocol encoded with the major number in the high order 4 bits, and the minor version in the low order 4 bits. The protocol version is not always the same as the specification version. The current protocol version is 1.0. See the example later in this section.

3.3.1.2 Connect flags

The flags have the following meanings:

bit	Meaning
0	Response: Indicates support for multiple IrLMP connections to the same LSAP-SEL. Request: reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

All request flags except bit 0 are currently reserved, and must be set to zero on the sending side and ignored by the receiving side. All reserved response bits must also be set to zero and ignored by the receiving side.

In a **CONNECT** Response packet bit 0 is used to indicate the ability of the OBEX server's transport to accept multiple IrLMP connections to the same LSAP-SEL. This capability, usually found in more robust IrDA stacks, allows the server to use the LM-MUX for multiplexing multiple simultaneous client requests. Conveying this information is necessary because the IrDA Lite specification does not support this type of multiplexing and attempts to connect to an already connected LSAP-SEL will result in an IrLAP disconnect.

Bit 0 should be used by the receiving client to decide how to multiplex operations to the server (should it desire to do so). If the bit is 0 the client should serialize the operations over a single TTP connection. If the bit is set the client is free to establish multiple TTP connections to the server and concurrently exchange its objects.

3.3.1.3 Maximum OBEX Packet Length

The maximum OBEX packet length is a two byte unsigned integer that indicates the maximum size OBEX packet that the device can receive. The largest acceptable value at this time is 64K bytes -1. However, even if a large packet size is negotiated, it is not required that large packets be sent - this just represents the maximum allowed by each participant. The client and server may have different maximum lengths.

This is one of the most important features of the **CONNECT** packet because it permits the application to increase the OBEX packet size used during an exchange. The default OBEX packet size of 255 bytes is inefficient for large object transfers and will impede transfer rates. Larger OBEX packet sizes such as 8k to 32k allow large amounts of data to be sent without acknowledgement. However, packet sizes should be intelligently limited on slower links to reduce abort request latency.

3.3.1.4 Minimum OBEX Packet Length

The minimum size of the OBEX Maximum packet length allowed for negotiation is 255 bytes. This is in order to provide a greater likelihood of meeting the requirement for single packet requests and responses in a broad range of cases. In addition, since an OBEX header must fit completely within one OBEX packet it is advantageous to mandate a minimum that allows for reasonable header sizes. Note that

OBEX does not exchange Minimum Packet Length values. This value is the minimum acceptable value for the OBEX Maximum Packet Length parameter exchanged in the **CONNECT** Operation.

3.3.1.5 Using Count and Length headers in Connect

The **Count** and **Length** headers, defined in the Object Model chapter, can be used in **CONNECT**. **Count** is used to indicate the number of objects that will be sent during this connection. **Length** is used to express the approximate total length of the bodies of all the objects in the transaction.

When used in the **CONNECT** Operation, the **Length** header contains the length in bytes of the bodies of all the objects that the sender plans to send. Note that this length cannot be guaranteed correct, so while the value may be useful for status indicators and resource reservations, the receiver should not die if the length is not exact. The receiver can depend on **Body** headers to accurately indicate the size of an object as it is delivered and the **End-of-Body** header to indicate when an object is complete.

3.3.1.6 Using Who and Target headers in Connect

Target and **Who** are used to hold a unique identifier, which allows applications to tell whether they are talking to a strict peer, or not. Typically, this is used to enable additional capabilities supplied only by an exact peer. If a **Who** header is used, it should be sent before any body headers.

On full-featured [PC] platforms, multiple OBEX applications may exist concurrently. This leads to the need for the client to be able to uniquely identify which server it wants to handle its request. The server is therefore identified with the OBEX **Target** header. If necessary, the client can also identify itself, using the OBEX **Who** header. The following text describes the exact uses of these headers.

To target a specific application with OBEX commands the client must set-up a connection to the application by using the OBEX **Target** header in a **CONNECT** request. This type of connection is called a directed connection and provides a virtual binding between the client and server. The **Target** header should specify the UUID of the desired application. The **Who** header can also be used when it is necessary to identify the client initiating the exchange. The **Who** header should be used in cases where the target server application supports different client applications and may care which one it is connecting to. It is unnecessary to send a **Who** header in the request if its only logical value is the same as the **Target** header.

The response to the targeted connect operation should contain a **Who** with the same UUID as sent in the request's matching **Target** header. If the **Who** header was present in the request, a **Target** header identifying the same client should be sent in the response. In addition, a unique connection identifier must be sent in a **Connection Id** header. This connection identifier is used by the client in all future operations. If the response does not contain the correct headers then it should be assumed that the connection has not been made to the specific application but to the inbox service. This will be the response when sending a directed connect to a system that does not parse these (Target) headers. In the event that a connection is made to the inbox service, it is the responsibility of the client application to determine whether to continue the exchange or disconnect.

3.3.1.7 Using Description headers in Connect

The **CONNECT** request or response may include a **Description** header with information about the device or service. It is recommended this information be presented through the user interface on the receiving side if possible.

3.3.1.8 The Connect response

The successful response to **CONNECT** is 0xA0 (Success, with the high bit set) in the response code, followed immediately by the required fields described above, and optionally by other OBEX headers as defined above. Any other response code indicates a failure to make an OBEX connection. A fail response

still includes the version, flags, and packet size information, and may include a **Description** header to expand on the meaning of the response code value.

3.3.1.9 Example

The following example shows a **CONNECT** request and response with comments explaining each component. The **CONNECT** sends two optional headers describing the number of objects and total length of the proposed transfer during the connection.

Client Request:	**bytes**	**Meaning**
Opcode	0x80	**CONNECT**, Final bit set
	0x0011	packet length = 17
	0x10	version 1.0 of OBEX
	0x00	flags, all zero for this version of OBEX
	0x2000	8K is the max OBEX packet size client can accept
	0xC0	HI for **Count** header (optional header)
	0x00000004	four objects being sent
	0xC3	HI for **Length** header (optional header)
	0x0000F483	total length of hex F483 bytes
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x0007	packet length of 7
	0x10	version 1.0 of OBEX
	0x00	Flags
	0x0400	1K max packet size

3.3.1.10 OBEX Operations without Connect

It is highly recommended that implementations assume default values for connection parameters (currently just a minimum OBEX packet size of 255 bytes) and accept operations such as **PUT** and **GET** without first requiring a **CONNECT** operation.

3.3.2 Disconnect

This opcode signals the end of the OBEX session. It may include a **Description** header for additional user readable information. The **DISCONNECT** request and response must each fit in one OBEX packet and have their Final bits set.

Byte 0	Bytes 1, 2	Bytes 3 to n
0x81	packet length	optional headers

The response to **DISCONNECT** is 0xA0 (Success), optionally followed with a **Description** header. A **DISCONNECT** may not be refused. However, if the disconnect packet contains invalid information, such as an invalid **Connection Id** header the response code may be “Service Unavailable” (0xD3). Server side handling of this case is not required.

Byte 0	Bytes 1, 2	Bytes 3 to n
0xA0 or 0xD3	response packet length	optional response headers

It is permissible for a connection to be terminated by closing the transport connection without issuing the OBEX **DISCONNECT** operation. Though, this precludes any opportunity to include descriptive information about the disconnection. Currently, this is common practice and **DISCONNECT** is infrequently used. However, it is good practice to send an OBEX **DISCONNECT** for each OBEX **CONNECT** sent but few applications track or care about such details.

3.3.3 Put

The **PUT** operation sends one object from the client to the server. The request will normally have at least the following headers: **Name** and **Length**. For files or other objects that may have several dated versions, the **Date/Time** header is also recommended, while the **Type** is very desirable for non-file object types. However, any of these may be omitted if the receiver can be expected to know the information by some other means. For instance, if the target device is so simple that it accepts only one object and prevents connections from unsuitable parties, all the headers may be omitted without harm. However, if a PC, PDA, or any other general-purpose device is the intended recipient, the headers are highly recommended.

A **PUT** request consists of one or more request packets, the last of which has the Final bit set in the opcode. The implementer may choose whether to include an object **Body** header in the initial packet, or wait until the response to a subsequent packet is received before sending any object body chunks.

Byte 0	Bytes 1, 2	Bytes 3 to n
0x02 (0x82 when Final bit set)	packet length	sequence of headers

Each packet is acknowledged by a response from the server as described in the general session model discussion above.

Byte 0	Bytes 1,2	Bytes 3 to n
Response code typical values: 0x90 for Continue 0xA0 for Success	Response packet length	optional response headers

3.3.3.1 Headers used in Put

Any of the headers defined in the Object model chapter can be used with **PUT**. These might include **Name**, **Type**, **Description**, **Length**, **Connection Id, HTTP** or other headers specifying compression, languages, character sets, and so on. It is strongly recommended that headers describing the object body precede the object **Body** headers for efficient handling on the receive side. If **Name** or **Type** headers are used, they must precede all object **Body** headers.

3.3.3.2 Put Response

The response for successfully received intermediate packets (request packets without the Final bit set) is 0x90 (Continue, with Final bit sent). The successful final response is 0xA0 (Success, with Final bit set). The response to any individual request packet must itself consist of just one packet with its Final bit set - multi-packet responses to **PUT** are not permitted.

Any other response code indicates failure. If the length field of the response is > 3 (the length of the response code and length bytes themselves), the response includes headers, such as a **Description** header to expand on the meaning of the response code value.

Here is a typical Final response:

Server Response:	**Bytes**	**Meaning**
response code	0xA0	SUCCESS, Final bit set
	0x0003	length of response packet

3.3.3.3 Put Example

For example, here is a **PUT** operation broken out with each component (opcode or header) on a separate line. We are sending a file called jumar.txt, and for ease of reading, the example file is 4K in length and is sent in 1K chunks.

Client Request:	**Bytes**	**Meaning**
opcode	0x02	**PUT**, Final bit not set
	0x0422	1058 bytes is length of packet
	0x01	HI for **Name** header
	0x0017	Length of **Name** header (Unicode is 2 bytes per char)
	JUMAR.TXT	name of object, null terminated Unicode
	0xC3	HI for **Length** header
	0x00001000	Length of object is 4K bytes
	0x48	HI for Object **Body** chunk header
	0x0403	Length of **Body** header (1K) plus HI and header length
	0x.....	1K bytes of body
Server Response:		
response code	0x90	CONTINUE, Final bit set
	0x0003	length of response packet
Client Request:		
opcode	0x02	**PUT**, Final bit not set
	0x0406	1030 bytes is length of packet
	0x48	HI for Object **Body** chunk
	0x0403	Length of **Body** header (1K) plus HI and header length
	0x.....	next 1K bytes of body
Server Response:		
response code	0x90	CONTINUE, Final bit set
	0x0003	length of response packet

Another packet containing the next chunk of body is sent, and finally we arrive at the last packet, which has the Final bit set.

Client Request:		
opcode	0x82	**PUT**, Final bit set
	0x0406	1030 bytes is length of packet
	0x49	HI for **End-of-Body** chunk
	0x0403	Length of header (1K) plus HI and header length
	0x.....	last 1K bytes of body
Server Response:		
response code	0xA0	SUCCESS, Final bit sent
	0x0003	length of response packet

3.3.3.4 Server side handling of Put objects

Servers may do whatever they wish with an incoming object - the OBEX protocol does not require any particular treatment. The client may “suggest” a treatment for the object through the use of the **Target** and **Type** Headers, but this is not binding on the server. Some devices may wish to control the path at

which the object is stored (i.e. specify folder information such as C:\bin\pizza.txt rather than just pizza.txt). Path information is transferred using the **SETPATH** operation, but again, this is not binding on the server. It is important that clients, who have a particular purpose in mind when transferring an object, connect to a specific service that it knows can perform the desired behavior.

3.3.3.5 Sending objects that may have read errors

When sending the last portion of an object in an **End-of-Body** header, ambiguity arises if there is any chance that there are read errors in this last portion. This is because the **End-of-Body** normally triggers the receiving side to close the received object and indicate successful completion. If an **ABORT** packet subsequently arrives, it is "too late".

The recommended approach when sending objects which may have such errors is to send the **End-of-Body** header only when the sender knows that the entire object has been safely read, even if this means sending an empty **End-of-Body** header at the end of the object. This applies to both **GET** and **PUT** operations.

3.3.3.6 Put-Delete and Create-Empty Methods

A **PUT** operation with NO **Body** or **End-of-Body** headers whatsoever should be treated as a *delete* request. Similarly, a **PUT** operation with an empty **End-of-Body** header requests the recipient to create an *empty* object. This definition may not make sense or apply to every implementation (in other words devices are not required to support delete operations or empty objects),

A folder can be deleted using the same procedure used to delete a file. Deleting a non-empty folder will delete all its contents, including other folders. Some servers may not allow this operation and will return the "Precondition Failed" (0xCC) response code, indicating that the folder is not empty. In this case, the client will need to delete the contents before deleting the folder.

3.3.4 Get

The **GET** operation requests that the server return an object to the client. The request is normally formatted as follows:

Byte 0	Bytes 1, 2	Bytes 3 to n
0x03	Packet length	sequence of headers

The **Name** header can be omitted if the server knows what to deliver, as with a simple device that has only one object (e.g. a maintenance record for a machine). If the server has more than one object that fits the request, the behavior is system dependent, but it is recommended that the server return Success with the "default object" which should contain information of general interest about the device.

The final bit is used in a **GET** request to identify the last packet containing headers describing the item being requested, and the request phase of the **GET** is complete. The server device must not begin sending the object body chunks until the request phase is complete. Once a **GET** is sent with the final bit, all subsequent **GET** request packets must set the final bit until the operation is complete.

A successful response for an object that fits entirely in one response packet is 0xA0 (Success, with Final bit set) in the response code, followed by the object body. If the response is large enough to require multiple **GET** requests, only the last response is 0xA0, and the others are all 0x90 (Continue). The object is returned as a sequence of headers just as with **PUT**. Any other response code indicates failure. Common non-success responses include 0xC0 bad request, and 0xC3 forbidden. The response may include a **Description** header (before the returned object, if any) to expand on the meaning inherent in the response code value.

Byte 0	Bytes 1,2	Bytes 3 to n

response code	Response packet length	optional response headers

A typical multi-step **GET** operation proceeds as follows: the client sends a **GET** request that may include a **Name** header; server responds with 0x90 (Continue) and headers describing the name and size of the object to be returned. Seeing the Continue response code, the client sends another **GET** request (with final bit set and no new headers) asking for additional data, and the server responds with a response packet containing more headers (probably **Body** Headers) along with another Continue response code. As long as the response is Continue, The client continues to issue **GET** requests until the final body information (in an **End-of-Body** header) arrives, along with the response code 0xA0 Success.

3.3.4.1 The default GET object

Refer to chapter 8.4 OBEX Get Default Object for more information on providing default objects. This chapter defines a mechanism by which the client can request a specific type of default object. Such as the default business card or web page.

3.3.5 Abort

The **ABORT** request is used when the client decides to terminate a multi-packet operation (such as **PUT**) before it would normally end. The **ABORT** request and response each always fit in one OBEX packet and have the Final bit set. An **ABORT** operation may include headers for additional information, such as a **Description** header giving the reason for the abort. Since there can only be one OBEX operation in progress at a time, the use of a **Connection Id** header to abort a directed multi-packet operation is optional.

Byte 0	Bytes 1, 2	Bytes 3 to n
0xFF	Packet length	optional headers

The response to **ABORT** is 0xA0 (success), indicating that the abort was received and the server is now resynchronized with the client. If anything else is returned, the client should disconnect.

Byte 0	Bytes 1,2	Bytes 3 to n
0xA0	Response packet length	optional response headers

3.3.6 SetPath

The **SETPATH** operation is used to set the “current folder” on the receiving side in order to enable transfers that need additional path information. For instance, when a nested set of folders is sent between two machines, **SETPATH** is used to create the folder structure on the receiving side. The Path name is contained in a **Name** header. The **SETPATH** request and response each always fit in one OBEX packet and have the Final bit set.

Byte 0	Bytes 1, 2	Byte 3	Byte 4	Byte 5 to n
0x85	Packet length	flags	constants	optional headers

Byte 0	Bytes 1, 2	Bytes 3 to n
response code	Response packet length	optional response headers

Servers are not required to store objects according to **SETPATH** request, though it is certainly useful on general-purpose devices such as PCs or PDAs. If they do not implement **SETPATH**, they may return C0 (Bad Request) or C3 (Forbidden), and the client may decide whether it wishes to proceed.

When a new OBEX or TinyTP connection is established, the OBEX server's current folder should be its root folder. In this manner a device may retrieve (or serve) objects from the root folder without requiring the client to perform a **SETPATH** to the root first.

3.3.6.1 Flags

The flags have the following meanings:

bit	Meaning
0	backup a level before applying name (equivalent to ../ on many systems)
1	Don't create folder if it does not exist, return an error instead.
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

The unused flag bits must be set to zero by sender and ignored by receiver. Refer to section 3.3.3.6 for a description of how to delete folders.

3.3.6.2 Constants

The constants byte is entirely reserved at this time, and must be set to zero by sender and ignored by receiver.

3.3.6.3 Name header

Allowable values for the **Name** header are:

- <*name*> - go down one level into this folder name, relative to the current folder
- <*empty*> - reset to the default folder

In addition, the **Name** header may be omitted when flags or constants indicate the entire operation being requested (for example, back up one level, equivalent to "cd .." on some systems.

The receiving side starts out in a root folder. When sending a folder, the client will start by sending a **SETPATH** request containing the name of the folder. For example, if the full path for the folder on a PC is C:\notes, the client will send "notes" as the **Name** header of a **SETPATH** operation. A PC server would create or switch to the "notes" folder relative to the root folder (or perhaps create a mangled name if "notes" already exists and that was the preferred behavior). As subfolders of "notes" are encountered by the sending routine, additional **SETPATH** requests with the subfolder names will be sent. As folders are completed (all objects in the folder sent), the client will send a **SETPATH** (generally with no **Name** header) and the "back up a level" flag set.

3.3.7 Session

The method proposed to add new session commands is to create one new OBEX command called **SESSION**. The **SESSION** request is formatted as follows:

Byte 0	Bytes 1, 2	Bytes 3 to n
0x87 (Final bit set)	Packet length	Sequence of headers. **Session-Parameters** must be the first header in the sequence.

The positive response to a **SESSION** request is 0xA0 (Success, with the high bit set. Any failure response code can be used to indicate a negative response. Certain types of failures and the corresponding failure codes are described in the sections corresponding to each session command. The format of the response is as follows:

Byte 0	Bytes 1, 2	Bytes 3 to n
response code	Response packet length	Optional sequence of response headers. If the **Session-Parameters** is used it must be the first header in the sequence.

3.3.7.1 CREATESESSION

The **CREATESESSION** command is sent by the Client to create a new session. This command must include a **Session-Parameters** header containing the *Session Opcode, Device Address* and *Nonce* fields. Optionally, a *Timeout* field can be included. The successful response to **CREATESESSION** is 0xA0 (Success, with the high bit set) in the response code followed by a **Session-Parameters** header containing *Device Address*, *Nonce*, and *Session ID* fields. Optionally, a *Timeout* field can be included. The client should verify that it creates the same session ID as sent by the server. If the Session ID does not match then the client should close the session by using the **CLOSESESSION** command otherwise the session is considered active.

The **CREATESESSION** command and response must fit in a single OBEX packet of size 255 bytes (default maximum size) and have their Final bits set.

Only one session can be in the active state per transport connection. If a reliable session is already active a second request should be rejected using the 0xC3 (Forbidden) response. To switch sessions the current session must first be suspended. A server is not required to maintain multiple sessions at the same time. All servers will have some maximum number of suspended sessions they can maintain. At some point a server will receive a **CREATESESSION** request that will exceed this maximum number. The server can either reject the new request or close an existing session and accept the new request. The correct response depends on how long the existing sessions have been suspended. The server may want to prompt the user if it cannot decide on its own. A proposal for an algorithm is given in the next paragraph. If a **CREATESESSION** request is rejected because the maximum number of sessions already exists then the proper response is 0xD3 (Service Unavailable).

When a server receives a **CREATESESSION** request that exceeds the maximum number of sessions supported the following algorithm can be used.

1. Keep all suspended sessions with infinite timeouts on a sorted list by time suspended. The session with the longest time is at the front of the list.
2. Sessions with timeouts less then infinite should not be closed until the timeout occurs.
3. When a **CREATESESSION** request occurs exceeding the maximum number of sessions, the server should check the suspended session list. If the list is not empty the first session on the list is removed. This session is closed and its resources are used for the new session. If the list is empty the incoming **CREATESESSION** request is rejected.

3.3.7.2 CLOSESESSION

CLOSESESSION is used to gracefully close an existing session. This command must include a **Session-Parameters** header containing the *Session Opcode and Session ID* fields. The value of the *Session ID*

field is the one sent by the server in the response to the **CREATESESSION** command corresponding to the session being closed. The successful response to **CLOSESESSION** is 0xA0 (Success, with the high bit set) in the response code. The **CLOSESESSION** command can be used to close the active session or any suspended sessions.

The **CLOSESESSION** command and response must fit in a single OBEX packet and have their Final bits set.

The CLOSESESSION command cannot be rejected if the *Session ID* corresponds to a valid session. If the *Session ID* does not correspond to a valid session then the proper response is 0xC3 (Forbidden). If the active session is closed the unreliable session becomes the active session.

3.3.7.3 SUSPENDSESSION

SUSPENDSESSION is used to gracefully suspend the active session. This command must include a **Session-Parameters** header containing the *Session Opcode* field. Optionally, a *Timeout* field can be included. The successful response to **SUSPENDESESSION** is 0xA0 (Success, with the high bit set) in the response code. Optionally, a *Timeout* field can be included.

The **SUSPENDSESSION** command and response must fit in a single OBEX packet and have their Final bits set.

The **SUSPENDSESSION** command cannot be rejected if a reliable session is active. If the unreliable session is the current active session then the proper response is 0xC3 (Forbidden). The unreliable session becomes the active session when a reliable session is suspended.

3.3.7.4 RESUMESESSION

RESUMESESSION is used to resume a session that has been suspended. A session is suspended by using the **SUSPENDSESSION** command or when the underlying transport is broken. This command must include a **Session-Parameters** header containing the *Session Opcode, Device Address*, *Nonce*, and *Session ID* and fields. Optionally, a *Timeout* field can be included. The *Session ID* corresponds to the session being resumed and is the value returned by the server in its response to the **CREATESESSION** command. The *Device Address* and *Nonce* are the values originally used in the **CREATESESSION** command. This is true even if the device address has changed or the session is being resumed over a different transport. Matching a session requires that all values match not just the *Session ID* (*Device Address*, *Nonce* and *Session ID* must match).

The successful response to **RESUMESESSION** is 0xA0 (Success, with the high bit set) in the response code. A **Session-Parameters** header containing *Device Address*, *Nonce*, *Session ID* and *Next Sequence Number* fields must be sent. A *Timeout* field is optional. If the *Device Address*, *Nonce* and *Session ID* do not correspond to a valid session then 0xD3 (Service Unavailable) should be returned. If the information returned by the server does not match then the client should disconnect from the server. **CLOSESESSION** should not be used since this could be a valid session for another device.

The **RESUMESESSION** command and response must fit in a single OBEX packet and have their Final bits set.

If the session is resumed successfully then the client must determine if a packet should be retransmitted. If the session was suspended gracefully then no packets need to be retransmitted otherwise the client must follow the algorithm below:

```
If the Next Sequence Number field equals the sequence number of the last
packet sent then {
    Re-Send last packet
}else if the Next Sequence Number field equals the sequence number of the
next packet to send then {
```

```
    If the response to the last packet was not received then {
        Re-Send last packet minus all headers except
        Session-Sequence-Number;
    } else {
        Send next packet;
    }
} else {
    /* Sequence number is invalid */
    Disconnect underlying transport connection to OBEX server;
}
```

The server sets the *Next-Sequence Number* field in the **RESUMESESSION** response to the sequence number of the next OBEX packet it expects to receive. The server must behave according to the algorithm below upon receipt of the first request packet from the client after resuming the session.

```
If the sequence number of the request is equal to the (Next Sequence Number
field minus one) mod 256 then {
    /* The client did not receive the previous response */
    Re-send the previous response;
    Ignore the contents of the request since it was received properly
    before the link was suspended;
} else if the sequence number of the request is equal to the Next Sequence
Number field then {
    /* This is the next packet */
    Process the packet;
    Send the proper response;
} else {
    /* The sequence number is invalid */
    Send an Abort;
}
```

3.3.7.5 SETTIMEOUT

The **SETTIMEOUT** command is used to negotiate a new timeout specifying the number of seconds a session should be maintained while the session is suspended. This command must include a **Session-Parameters** header containing the *Session Opcode* field. Optionally, a *Timeout* field can be included. If the **Session-Parameters** header does not contain the *Timeout* field, the request is for an infinite timeout. The successful response to **SETTIMEOUT** is 0xA0 (Success, with the high bit set) in the response code. A **Session-Parameters** field may be sent with a *Timeout* field. The lowest value for the timeout is the value used by both sides. This command can only be used to change the timeout of the active connection. If the active connection is the unreliable connection then 0xC3 (Forbidden) is the proper response code.

The **SETTIMEOUT** command and response must fit in a single OBEX packet and have their Final bits set.

3.3.7.6 Generating a Session ID

The purpose of the *Session ID* is to uniquely identify the session. The *Session ID* is used when closing and resuming sessions. Both the client and the server want to be certain that the remote device is referring to the same session when issuing commands, especially when resuming a suspended session. Both the client and the server provide information used to generate the *Session ID*. This helps to make certain that the *Session ID* is unique. Each device provides two pieces of information. The first is its *Device Address*. In the case of IrDA this is the 32-bit device address, which is not guaranteed to be unique. The second is a *Nonce*, which must be unique for the given device. Given that the *Nonce* is unique as far as each device is concerned and that the device addresses are reasonably unique (even in the case of IrDA) the resulting *Session ID* has a high probability of being unique.

The Session ID is generated by applying the MD5 algorithm to the string created by concatenating the *Client Device Address*, *Client Nonce*, *Server Device Address* and *Server Nonce*. The reason for applying an algorithm such as MD5 is so that the size of *Session IDs* is fixed (16 bytes). Device Addresses and Nonces can very in size so the MD5 algorithm is used to produce a 16-byte quantity. The MD5 algorithm was designed to produce a unique number given a string or file. MD5 was chosen since it is used in the OBEX authentication procedure and was very likely to already be present in an OBEX implementation.

3.3.7.7 Negotiating Timeouts

All commands except **CLOSESESSION** involve timeout negotiation. Timeout negotiation is simple. Each time a Session Command exchange occurs (except **CLOSESESSION**) both sides take the smallest timeout values sent and use that as the new timeout for the session. If a timeout value is not explicitly sent it is assumed that the device is requesting an infinite timeout.

3.4 Packet Timeouts

OBEX does not impose any timeouts between OBEX packets. Deciding whether to continue or cancel a session is normally left to user control, since waiting for user action is often what creates long periods between packets. In devices with reasonable user interface capabilities, timeouts are not recommended. However this specification does not prohibit the use of timeouts. Some devices may find timeouts useful or even necessary, particularly when sufficient user interface to understand and/or control the reason for delay is not available.

3.5 Authentication Procedure

The OBEX authentication procedure is based on two OBEX headers, the **Authenticate Challenge** and **Authenticate Response** headers. It is assumed that the client and the server both share a secret such as a password or pin number. This value is not sent over OBEX during the authentication process.

The client authenticates the server by sending an OBEX command with an **Authenticate Challenge** header. This header will contain the digest-challenge, which is described later. To be authenticated, the server's response packet must contain the SUCCESS response code and an **Authenticate Response** header. The content of the **Authenticate Response** header contains the digest-response string, which is described later. The client verifies the server by generating its own request-digest string and comparing it to the one sent as part of the digest-response. If the server is not authenticated the client can simply disconnect from the server.

The server authenticates the client using the same basic algorithm. When a client attempts an operation to the server which requires authentication, the server sends an **Authenticate Challenge** header along with an UNAUTHORIZED response code in a response packet. When the client receives this response, it must re-send the command packet with an **Authenticate Response** header. The server verifies the client by generating its own request-digest and comparing it to the one sent in the digest-response by the client. If they are the same, the client is authenticated. If the client is not authenticated, the server will simply respond to all operations, including the current one, with the UNAUTHORIZED response code.

The algorithm used by the client to authenticate a server is straightforward. The client will normally authenticate the server as part of the OBEX **CONNECT** procedure. The server can authenticate the client for any operation including the **CONNECT** operation and individual **PUT** and **GET** operations. When the server authenticates the client for an OBEX connection the server will send the **Authenticate Challenge** header in the response to the **CONNECT** command with a response code of UNAUTHORIZED. When the client receives a response to the **CONNECT** command containing an **Authenticate Challenge** header and the UNAUTHORIZED response code, it should repeat the **CONNECT** command and include an **Authenticate Response** header. If the original **CONNECT** command also contained an **Authenticate**

Challenge and/or **Target** header, these headers must also be present in the second **CONNECT** command.

When the server wants to authenticate an individual operation, it will reject the first attempt by the client with an OBEX response containing the UNAUTHORIZED response code and an **Authenticate Challenge** header. The client must retry the operation this time including an **Authenticate Response** header. If every operation requires authentication then it probably makes sense to authenticate the **CONNECT** operation and not perform individual operation authentication.

Multiple **Authenticate Challenge** headers can be sent in one OBEX packet. This is done for a number of reasons:

- In the future, there may be different types of challenge algorithms or hashing functions used. Each challenge will represent a possible algorithm to use.
- There may be different passwords for full access versus read only access. Two headers are used, one for each access method.

When more than one **Authenticate Challenge** header exists in an OBEX packet, the nonce must be different for each one. Devices are not required to deal with multiple **Authenticate Challenge** headers so some devices may only process the first one. Therefore, it is important that the most general or common challenge be sent in the first header. When multiple **Authenticate Challenge** headers are sent, the sender will use the nonce in the digest-response to determine which nonce to use when generating the request-digest. If a nonce does not exist in the digest-response then the nonce sent in the first **Authenticate Challenge** header is used.

3.5.1 Digest Challenge

The OBEX digest-challenge is a simplified version of the HTTP digest-challenge. One of the limitations of OBEX is that the response to a **PUT** or **CONNECT** command must fit in a single OBEX packet. Some devices only support a maximum size OBEX packet of 255 bytes so this limits the amount of data that can fit in a response and therefore, limits the size of the digest-challenge and digest-response.

The HTTP challenge contains a number of fields including a string that identifies the challenge as basic or digest. Within a digest challenge the digest-challenge can specify the hashing algorithm to use, the realm, the domain, a nonce, and option fields. The main element is the nonce. The nonce is the random string, which is combined with the secret to produce the main portion of the digest-response. The OBEX authentication algorithm needs to be simple yet flexible. The minimal challenge should only contain a nonce but there may be cases where a userid/username is needed to determine which password to use. Some OBEX servers may support multiple different users each with their own passwords. Therefore, the server must be able tell the client that a userid is needed.

The default hashing function in HTTP is MD5 and OBEX will use this algorithm. In the future, other algorithms may be introduced but at this time, only MD5 is supported.

The OBEX digest-challenge string can contain multiple fields. The HTTP digest-challenge is based on ASCII strings. To keep the size small, the OBEX digest-challenge will use a tag-length-value encoding scheme. Both the tag and value length fields, are one byte in length. The table below shows the OBEX digest-challenge fields.

Tag	Name	Value Len	Value Description	Default Value
0x00	Nonce	16	String of bytes representing the nonce.	
0x01	Options	1	Optional Challenge Information	0

0x02	Realm	*n*	A displayable string indicating which userid and/or password to use. The first byte of the string is the character set to use. The character set uses the same values as those defined in IrLMP for the nickname.	

Each field is described in more detail below.

3.5.1.1 Nonce

The nonce is required. It is important that the nonce be different each time it is sent. An example nonce, based on the example given in the HTTP authentication document, is as follows:

```
H(time-stamp ":" private-key)
```

Where time-stamp is a sender-generated time or other non-repeating value and private-key is data known only to the sender. The MD5 hashing algorithm is applied to the time-stamp string to produce a 128 bit (16 byte) binary string. This resulting string is the nonce.

3.5.1.2 Options

The option field is optional. If an option field is not sent the assumed value for it is 0. Two options are defined. The first controls the sending of a userid. If this option is set then the entity receiving the challenge must respond with a userid in the digest response string. The second option indicates the access mode being offered by the sender of the challenge. The two access modes are full access (both read and write) and read only.

bit	Meaning
0	When set, the User Id must be sent in the authenticate response.
1	Access mode: Read Only when set, otherwise Full access is permitted.
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

The unused options bits must be set to zero by sender and ignored by receiver.

3.5.1.3 Realm

The realm field is optional. The realm is intended to be displayed to users so they know which userid and password to use. The first byte of the string is the character set of the string. The table below shows the different values for character set.

Char Set Code	Meaning
0	ASCII
1	ISO-8859-1
2	ISO-8859-2
3	ISO-8859-3
4	ISO-8859-4
5	ISO-8859-5
6	ISO-8859-6
7	ISO-8859-7
8	ISO-8859-8

9	ISO-8859-9
0xFF = 255	UNICODE

3.5.2 Digest Response

The digest-response string can contain multiple fields and uses a tag-length-value encoding scheme. The digest-response fields are shown below.

Tag	Name	Value Len	Value Description	Default Value
0x00	Request -digest	16	String of bytes representing the request digest.	
0x01	User Id	*N*	User ID string of length n. Max size is 20 bytes.	
0x02	Nonce	16	The nonce sent in the digest challenge string.	

Each field is described in more detail below.

3.5.2.1 Request-digest

The request-digest is required and is calculated as follows:

```
H(nonce ":" password)
```

The nonce is the string sent in the digest-challenge. The password is the secret known by both the client and server. The MD5 hashing function is applied to the nonce/password string to produce a 128-bit (16 byte) string. This resulting string is the request-digest

3.5.2.2 Userid

The userid is required if the digest-challenge contains an options field with the userid bit set to 1. The userid is used to identify the proper password.

3.5.2.3 Nonce

The nonce is optional. When multiple **Authenticate Challenge** headers are sent, the nonce is used to indicate to which header the request-digest is responding. The nonce is the same value as sent in the digest-challenge. If a nonce is not sent in the digest-response then it is assumed that the digest-response is a response to the first **Authenticate Challenge** header. This allows devices that do not parse multiple **Authenticate Challenge** headers to work with devices that send multiple headers.

3.5.3 Hashing Function

The default hashing function used in OBEX is MD5. The following paragraph briefly describes the MD5 algorithm. Refer to RFC 1321 at http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1321.txt for more detailed information, including example source code. Additional MD5 source code can be found in Appendix 12.5 MD5 Algorithm for Authentication at the end of this specification.

The MD5 algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications,

where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 algorithm does not require any large substitution tables; the algorithm can be coded quite compactly.

3.5.4 Authentication Examples

Example 1: Client and Server Connection Authentication (no userid)

	Opcode or Response plus Headers	Final bit	Header Data	Header Length
Request →	**CONNECT** **Authenticate Challenge** **Target**		0x00, 0x10, <nonce>, 0x01, 0x01, 0x00 <target-UUID>	24 19
Response ←	UNAUTHORIZED **Authenticate Challenge**	✓	0x00, 0x10, <nonce>, 0x01, 0x01, 0x00	24
Request →	**CONNECT** **Authenticate Challenge** **Authenticate Response** **Target**		0x00, 0x10, <nonce>, 0x01, 0x01, 0x00 0x00, 0x10, <request-digest> <target-UUID>	24 21 19
Response ←	SUCCESS **Connection Id** **Who** **Authenticate Response**	✓	0x00000001 <target-UUID> 0x00, 0x10, <request-digest>	5 19 21

Example 2: Server Authenticates Get operation (userid required)

	Opcode or Response plus Headers	Final bit	Header Data	Header Length
Request →	**GET** **Name**	✓	 "Test File Object"	 37
Response ←	UNAUTHORIZED **Authenticate Challenge**	✓	 0x00, 0x10, <nonce>, 0x01, 0x01, 0x01	 24
Request ←	**GET** **Name** **Authenticate Response**	✓	 "Test File Object" 0x00, 0x10, <request-digest>, 0x01, 0x08, <userid>	 37 31
Response ←	CONTINUE **Length** **Time** **Body**	✓	 "2000" "0x41a50016" "… start of file data…."	 499

3.6 Multiplexing with OBEX

It is sometimes necessary for different clients to perform operations in a manner that appears to the user to be simultaneous. In addition, some connections may be very long lived while others come and go quickly. This creates the need for a mechanism that allows multiple clients to access OBEX services concurrently. The following sections cover two methods recommended for multiplexing commands over OBEX. The first method is geared toward use with applications that run under the umbrella of the default OBEX server.

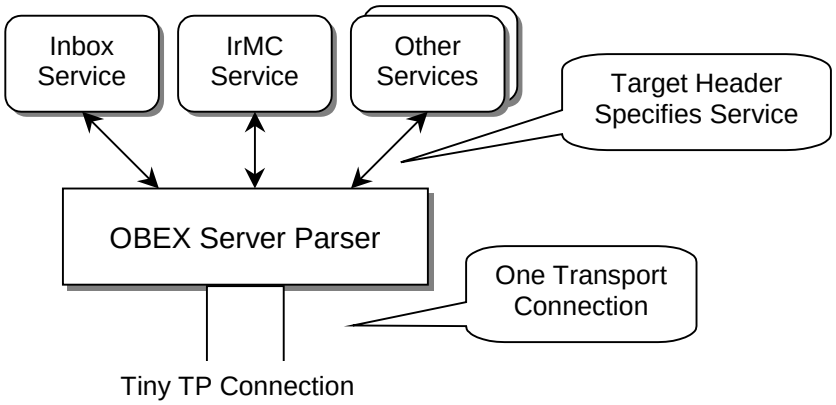


Figure 3.6.1: OBEX Command Level Multiplexing

3.6.1 Connections multiplexed at the OBEX Command level

The OBEX protocol layer provides this form of multiplexing. Multiple OBEX connections (using the **Connection Id** header) are allowed at one time but only one command can be executed at a given time. This type of multiplexing only requires one OBEX parser because the multiplexing occurs above the parser. This type of multiplexing is necessary because most services are accessed through the default OBEX server and hence over a single transport connection.

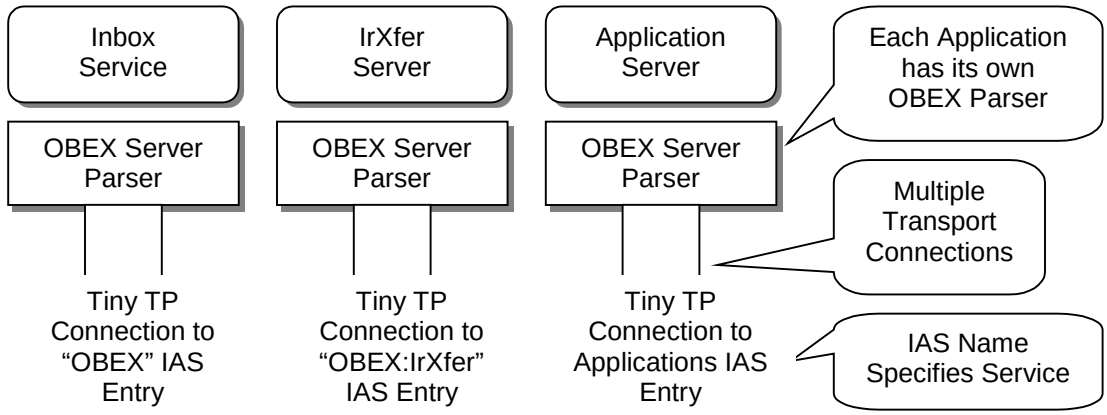


Figure 3.6.2: OBEX Transport Layer Multiplexing

3.6.2 Connections multiplexed at the OBEX Transport layer

This type of multiplexing is external to OBEX and relies on the multiplexing capabilities of the underlying transport layer. Multiple OBEX servers exist, one for each different application. In turn each of these OBEX servers may themselves offer the connection and multiplexing types described above. In IrDA for example, the LM-MUX provides this form of multiplexing. Multiplexing at the transport layer is most useful when neither side of the application are constrained or limited by resources. It also allows for a finer grained multiplexing which may be important for some applications.

4. OBEX Application Framework

The OBEX application framework is a set of conventions and services designed for the purpose of creating interoperable devices. OBEX is a very flexible protocol and the application framework tightens up the usage of OBEX while creating a set of standard services that satisfy many object exchange needs. The foundation of the OBEX application framework is the Default OBEX Server.

4.1 The Default OBEX Server

The Default OBEX Server is the server that resides at the LSAP-Sel specified by the IAS entry with class name “OBEX” (see section 6.1 IAS entry for details on this IAS entry). The Default OBEX server is the “Well Known” OBEX server. All the standard services along with many applications are accessed via this server including IrMC applications. For general device to device interoperability the default OBEX server is used. There can only be one entity registered to be the default OBEX server. If more than one exists it will be impossible for another device to distinguish between the two.

The default OBEX server can be used to access file systems, databases, services and applications. OBEX provides conventions for distinguishing the types of accesses. The **Name**, **Type**, **Target** and **Connect Id** headers can be used for this purpose. The most basic service accessed via the default OBEX server is the inbox service.

4.2 The Inbox Service

The most basic form of object exchange is pushing a file/object to another device. The sending device (client) is only responsible for making sure the object is transferred correctly. The client does not know or care what happens to the object after it has been received. The receiver (server) is responsible for placing the object in the correct location. This basic from of file transfer is accomplished with the OBEX **PUT** operation. The client does not need knowledge of the server’s object store or folder hierarchy. The client initiates the operation by sending the object to the default OBEX server. When pushing an object, a **Name** header is used to name the object and a **Type** header may be used to identify the type. Other headers such as **Length** and **Time** can be also sent.

In this simple form of object transfer, the object is pushed into the server’s “inbox”. Inboxes come in many forms. A PC based application could have a folder in the PC’s file system, which represents the inbox. But, an inbox does not necessarily have to be a fixed folder or location. It is also possible to dispatch the object based on the **Type** header or the extension in the **Name** header to a more appropriate location such as an application or database. Products, which do not have a file system, can operate in this way. The structure of the inbox is an issue strictly for the server. The client simply pushes objects to the “inbox”. What happens after that is up to the server. The push model works best for single objects. It can also be used for moving whole folder/directory hierarchies.

It is important to note that what may be an application specific object for one device is just a file in another device. Thus, it is important that all devices standardize on using the default OBEX server for performing this simple operation. In this way, interoperability is more likely to occur. The capability service is also used to help increase the interoperability for simple object pushing. Refer to chapter 8.2 Simple OBEX Put file transfer (+ SetPath) for more information on the Inbox model.

The Inbox service also enables a device to **GET** certain objects from another device without understanding the semantics of the service or database that contains the object. This is called the default **GET** and is intended for use with a narrow band of object types that have a default object. A vCard is a good example of an object type that has a default object.

4.3 Inbox Connection

The inbox is accessed via an OBEX connection called the “inbox” connection. This connection is made to the default OBEX server by sending a **CONNECT** command with no **Target** headers. Objects sent using Ultra are placed in the inbox but in this case the **CONNECT** command is not used. Other services are also accessed via the inbox connection. These services all have specific characteristics that allow an implementation to distinguish between the different service requests. The table below gives the complete list of services available via the inbox connection along with the distinguishing characteristics.

Service type	Service description
Capability Service	The capability object is accessed by using a **GET** command with a **Type** header containing the MIME type of the capability object, “x-obex/capability”.
IrMC Level 2 and 3	Level 2 and 3 services of the IrMC application are accessed using **PUT** and **GET** commands with a **Name** header in which the first part the name contains “telecom/”.
Pushing objects into the inbox	Objects are pushed into the inbox by using the **PUT** command with a **Name** header. The string in the **Name** header should not contain any path characters such as ‘:’, ‘/’ or ‘\’. Objects with improperly formed names should be rejected.
Pulling objects from the inbox	Default objects are pulled from the inbox by using the **GET** command with a **Type** header containing the MIME type of the desired object. **Name** headers are not allowed.

4.4 Capability Service

The OBEX capability service is designed to provide a general-purpose method for accessing service information with OBEX. The capability service lists the type of objects supported and details about the fields or formats of specific types. A client may be able to render an object into multiple formats. By reading the server’s capability object the client can determine the best format to send the object. The client can also determine if it makes sense to even send an object at all.

The capability service is based upon two OBEX objects, the capability object and the object profile object. The capability object contains general-purpose information about the device, including the services and applications that are supported. The object profile object contains information specific to the objects that the device supports. The details of the capability service and its objects can be found in sections 8.5 Capability Service, 9.3 The Capability Object and 9.4 The Object Profile Object.

4.5 Custom OBEX applications

OBEX can also be used as a protocol within custom applications, providing the basic structure for the communication. In this case, the application may set the OBEX IAS hint bit, but it must not use the OBEX IAS entry. Instead, the custom application should define its own unique IAS entry.

While a custom application will (by definition) work best when connected to a strict peer, there are cases where it may be useful to connect to the default OBEX server. In particular, if a custom application can send any objects of general interest but cannot find a peer application, it could send the data to a default OBEX server inbox, perhaps modifying the sending format to be a common computer format such as text, GIF, JPEG, or such.

For instance, a digital camera could look for digital film development kiosks, but also want to share photos from last summer’s camping trip with any PC, PDA, or smart TV. A pager might delete messages automatically after sending them to its peer application (knowing they are completely taken care of), but

send them as text objects to any OBEX enabled device just for easy viewing or sharing. Any IR device whatsoever might offer its “business card”, describing who and what it is.

4.6 Directed Operations and Connections

A directed operation is an operation that is sent toward a specific service. An OBEX **Target** header is used to identify the designated service. If the receiving side has a matching service, all packets in the request are forwarded (or re-directed) to that service. In addition, a **Who** header is sent in the response to indicate that the request is being serviced by the requested service. If a matching service cannot be located, the server is encouraged to accept the request if at all possible. This allows the client to make the final decision on whether to continue the operation or abort. In this case a **Who** header is not returned in the response. When the operation is complete, the association is dissolved. If the client wishes to direct another operation to the service it must again include the **Target** header. This is where directed connections come in handy.

Directed connections enable a client and server to establish a virtual connection based on the principles of directed operations. They exist for clients that need to perform multiple operations to a specific service. Once a directed connection is established, the client can direct operations to the same service by simply including the **Connection Id** header in each request. A directed connection is initiated when a client sends an OBEX **CONNECT** packet with a **Target** header containing the UUID of the service it wishes to connect to. A successful match is indicated with a **Who** header in the response packet as well as a **Connection Id** header. The **Connection Id** header contains the identifier of the connection that the client will use in future requests as shorthand for a **Target** header.

4.6.1 Directed Connection Mechanics

Directed connections do not follow the usual paradigms one would associate with a “connection”. It is recommended that directed connections always be “open”. Therefore, it is unnecessary to close a connection with an OBEX **DISCONNECT** operation. In essence, a directed connection provides a shortcut so the client doesn’t have to specify a **Target** header in every operation. As well as providing insurance that the designated service will receive the clients requests.

It is not necessary to track the open or closed status of directed connections. Therefore OBEX **CONNECT** operations should not be reference counted and matched with OBEX **DISCONNECT** operations.

It may or may not be necessary for a server to differentiate an operation that arrives with a **Connection Id** from one that arrives with a **Target** header for the same service (other than making sure to respond with the **Who** header). This is unlikely to be an issue in normal implementations but it should be noted that the acceptance of a directed connection by a service, does not guarantee that another operation sent with a **Target** header to the same service will be accepted by that service.

It is anticipated that **CONNECT** operations that are intended to establish directed connections will contain the same flags, OBEX packet size and protocol versions as did all previous **CONNECT** operations. Handling of **CONNECT** options that differ from original or previous values is implementation dependent.

4.6.2 Target Header Processing

In some cases, more than one **Target** header may be present in a request. The server should match the headers in the order received and ignore any **Target** headers encountered after a match is made. The matching header is then sent in the response **Who** header.

4.7 OBEX Access Methods

Since not all objects are the same, there arises the need to define methods for accessing different types of objects. This section defines sets of standard procedures or templates for each type of access. Three basic forms of access have been identified: File, Database and Process. Interoperability will become easier when, for example, accessing a database all clients follow the same basic procedures. Of course, the values in the headers and such will differ for each database but the basic procedure should be the same.

4.7.1 File Access

File level access is the lowest level of object access provided by OBEX. Using file level access an object can be retrieved and stored only as a whole. It relies on addressing an object by providing a **Name** and optionally a **Type** header. The **Name** header identifies the instance of the object. This is often sufficient when sending an object. The **Type** header is used to refine the objects identification when multiple objects may exist with the same name.

4.7.2 Database Access

A database is different from a file system. The objects manipulated in a file system tend to be course blobs. While the items in a database are more refined. Often, the individual fields and records can be manipulated in a database. Whereas many different types of objects are stored in a file system, databases tend to store or organize objects of a similar nature.

OBEX database level access allows for finer granularity access to objects. Objects accessed using database level headers can be manipulated by part. In other words pieces of a database object can be requested and stored instead of having to exchange the entire object.

Database access relies on a **Name** header to identify the instance of the database object. In addition, OBEX has an **Object Class** header, which is designed for use with database objects.

4.7.3 Process / RPC

In interacting with a process, the actual information transferred is generated and consumed on the fly. The best example is the Synchronization command interpreter in the IrMC specification. The items being **PUT** are often commands and not objects.

5. Using OBEX over IrDA Ultra-Lite (Connectionless use)

OBEX is constructed to take advantage of the benefits of a connection-oriented transport, for instance by exchanging capabilities and version information just once in the **CONNECT** packet. However, some devices support only connectionless operations, such as those described by the IrDA Ultra proposal.

Ultra refers to communications carried out over connectionless data services of IrLAP, as described in "Guidelines for *Ultra* Protocols". The Ultra protocol is not reliable and does not guarantee ordered delivery, so the request/response model of a full OBEX implementation is not appropriate. However, for delivery of small objects, the simple structure of OBEX requests and the header based object wrapper are still useful. A number of issues arise and are discussed in the following paragraphs.

Ultra does not assure reliable ordered delivery. However, it does use IrLAP framing, which has a CRC, so it is possible to tell if an individual packet arrived intact. It also uses a segmentation-reassembly (SAR) counter in each packet header, and a maximum time constraint between consecutive packets using SAR. The count allows up to 15 packets to be reassembled. The CRC and SAR acting together put a maximum upper bound on the size of an Ultra OBEX packet of 15 times the max packet size permitted by the link speed of either 2400 or 9600 bps. Connectionless IrLAP packets are limited to 64 bytes, of which 2 bytes are used by IrLMP and 2 bytes are used by the PID and SAR fields. Thus leaving 15 packets of 60 bytes (900 bytes total) for use by Ultra OBEX. In practice, transfers of 15 packets will take too long, and Ultra OBEX is best suited for very small objects that will fit in one (or a few) packets. A typical well-suited operation would be beaming a single name and phone number between two devices.

Ultra OBEX uses only the OBEX **PUT** command and all object data and headers must be sent in one OBEX packet. The packet must have the final-bit set and all object data is contained in a single **End-of-Body** header. Minimum implementations of Ultra OBEX need only support the default OBEX packet size of 255 bytes. Legal implementations can however, support up to the maximum Ultra OBEX packet size of 900 bytes. For compatibility, it is recommended that devices do not send more than 255 bytes over Ultra.

To guarantee a successful transfer of data over Ultra, the exchange should be kept to 255 bytes or less . This includes all headers. The data should be contained in one OBEX packet. Meaning the first and only frame contains the **PUT**+FINAL opcode.

NOTE: Implementations are not required to receive OBEX packets larger than 255 bytes. Possible ways that implementations can deal with Ultra OBEX packets larger than 255 bytes is as follows:

1. Ignore the packet completely
2. Only accept the first 255 bytes
3. Accept the whole packet.

The OBEX **CONNECT**, **DISCONNECT, ABORT** and **SETPATH** operations are not used in Ultra OBEX, because Ultra OBEX does not have any notion of a session with multiple operations. Each operation is completely distinct from every other, and each operation must fit within the SAR constraint described above. Therefore, any version or capability information must be included as headers in the operation being performed.

Ultra OBEX does not use any response packets. All feedback as to the success or failure of the operation is performed by some out-of-band means such as a beep or visual indication to the user that the operation was successful. For this reason, the OBEX **GET** operation is also not used in Ultra OBEX.

6. IrDA OBEX IAS entries, service hint bit and TCP Port number

6.1 IAS entry

The default IrDA OBEX server application must have an IAS entry with the following attribute.

```
Class OBEX {
        Integer        IrDA:TinyTP:LsapSel     IrLMP LSAP selector, legal values from Ox00 to 0x6F
}
```

There should be only one such entry on a system, or it may be impossible for an incoming connection to decide who to connect to except by flipping a coin.

The OBEX class name should be used only by implementations that are intended as default OBEX servers, and in particular have some provision for dealing with multiple object types and services. Special purpose programs (custom applications) that use this protocol should define unique IAS class names that they alone will use. Alternatively, special purpose applications can be accessed via the default OBEX server by utilizing a directed connection as discussed in this specification. There is one very common subset of object exchange application - applications that are strictly for file exchange. They should use a class name of OBEX:IrXfer.

A general purpose OBEX implementation should look first for an OBEX IAS object, and if not found then look for an OBEX:IrXfer entry if the object can reasonably be represented as a file. Similarly, a special purpose program using OBEX should look first for a strict peer using whatever class name the peer normally uses, and if not found should look for the default OBEX server to accept the transfer.

6.1.1 IrDA:TinyTP:LsapSel

This attribute contains the TinyTP/IrLMP MUX channel number for the service. This attribute must be present for connection oriented use, whatever the class name is.

6.2 Service Hint bits

The OBEX IrLMP service hint bit has a value of 0x20 in the second hint byte. See section 3.4.1.1 in [IRDALMP] specification for the definition of service hint bits and [IRDAIAS] for the complete listing of service hint bits. Setting the OBEX hint bit is required. Many OBEX clients will not even attempt a connection unless the OBEX hint bit is set.

6.3 TCP port number

IANA has assigned the port number 650 to the OBEX protocol. This port number should be used when transporting OBEX protocol data over a TCP network. More specifically, it represents the location of the default OBEX server in the TCP network. As with TinyTP, there is no requirement on the alignment of OBEX packets with TCP PDUs. Except that all OBEX data shall be carried in data packets, not in TCP Connect or Disconnect packets.

7. OBEX Examples

The following examples are provided to round out the readers understanding of the procedures used in OBEX.

7.1 Simple Put - file/note/ecard transfer

This example describes a simple **PUT** operation, a scenario with many applications. It illustrates the basic request/response cycle, the exchange of version and capability information at the start of the connection, and the use of the **Name** and **Length** headers in the **PUT** operation.

Ms. Manager worked up an outline for a presentation last night on her laptop, and needs to give it to a staffer to expand on. She sets her laptop down next to an infrared adapter attached to the staffer's desktop machine, drags the Word document containing the outline to her OBEX app, and ...

A connection is made after her OBEX client queries staffer's IAS to find the required LSAP of the staffer's OBEX application. In the first two examples, we will show the transaction byte by byte. For ease of reading, the file and packet sizes are kept simple - file data is sent 1K at a time.

Client Request:	**bytes**	**Meaning**
opcode	0x80	**CONNECT**, Final bit set
	0x0007	7 bytes is length of packet
	0x10	version 1.0 of OBEX
	0x00	no connect flags
	0x2000	8K max packet size
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x0007	packet length of 7
	0x10	version 1.0 of OBEX
	0x00	no connect flags
	0x0800	2K max packet size
Client Request:	**bytes**	**Meaning**
opcode	0x02	**PUT**, Final bit not set
	0x0422	1058 bytes is length of packet
	0x01	HI for **Name**
	0x0017	Length of **Name** header
	THING.DOC	name of object, null terminated
	0xC3	HI for **Length**
	0x00006000	Length of object is 0x6000 bytes
	0x48	HI for Object **Body** chunk
	0x0403	Length of **Body** header (1K) plus HI and header length
	0x.....	1K bytes of body
Server Response:		
response code	0x90	CONTINUE, Final bit set
	0x0003	length of response packet
Client Request:		

opcode	0x02	**PUT**, Final bit not set
	0x0406	1030 bytes is length of packet
	0x48	HI for Object **Body** chunk
	0x0403	Length of **Body** header (1K) plus HI and header length
	0x.....	next 1K bytes of body
Server Response:		
response code	0x90	CONTINUE, Final bit set
	0x0003	length of response packet

A number of packets containing chunk of file body are sent, and finally we arrive at the last packet, which has the Final bit set.

Client Request:		
opcode	0x82	**PUT**, Final bit set
	0x0406	1030 bytes is length of packet
	0x49	HI for **End-of-Body** chunk
	0x0403	Length of header (1K) plus HI and header length
	0x.....	next 1K bytes of body
Server Response:		
response code	0xA0	SUCCESS, Final bit sent
	0x0003	length of response packet

The transaction is complete, so the OBEX client disconnects. 3 seconds have passed, and Ms. Manager heads down the hall to a meeting. No **Type** header was used, so the server assumes a binary file and stores it exactly as is.

7.2 Simple Get - field data collection

This example illustrates a **GET** operation with user defined headers, which, in this case, are for application specific access identifier and version information.

A meter keeps tabs on electricity used, and is intermittently visited by a meter reader. The meter reader points a collection device at the meter and presses a button which causes the following:

Client Request:	**bytes**	**Meaning**
opcode	0x80	**CONNECT**, Final bit set
	0x001A	26 bytes is length of packet
	0x10	version 1.0 of OBEX
	0x00	no special flags
	0x0800	2K max packet size
	0x70	HI for a user defined, length prefixed header
	0x0013	length of header
	XuseElectricityX	actual contents of user defined access header
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x000B	packet length of 11
	0x10	version 1.0 of OBEX
	0x00	no flags set
	0x0040	64 byte max packet size

	0xF0 0x00000603	HI for user defined 4 byte header meter version info
Client Request:	**bytes**	**Meaning**
Opcode	0x83 0x0003	**GET**, Final bit set length of **GET** packet
Server Response:		
Response code	0xA0 0x0038 0x49 0x0035 0x.....	SUCCESS, Final bit set length of response packet HI for **End-of-Body** chunk Length of header 0x32 bytes of meter information

A security code (“XUseElectricityX”) was passed in a user defined header during connection to keep prying eyes at bay. The **GET** operation specified no **Name** or other headers, so the meter returned a reading. A header entry at this point might have instructed the meter to return service information. No **Type** or **Name** headers were needed on the returned object because the client application knows just what it is getting.

7.3 Example Get of the Capability Object

This example shows a client requesting the capability object from a device. The OBEX packet size is 512 bytes.

	Opcode or Response plus Headers	Final bit	Header Data	Header Length	Running Total of Data Exchanged
Request →	**GET** **Type**	✓	“x-obex/capability	20	
Response ←	CONTINUE **Length** **Body**	✓	“2000” “start of capability object”	499	496 bytes
Request →	**GET** no headers	✓			
Response ←	CONTINUE **Body**	✓	“..continuation of object..”	509	1002 bytes
Request →	**GET** no headers	✓			
Response ←	CONTINUE **Body**	✓	“..continuation of object..”	509	1508 bytes
Request →	**GET** no headers	✓			
Response ←	SUCCESS **End-Of-Body**	✓	“..final segment of object”	495	2000 bytes

7.4 Connect using Target, Who and Connection Id headers

Here is a **PUT** operation broken out with each component (opcode or header) on a separate line. We are sending a file called jumar.txt, and for ease of reading, the example file is 4K in length and is sent in 1K chunks. This illustrates the format of a directed Connection, using the **Target**, **Who** and **Connection Id** headers. The connection was accepted by the Targeted application.

Client Request:	**bytes**	**Meaning**
opcode	0x80	**CONNECT**, Final bit set
	0x0023	packet length = 35
	0x10	version 1.0 of OBEX
	0x00	flags, all zero for this version of OBEX
	0x2000	8K is the max OBEX packet size client can accept
	0x46	**Target** HI
	0x0013	Length of **Target** Header
	0x382D2BD03C39	UUID for desired service/application (binary
	11D1AADC0040F6	representation)
	14953A	{382D2BD0-3C39-11d1-AADC-0040F614953A}
	0xC3	HI for **Length** header (optional header)
	0x0000F483	total length of hex F483 bytes
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x0023	packet length of 35
	0x10	version 1.0 of OBEX
	0x00	flags (LSAP-SEL multiplexing not supported)
	0x0800	2K max packet size
	0xCB	HI for **Connection Id** header
	0x00000001	ConnId = 1
	0x4A	**Who** HI
	0x0013	Length of **Who** Header
	0x382D2BD03C39	UUID of the responding application (same value as
	11D1AADC0040F6	**Target** header in request)
	14953A	{382D2BD0-3C39-11d1-AADC-0040F614953A}
Client Request:		
opcode	0x02	**PUT**, Final bit not set
	0x0427	1063 bytes is length of packet
	0xCB	HI for **Connection Id** header
	0x00000001	ConnId = 1
	0x01	HI for **Name** header
	0x0017	Length of **Name** header (Unicode is 2 bytes per char)
	JUMAR.TXT	name of object, null terminated Unicode
	0xC3	HI for **Length** header
	0x00001000	Length of object is 4K bytes
	0x48	HI for Object **Body** chunk header
	0x0403	Length of **Body** header (1K) plus HI and header length
	0x.....	1K bytes of body
Server Response:		
response code	0x90	CONTINUE, Final bit set
	0x0003	length of response packet

	The PUT operation	…proceeds to completion…
Client Request:	**bytes**	**Meaning**
opcode	0x81 0x0008 0xCB 0x00000001	**DISCONNECT**, Final bit set packet length = 8 HI for **Connection Id** header ConnId = 1
Server Response:		
response code	0xA0 0x0003	SUCCESS, Final bit set packet length = 3

7.5 Combined Get and Put - paying for the groceries

Now that you have the idea of what the byte sequence looks like, we will represent the remaining examples in a more readable format.

Smiling vacantly, the checkout clerk says “$45.12, please”. You take out your bit-fold (electronic bill-fold). You point it at the IR window at the checkstand, and press the “Do-it” key. An encrypted IR connection is made to the register. During the connection negotiation, you discover that the store takes Visa and the local bank’s debit card.

Request:
<opcode=Connect>
version info
encryption information

Response:
<response code = success>
version and capabilities info
encryption information
Accepted-payment-forms: Visa, MasterCard, ...

Request:
<opcode=Get> // Get with no arguments tells register to return itemized amount

Response:
<response code = success>
Type: text/itemized-receipt
Len: 2562 // wow, you bought a lot of items

<body of receipt in form standardized by finance industry.
It includes date, a UUID for the transaction, and the total amount due>

You examine the list of items, making sure you got the sale price on asparagus, and make sure your coupons were deducted. You press the Accept key, and select Visa from the offered payment choices...

Request:
<opcode=Put>
Type: payment/Visa
Len: 114
Payment-id: <UUID of payment sent to you in receipt>

<body of standardized Visa Payment>

The cash register chimes pleasantly as your payment is verified.

Response:

<response code = success>
Type: text
Len: 123

<hey, it says you got some frequent flyer miles!>

Connection disconnects, you pocket the bit-fold and head out past the stacks of dog food and charcoal briquettes. At home you set the bitfold down by your PC and press the “Reconcile” key, and it connects to your PC, sends the days purchases with itemized receipts over to the OBEX server. With a home accounting program you make short work of the financial record keeping, since the data is all entered for you, free of errors. Of course, it is harder to hide the purchase of the Twinkies from your spouse...

7.6 Create Session Followed by a Directed Connection

Client Request:	**bytes**	**Meaning**
opcode	0x87	**SESSION**, Final bit set
	0x0015	packet length = 21
	0x52	**Session-Parameters** HI
	0x0012	Length of **Session-Parameters** header (18 bytes)
	0x05, 0x01,	Client *Session Opcode* field (1-byte)
	0x00	Create Session opcode
	0x00, 0x04, 0xXXXXXXXX	Client *Device Address* field (IrDA size)
	0x01, 0x04, 0xYYYYYYYY	Client *Nonce* field (4 bytes)
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x0024	packet length of 36
	0x52	**Session-Parameters** HI
	0x0021	Length of **Session-Parameters** header (33 bytes)
	0x00, 0x04, 0xXXXXXXXX	Server *Device Address* field (IrDA size)
	0x01, 0x04, 0xYYYYYYYY	Server *Nonce* field (4 bytes)
	0x02, 0x10, 0xZZZZZZZZZZZZZZZZ ZZZZZZZZZZZZZZZZ	*Session ID* field (16 bytes)
Client Request:		
opcode	0x80	**CONNECT**, Final bit set
	0x001C	packet length = 28
	0x10	version 1.0 of OBEX
	0x00	flags, all zero for this version of OBEX
	0x2000	8K is the max OBEX packet size client can accept
	0x93	**Session-Sequence-Number** HI
	0x00	First packet sent by Client has sequence number of 0
	0x46	**Target** HI
	0x0013	Length of **Target** Header

	0x382D2BD03C3911D1 AADC0040F614953A	UUID for desired service/application {382D2BD0-3C39-11d1-AADC-0040F614953A}
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x0021	packet length of 33
	0x10	version 1.0 of OBEX
	0x00	flags (LSAP-SEL multiplexing not supported)
	0x0800	2K max packet size
	0x93	**Session-Sequence-Number** HI
	0x01	Next OBEX packet expected by Server is 1
	0xCB	**Connection Id** HI
	0x00000001	ConnId = 1
	0x4A	**Who** HI
	0x0013	Length of **Who** Header
	0x382D2BD03C3911D1 AADC0040F614953A	UUID of responding application (same value as **Target** header in request){382D2BD0-3C39-11d1-AADC-0040F614953A}

7.7 Suspend a Session

Note that **Session-Sequence-Number** headers are not used when sending Session commands in an active session.

Client Request:	**Bytes**	**Meaning**
opcode	0x87	**SESSION**, Final bit set
	0x0009	packet length = 9
	0x52	**Session-Parameters** HI
	0x0006	Length of **Session-Parameters** header (6 bytes)
	0x05, 0x01,	Client *Session Opcode* field (1-byte)
	0x02	Suspend Session opcode
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x000C	packet length of 12
	0x52	**Session-Parameters** HI
	0x0009	Length of **Session-Parameters** header (9 bytes)
	0x04, 0x04,	*Timeout* Field
	0x000004B0	Requesting timeout of 20 minutes (1200 seconds)

7.8 Resume a Session

If a session was suspended using the SUSPENDSESSION command then resuming the session does not require retransmission of OBEX packets. If the session was suspended because the underlying transport was lost it might be necessary to retransmit packets. Three cases exist. The first case is when the last client request is lost. The second case is when the last server response is lost. The third case is when the client receives the last response but the server does not know if the client has received it (packets do not need to be retransmitted in this case). In the first case the client knows the server did not receive the request because the *Next-Sequence-Number* field in the response to the RESUMESESSION command indicates this, so the client will retransmit the command. In the second and third cases the server knows what to do based on the value of the *Session-Sequence-Number* field in the first packet sent by the client.

The example below shows the sequence needed to resume the session. Retransmission of packets is not shown.

Client Request:	**Bytes**	**Meaning**
Opcode	0x87	**SESSION**, Final bit set
	0x0027	packet length = 39
	0x52	**Session-Parameters** HI
	0x0024	Length of **Session-Parameters** header (36 bytes)
	0x05, 0x01,	Client *Session Opcode* field (1-byte)
	0x03	Resume Session opcode
	0x00, 0x04, 0xXXXXXXXX	Client *Device Address* field (IrDA size)
	0x01, 0x04, 0xYYYYYYYY	Client *Nonce* field (4 bytes)
	0x02, 0x10, 0xZZZZZZZZZZZZZZZZ	*Session ID* field (16 bytes)
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x0027	packet length of 39
	0x52	**Session-Parameters** HI
	0x0024	Length of **Session-Parameters** header (36 bytes)
	0x00, 0x04, 0xXXXXXXXX	Server *Device Address* field (IrDA size)
	0x01, 0x04, 0xYYYYYYYY	Server *Nonce* field (4 bytes)
	0x02, 0x10, 0xZZZZZZZZZZZZZZZZ	*Session ID* field (16 bytes)
	0x03, 0x01, 0x05	*Next Sequence Number* field (1-byte). As an example, the server expects 0x05

7.9 Close a Session

Client Request:	**bytes**	**Meaning**
opcode	0x87	**SESSION**, Final bit set
	0x001B	packet length = 27
	0x52	**Session-Parameters** HI
	0x0018	Length of **Session-Parameters** header (24 bytes)
	0x05, 0x01,	Client *Session Opcode* field (1-byte)
	0x01	Close Session opcode
	0x02, 0x10, 0xZZZZZZZZZZZZZZZZ	*Session ID* field (16 bytes)
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x0003	packet length of 3

7.10 Negotiate a Timeout

Client Request:	**bytes**	**Meaning**
opcode	0x87	**SESSION**, Final bit set
	0x000F	packet length = 15
	0x52	**Session-Parameters** HI
	0x000C	Length of **Session-Parameters** header (12 bytes)
	0x05, 0x01,	Client *Session Opcode* field
	0x04	Set Timeout opcode
	0x04, 0x04,	*Timeout* field
	0x000004B0	Requesting timeout of 20 minutes (1200 seconds)
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x000C	packet length of 12
	0x52	**Session-Parameters** HI
	0x0009	Length of **Session-Parameters** header (9 bytes)
	0x04, 0x04,	*Timeout* field
	0x00000384	Requesting timeout of 15 minutes (900 seconds)

8. OBEX Services and Procedures

The following section covers the services currently defined for use over OBEX. This list in no way represents the entire set of services eligible for use over OBEX. It is intended to provide an understanding of what services currently use the OBEX protocol and how they use it. In addition, common processes or methods, such as the "Simple OBEX **PUT**" are also covered.

8.1 Folder Browsing Service

A folder/file system is a hierarchy of folders that contain objects/files and folders. A client can browse folders and **GET** the contents. The client can also traverse the folders and **PUT** and **GET** objects into and out of the folders. Life in a folder is more permanent than life in an inbox. It is very likely that after a client has **PUT** an object into a folder it can **GET** it at a later time.

The Folder Browsing service provides access to a device's object store via OBEX. The application that provides the service is called the *folder browsing server*, the client application is called the *folder browser*. The service is built using OBEX **PUT**, **GET** and **SETPATH** commands as well as the OBEX Folder-Listing object. It uses file access methods to **PUT** and **GET** Objects to the remote object store over a directed OBEX connection. To connect to a folder browsing service the client issues an OBEX **CONNECT** with a **Target** header containing the UUID (in binary) for the Folder Browsing service to the Default OBEX server. This UUID is F9EC7BC4-953C-11d2-984E-525400DC9E09.

The availability of the folder browsing service can be determined by inspecting the OBEX capability Object for the folder browsing service. Alternatively, a connection attempt to the folder browsing UUID can be performed. There should only be one instance of a folder browsing service on the default OBEX server. You could have folder browsing using a different OBEX server but it would not be a standard service.

8.1.1 Exchanging Folder Listings

Often, an application will request a listing of the remote devices folder system, in-order to locate an object it wants or to position an object for storage. When an application requests a listing of a folder's contents, it receives a folder-listing object. The specific format of this object is discussed in chapter 9.1 The Folder Listing Object. It should be noted that any OBEX headers sent with the folder-listing object refer to the object itself and not the folder for which the object is a listing of.

The folder-listing object is exchanged as octet-sequence data in one or more OBEX **Body** headers. The boundaries of the **Body** headers have no relevance to the internal structure of the folder listing. Individual **Body** headers should be concatenated to form the complete folder listing.

8.1.1.1 Requesting A Folder Listing

To retrieve a folder listing an OBEX **GET** request is sent to the folder-browsing server. Within the **GET** request, a **Name** header is used to convey the name of the folder for which a listing is desired. A **Type** header is used to indicate to the server the type of object requested. This is necessary because the client could be requesting a folder listing, or an object located within the folder. A connection must have already been established to the folder browsing service before requests can be sent.

The method used when requesting a folder is based on the relative position of the current folder. The following is an overview describing the four methods used when requesting a folder.

1. To retrieve the ***current*** folder: Send a **GET** Request with an empty **Name** header and a **Type** header that specifies the folder object type.

2. To retrieve a ***child*** folder: Send a **GET** Request with the **Name** of the child folder in a **Name** header and a **Type** header that specifies the folder object type.
3. To retrieve the ***parent*** folder: Send a **SETPATH** with the Backup flag set, to change to the parent folder. Then send a **GET** Request with an empty **Name** header and a **Type** header that specifies the folder object type.
4. To retrieve the ***root*** folder: Send a **SETPATH** with an empty **Name** header, to change to the root folder. Then send a **GET** Request with an empty **Name** header and a **Type** header that specifies the folder object type.

A recap of OBEX headers used when requesting a folder object:

- **Name**: *(Required)* The **Name** header specifies the name of the folder for which the listing is requested.
- **Type**: *(Required)* The **Type** header conveys the type of object requested. The value of this header must be "x-obex/folder-listing" when requesting a folder listing.
- **Connection ID**: (Required) The **Connection Id** of the folder browsing service.

8.1.1.2 Responding with a Folder Listing Object

A successful response to the **GET** operation should either indicate a **SUCCESS** or **CONTINUE** response code followed by one or more **Body** headers. The following headers may also be present in the response: **Name**, **Size** and **Time**. When present these headers describe the folder-listing object that is being retrieved.

- **Name**: *(Optional)* The **Name** header specifies the name of the folder for which the object is a listing. This should match the name present in the request, but is not required to.
- **Size**: *(Optional)* The **Size** header represents the total size of the folder-listing object in bytes.
- **Time**: *(Optional)* The **Time** header is used to convey the ISO **Time** that the folder-listing object was last updated.
- **Body / End-Of-Body** *(Required if successful)* The **Body** and **End-of-Body** headers are used to return the folder-listing object.

8.1.2 Navigating Folders

This section reviews the methods used for navigating a remote object store. The concepts are the same as those presented elsewhere in the OBEX specification. They are discussed here simply to make the discussion more complete.

8.1.2.1 Changing Folders

The OBEX **SETPATH** operation is used to change folders. A **Name** header is used for downward navigation and the Backup flag is used for upward navigation. To change to the root folder a **SETPATH** operation with an empty **Name** header is sent.

8.1.2.2 Creating Folders

The OBEX **SETPATH** operation is used to create folders in the same manner as it is used to traverse them. Implementations usually support a "create if non-existent" behavior for the **SETPATH** operation. In this way, entire trees of objects can be moved from one device to another.

8.1.3 Security

Anytime a device exposes its internal object store to a remote user, security is of some concern. This specification does not address any specific security measures because such behavior is highly device and operating system dependent. It is recommended that any application that provides access to its object store take necessary precautions to protect the privacy and consistency of the local object store.

8.2 Simple OBEX Put file transfer (+ SetPath)

This process transfers an object or a set of objects to the receiving device's inbox. The sending device has no understanding of what was done with the object after it was received. The **PUT** command is used with **Name** and optionally **Type** headers. An OBEX connection is required for compatibility with existing applications and is also used to increase transfer performance. This service supports file level access.

The QuickBeam and IrXfer applications are both based on this model of simple file transfer. These applications do not use directed OBEX connections, all objects go to the "inbox". The QuickBeam application provides a basic implementation of an inbox server, storing received objects on the file system in an inbox folder. The QuickBeam and IrXfer clients push generic file objects, using the headers and methods discussed in section 9.2 Generic File Object.

Applications exist which use the simple push file transfer model in conjunction with the **SETPATH** command to transfer an entire folder to the receiving device's inbox. This process is more complex than the simple push model because it implies that the sender has some knowledge of how the receiver will process or organize the objects. This creates difficulties for the receiving application and future applications should avoid sending the **SETPATH** command to the Inbox. On the receiving side, server applications should make an effort to permit the **SETPATH** operation to the extent possible.

Note that this process is not defined as a service because it is just a use of the inbox service. It is simply a description of existing implementations of an inbox server.

8.3 Telecom/IrMC Synchronization Service

This service allows a client to communicate with a server in a predefined manner so that the client can provide synchronization and backup services for the server. This service uses the OBEX **PUT** and **GET** commands as well as IMC vCard, vCalendar objects and other IrMC specifically defined objects. Refer to the IrMC specification for a complete object description. This service uses both File and Database access methods.

The IrMC server is targeted by appending the string, "telecom/", in a **Name** header before all object names. This service is organized by 4 levels of operation, with each successive level increasing in functionality. The levels 1, 2 & 3 do not use a dedicated OBEX connection and all requests are sent to the default OBEX server's inbox. Devices that support the IrMC service must handle this special case by redirecting all such requests to the IrMC service. Level 4 uses a directed connection to target the service (UUID, "IRMC-SYNC") via the default OBEX server.

Both simple **PUT** file transfer and IrMC sync use the default OBEX server's inbox service and they both use **PUT** with the **Name** header. They are able to distinguish the target because IrMC uses names with "telecom/" while simple **PUT** file transfer names do not contain paths.

8.4 OBEX Get Default Object

This process enables a device to request an object from a device's inbox. Typically, the objects available for request using this method will be few. Currently only business cards and the capability object have been identified for use with this method. The purpose of providing this capability is to enable a device to request basic information from a second device, instead of requiring that the second device initiate the exchange.

The procedure for requesting a default object is to send a **GET** command with a **Type** header specifying the MIME type of the desired object and an empty **Name** header. In this situation an empty **Name** header should be treated the same as no **Name** header. If a default object of the **Type** requested is available, it should be returned in one or more **Body/End-of-Body** headers in a successful response. If the object is not available, the response code NOT FOUND should be returned. If the request contained a value in the

Name header then the request is illegal (unless it is for the capability service) and FORBIDDEN should be returned.

8.4.1 Get default vCard example

In this example, the client connects to the default OBEX server and requests the devices default vCard. The device contains a default vCard and returns it successfully.

Client Request:	**bytes**	**Meaning**
opcode	0x80	**CONNECT**, Final bit set
	0x0007	packet length
	0x10	Version 1.0 of OBEX
	0x00	flags, all zero for this version of OBEX
	0x2000	8K is the max OBEX packet size client can accept
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x0007	packet length
	0x10	Version 1.0 of OBEX
	0x00	no flags
	0x0800	2K max packet size
Client Request:		
opcode	0x83	**GET**, Single packet request, so final bit set
	0x0018	packet length
	0x01	HI for **Name** header
	0x0003	length of empty **Name** header
	0x42	HI for **Type** header
	0x0010	Length of **Type** header
	text/x-vCard	Type of object
Server Response:		
response Code	0xA0	SUCCESS, Final bit set
	0x0210	length of response packet
	0x49	HI for **End-of-Body** header
	0x020D	522 byte vCard length
	0x……..	default vCard

8.5 Capability Service

The OBEX capability service is designed to provide a general-purpose method for accessing service information with OBEX. Traditionally, OBEX services and applications have derived private methods for handling the device and service information that is of interest to them. However a broader scope of services could interoperate if there were a more common way to access this information. For example, an IrMC device represents the vCard fields it can handle, in a file called “telecom/pb/info.log”. This is fine for IrMC devices because they see eye-to-eye, but a general-purpose application has no way of knowing how to retrieve this information. Additionally, different applications that want to support vCards may choose the same representation for acceptable vCard fields but will unlikely provide it in a file called “telecom/pb/info.log”. The capability service aims to bridge this gap by providing a simple access methods for retrieving commonly used information from a device.

8.5.1 The Capability Object/Database

The capability object is designed to provide a single place where general-purpose device, service and application information can be stored. Collecting this information in one place allows an application to easily retrieve a host of interesting device specific information. The capability object contains object

information such as what objects the device supports and which ones can be **PUT** to the inbox. It contains general-purpose information such as the devices serial number and manufacturer. It also contains a specification of service connection procedures for the various services available on a device. Such as, **CONNECT Target** Header values and Alternate LSAPs where the service is available.

The standardization of the capability object and its access methods allows a variety of services to retrieve common information in a common manner. New services will know how to advertise their object and service information and will not need to derive a private method. In addition, existing general purpose services will be able to locate and retrieve information about new services in the same manner that they retrieve existing service information. This will not only allow a more diverse set of OBEX applications to exchange objects but it will also allow more applications to be installed on a device because they'll use fewer resources.

There are three main sections in the Capability Object they are; General Information, Supported Objects and Service/Application-Info. The General section contains information about the device, which is likely to be of general interest. Information such as serial number and manufacturer are contained in this section. The Supported Objects section is separated into two sub-sections. The first section, the Inbox, lists the objects that are recognized by the device's inbox. This allows a connected device to determine if the recipient will accept the object it wishes to send before it initiates the transmission. The Supported Objects section provides information about other objects that are used in the device but are not permissible in the inbox. The Service Info section is designed for use by applications that need to convey static configuration information. Information such as application version and supported options is recorded here

8.5.2 The Object Profile Database

The object profile database works in conjunction with the capability object to provide information about individual objects. It is designed to allow a client to retrieve specific information about the level of support that is provided for a specific object. The server builds an *object profile* for each object that the device wants to support in the object database. An object profile contains the same information as the IrMC protocol provides in the info.log file, object fields section. Under IrMC, this vital information is hidden from most services because of its obscure location. By supplying this information in a service independent manner, the information is more useful.

Each *object profile* must have a specified format that is self-supporting. No knowledge of the supporting service can be required to understand the object profile. The method used by IrMC to describe vCards is valid under this criterion. In order to be useful to a variety of applications, the object profile must be generally available. This document is the repository for all object profiles. In addition, a method for defining the access method for each object profile must be available.

Object profiles are accessed in a very specific manner. By providing a common method, new objects can easily be added to the database. The capability service is responsible for providing the object profiles. Requests are made of the capability service, by sending the MIME type of the object for which the profile is requested. Since the capability service is designed to support many different types of objects, the type of object requested is also provided. In this case, it is an object profile object, which has the type "x-obex/object-profile".

8.5.3 Locating the Capability Service

Before a request for the capability object or an object profile can be made the application must connect to the capability service. Because the capability service is integral to the OBEX protocol it was designed to be located in the default OBEX server, along side the Inbox service. Therefore the capability service is connected to by sending a **CONNECT** packet with no targeting information. This is the same method used to connect to the inbox. The OBEX server can differentiate between requests sent to the capability service from that of the inbox because the Type header of requests to the Capability service will contain

either "x-obex/capability" or "x-obex/object-profile". In addition, GET requests to the Inbox service are not allowed to contain any Name header.

Object profiles are a part of the capability service. Therefore, the same connection that is used to retrieve the capability object is used for retrieving object profiles. The server can differentiate between the requests by the **Type** header sent with the request.

Since IrMC also uses the default OBEX server with no targeting information, it is necessary to discuss how capability service transactions are differentiated from IrMC transactions. There are two significant differences that IrMC devices can use to differentiate IrMC from Capability Service requests. All Capability Service requests will contain a **Type** header with the value “x-obex/object-profile” or “x-obex/capability”. Neither of these values are valid in IrMC. However, if the **Type** header is ignored by the server, the **Name** header also provides uniqueness. All **Name** headers used by IrMC start with “telecom/”. Since “telecom/” is not a valid root MIME type, the Capability Service naming convention will not conflict with the IrMC namespace.

8.5.3.1 Requesting the Capability Object

The capability object is requested by sending a **GET** request with the MIME type of the capability object provided in an OBEX **Type** header. The MIME type of the capability object is “x-obex/capability”. A successful response will contain one or more OBEX **Body** headers with the full capability object as their contents.

8.5.3.2 Requesting an Object Profile

The format for an object profile object request is based on the MIME type of the object for which the profile is requested. A **GET** request containing the MIME type of the object whose profile is desired in a **Name** header, with a **Type** header containing the value “x-obex/object-profile” is interpreted by the capability service as an object profile request. A successful response will contain one or more **Body** headers with the object profile object contents.

If a client does not know the MIME type of an object it can look-up the object in the capability object using the name extension. This record also contains the MIME type of the object, which can then be used to make an object profile object query. Below is an example **GET** sequence for the vCard object profile.

Client Request:		
Opcode	0x83	**GET**, Single packet request, so final bit set
	0x003B	packet length
	0x42	HI for **Type** header
	0x0019	length of **Type** header
	x-obex/object-profile	MIME type of an object profile object
	0x01	HI for **Name** header
	0x001F	length of **Name** header (Unicode)
	text/x-vCard	MIME type of the object for which a profile is requested
Server Response:		
response code	0xA0	SUCCESS, Final bit set
	0x012C	length of response packet
	0x49	HI for **End-of-Body** header
	0x0129	294 byte object length
	0x…..	vCard object profile

8.5.3.3 Sending an Object Profile (unilaterally)

In some application environments, it may be necessary for a client to inform the server of its object profile. This is the reverse of the service offered by the capability service. This can be the case when the client is requesting objects from the server and wishes for them to adhere to the object profile that it supports.

To fulfill this requirement, an object profile can be **PUT** to an application or service (NOT the capability service). The service receiving the object profile can ignore and discard the object if it does not support customized object responses. In this case, the **PUT** request should be failed by responding with a non-success OBEX response code. If the service supports customized object responses then it should process the object profile and apply its rules to any request for that objects type. The rules should be applied for the duration of the connection or until another object profile is received. To **PUT** an object profile the client sends a **PUT** request with the same headers as a **GET** request, a **Type** header of "x-obex/object-profile" and a **Name** header with the object profile's MIME type.

Client Request:		
Opcode	0x82	**PUT**, Single packet request, so final bit set
	0x0164	packet length
	0x01	HI for **Name** header
	0x001F	length of **Name** header
	text/x-vCard	name of object profile object (Unicode, null-terminated)
	0x42	HI for **Type** header
	0x0019	length of **Type** header
	x-obex/object-profile	MIME type of an object profile object
	0x49	HI for **End-of-Body** Header
	0x0129	294 byte object length
	0x……	vCard object profile

One instance where this is useful is when accessing the folder browsing service. In this service, the client **GET**s folder-listing objects from the server. In order to convey its desire to receive only limited information in the returned object the client sends a folder-listing object profile to the server before the first folder-listing request. In this manner, the server can avoid overwhelming the client and wasting bandwidth by eliminating information that the client is not interested in receiving.

9. OBEX Objects

This section contains definitions for objects, which have been created or adopted for use within the OBEX protocol and its applications. Whenever possible industry standard objects are used, such as vCards, etc. However occasionally the need arises to create an object specific to OBEX. Such objects are the topic of this chapter. The following is a list of the objects covered:

- The Folder Listing Object
- The Generic File Object
- The Capability Object
- The Object Profile Object

Often when creating a new OBEX object, the problem of assigning a MIME type for use with the OBEX **Type** header arises. It is recommended that the following approach be used to determine the MIME type for an object.

- If the object's domain is outside of OBEX it should be given an experimental MIME type under one of the existing MIME root types. A vCard is an example of such an object, it is rooted in the "text" type. These types must follow the rules for creating an experimental MIME type, specifically they must start with "x-". The MIME type for vCards is "text/x-vCard".
- If the object was created for use within OBEX than the type should be rooted in the OBEX experimental root type "x-obex". The capability and folder-listing objects, which have MIME types "x-obex/capability" and "x-obex/folder-listing", are examples.

9.1 The Folder Listing Object

A folder-listing object is a detailed itemization of the contents of a particular folder. Each object in a listing can contain a variable amount of information, conveyed as attributes of the object. A folder can be a folder or other similar type of container object. It is expressed as an application of the Extensible Markup Language (XML) specified by the W3C. Folder object data is exchanged in one or more OBEX **Body** headers. An example of a folder listing conforming to the OBEX folder object format is shown below.

```
<folder-listing>
    <folder name = ”System” created = ”19961103T141500Z”/>
    <file name = ”Jumar.txt” created = ”19971209T090300Z” size = ”6672”/>
    <file name = “Obex.doc” created = ”19970122T102300Z” size = ”41042”/>
</folder-listing>
```

Folder-listing objects are provided by a Folder Browsing service. This service is designed to provide access to the folder/file system of the serving device. The folder system is represented as a hierarchy of folders that contain objects/files and sub-folders. A client can browse folders and **GET** their contents. The folder browsing service defines the methods used to traverse folders, as well as for storing and retrieving objects.

9.1.1 Element Specification

9.1.1.1 Document Definition

The syntax for describing OBEX folder objects is based on the XML specification. A folder description is written using the syntax of XML because it provides for flexible data tagging that fits the needs for an open, flexible and general-purpose way of describing objects. An OBEX folder listing is a set of various XML elements. Collectively these elements form an XML document, referred to as the folder listing.

A folder-listing document may contains zero or more of the following elements:

- **folder** – A description of a folder type object.
- **parent-folder** – Indicates the existence of a parent folder.
- **file** – A description of a file type object.

These elements may also contain attributes that describe the file or folder element.

9.1.1.2 Elements of a Folder Listing Document

Each object in a listing is described by an element with attributes. This section details the syntax and use of the defined elements. For more detailed information on the XML syntax and encoding rules, refer to the XML specification available at http://www.w3.org/TR/REC-xml.

9.1.1.2.1 Elements

Element Name	Meaning
file	Indicates the existence of a file contained within the folder.
folder	Indicates the existence of a folder contained within the folder.
parent-folder	Represents the existence of a parent to the folder being presented.
folder-listing	The document type.

9.1.1.2.2 Attributes used in File and Folder Elements

Attribute Name	Meaning
name	The name of the folder or file. It does not include any path or similar information. The value of this attribute is used in a **NAME** header to retrieve the object if desired. This value must be unique in all similar elements in the document. The current folder may be represented by a folder element with the name ".". For example: "<folder name=".">".
size	The size of the folder or file object in bytes. This size is an estimate and is not required to be exact. It is expressed as an unsigned base 10 integer.
modified	This attribute represents the last modified time for the object. It is expressed in the same format used by the OBEX ISO **Time** header. That format is YYYYMMDDTHHMMSS, where the capital letter 'T' is expressly inserted between the day and the hour fields. It is recommended that whenever possible UTC time be used. When expressing UTC time, the letter "Z" is appended to the end (for example: 19670110T153410Z).
created	This attribute represents the creation time for the object. It is expressed in the same format used by the OBEX ISO **Time** header. It is recommended that whenever possible UTC time be used.
accessed	This attribute represents the last accessed time for the object. It is expressed in the same format used by the OBEX ISO **Time** header. It is recommended that whenever possible UTC time be used.

user-perm **group-perm** **other-perm**	These attributes convey the access permissions for the object. The permissions are encoded based on the access currently available to this user over the object listed. The following alphabetic characters are used to describe access: **"R", "W", "D"** The value of the permissions type is an unordered sequence of these alphabetic characters. The permissions indicators are case independent. R: The READ permission applies to all object types. It indicates that an attempt to **GET** the named object should successfully retrieve its contents. D: The DELETE permission applies to file types. It indicates that the file may be removed by sending a PUT-DELETE command. W: The WRITE permission applies to all object types. It indicates that an attempt to modify the contents of the file by **PUT**'ing to the file should succeed. For folder objects it indicates that attempts to create a folder or other object within that folder should succeed. There are three levels of permissions. These are for systems that distinguish between access for the user, the group and other. The default permissions attribute is user. Note: A permission indicator does not imply that the appropriate command is guaranteed to work – just that it might. Other system specific limitations, such as limitations on available space for storing objects, may cause an operation to fail, where the permission flags may have indicated that it was likely to succeed. The permissions are a guide only. Some systems may have more specific permissions than those listed here, such systems should map those to the flags defined as best they are able.
owner	The Owner attribute is used to convey the user associated with ownership or responsibility for this object.
group	Some file systems have the notion of group ownership. This attribute is used to convey that information when present
type	This attribute works similarly to the OBEX **Type** header and expresses the MIME type of the file object. It can be used to interpret the files' internal format or an application association.
xml:lang	The XML defined language attribute may be used to specify the language of both content and attribute values. The default language of English (us-EN) need not be specified. This attribute affects the interpretation of the following attributes: **name**, **owner** and **group**. As well as any element content.

9.1.1.2.3 Attributes used with the Folder Listing Element

Attribute Name	Meaning
version	The version attribute is used in the folder-listing element to convey the version of the folder listing DTD that the document conforms to. The current version is 1.0.

9.1.1.2.4 File and Folder Element Content

Both file and folder elements may contain element content. When present, the content expresses the recommended display name of the file or folder. This differs from the name attribute in that it does not need to be unique or be a valid file/folder name. In the absence of this content, the name attribute should be used when displaying the element.

9.1.2 Folder Listing Details

9.1.2.1 The Folder Object Type

When requesting a folder listing, the **Type** header must contain the value “x-obex/folder-listing”. By specifying this type, ambiguities about the type of document requested can be avoided. This value is not case sensitive. It is not necessary to send the **Type** header in the response.

9.1.2.2 Empty Folder Listing Objects

When a request for a folder listing results in an empty listing, the response should follow the behavior of any successful request. The folder-listing object contained in the response **Body** header should be a valid XML empty object. Here is an example of such an object:

```
<?xml version="1.0"?>
<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd">
<folder-listing version=”1.0”/>
```

9.1.2.3 Customizing Folder Listings

A folder listing object profile can be used to provide information about the elements and attributes supported by the folder browsing server or browser. If available, the object profile should be registered with the capability service under the name “x-obex/folder-listing”. The object profile is an empty XML folder object. It lists all the supported elements and attributes in their regular format but with no values. If a client browser wishes to inform the server of its object profile it should **PUT** the object profile to the server before it performs the first **GET** of a folder listing.

Adherence to a received object profile is not required. However, it is recommended that devices, which, by default provide detailed responses, support this feature. It is most useful in the case where a simple device is querying a more robust device and it doesn’t want to get flooded with information. For example, if a PDA chooses to browse a desktop device it might want only a list of file and folder *names* and *sizes.* Below is an example **PUT** of an object profile for an application that only recognizes name and size attributes. Note that the syntax can be completely parsed by an XML processor.

Operation	Header	Content
PUT	**Name**	x-obex/folder-listing
	Type	x-obex/object-profile
	Body	<?xml version="1.0"?> <!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd"> <folder-listing version=”1.0”> <parent-folder /> <folder name=”” size=”” /> <file name=”” size=”” /> </folder-listing>

9.1.3 Encoding Folder Listing Objects

9.1.3.1 XML Basics

The definition of a folder-listing object is based on the W3C Specification of XML. XML is used because it provides the structure and syntax for the folder-listing object. For a folder-listing object to be correct, it must follow the syntax rules specified by the XML specification. In XML, the content and organization of a particular object type is expressed by its Document Type Definition (DTD). In order for folder-object to be correct, it must also adhere to the obex-folder-listing DTD. This DTD is provided in section 9.1.4.1 of this document.

XML allows the DTD to be either internal or external to the document that relies on it. The folder object DTD is always external and is never sent as part of the folder object. It is assumed that the receiving entity will be able to handle the object without an internal DTD.

The content of an XML document is contained in its elements and attributes. This document defines the elements and attributes used in folder-listing objects. The XML specification states that element and attribute names are case sensitive. Therefore, the element names “Parent-Folder”, “parent-folder” and “PARENT-FOLDER” all refer to different elements.

9.1.3.2 Character Encoding Format

An XML document allows for the specification of the character encoding used in the document. The encoding declaration is positioned in the very beginning of the document to enable quick determination. The default encoding for XML documents is UTF-8 and need not be specified. If an alternate encoding is used it must be specified in the encoding declaration.

It is recommended that folder-listing objects which contain Japanese characters be encoded using the SHIFT_JIS encoding. This encoding has the benefit of ensuring that the characters of ASCII have their normal values, which makes the processing of the encoding declaration straightforward. The following example illustrates an XML declaration for a SHIFT_JIS encoded document.

<?xml encoding=”SHIFT_JIS”?>

9.1.3.3 Folder-Listing Object Examples

The following listing is used to illustrate the encoding of a folder into a folder-listing object.

Name	Size	Creation Time (UTC)		Last Modified (UTC)		Type
	--					parent folder
System	--	Nov 3,1996	2:15p	Nov 3, 1996	2:15p	folder
IR Inbox	--	Mar 30, 1995	10:50a	Mar 30, 1995	10:50a	folder
Jumar.txt	6,672	Dec 9, 1997	9:03a	Dec 22, 1997	4:41p	text/plain
Obex.doc	41,042	Jan 22, 1997	10:23a	Jan 22, 1997	10:23a	application/msword

9.1.3.3.1 Detailed encoding of Example Data

```
<?xml version="1.0"?>
<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd">
<folder-listing version=”1.0”>
    <parent-folder />
    <folder name = “System” created=”19961103T141500Z”/>
    <folder name =” IR Inbox” created=”19950330T105000Z”/>
    <file name = “Jumar.txt” created=”19971209T090300Z” size=“6672”
          modified=“19971222T164100Z” user-perm=”RW”/>
    <file name =”Obex.doc” created=“19970122T102300Z” size = “41042”
          type=”application/msword” modified=“19970122T102300Z”/>
```

```
</folder-listing>
```

9.1.3.3.2 A Simpler encoding illustrating display names

```
<?xml version="1.0"?>
<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd">
<folder-listing version="1.0">
    <parent-folder />
    <folder name="System"/>
    <folder name="IR Inbox"/>
    <file name="Jumar.txt" size="6672">Jumar Handling Guide</file>
    <file name ="Obex.doc" type = "application/msword">OBEX Specification v1.0</file>
</folder-listing>
```

9.1.3.4 Simple Encoding Methods

In the simplest case, devices that want to provide a listing of the objects available on the device can construct a static representation of the folders. If the object list can dynamically change, the application providing the listing may need to be more adaptive. In general, the construction of an XML folder-object is simple and straightforward. The first three lines of the document are generally static text. The following encoding of the attributes for the various files and folders will generally be a consistent straightforward format.

9.1.3.5 Room for Future Extensions

The folder-listing element contains an attribute that specifies the version of the DTD that was used to construct the object. If the future necessitates changes to the folder-listing DTD, the version information can then be used to indicate what revision of the DTD the object was based on. Since it is intended that folder-listings be exchanged without an internal DTD, it is essential that different versions of the DTD be readily discernable.

9.1.4 XML Document Definition

9.1.4.1 DTD Specification

```
<!-- DTD for the OBEX Folder-Listing Object -->

<!ELEMENT folder-listing ( folder | file | parent-folder )* >
<!ATTLIST folder-listing version CDATA "1.0">

<!ELEMENT file (#PCDATA)>
<!ATTLIST file name CDATA #REQUIRED>
<!ATTLIST file size CDATA #IMPLIED>
<!ATTLIST file type CDATA #IMPLIED>
<!ATTLIST file modified CDATA #IMPLIED>
<!ATTLIST file created CDATA #IMPLIED>
<!ATTLIST file accessed CDATA #IMPLIED>
<!ATTLIST file user-perm NMTOKEN #IMPLIED>
<!ATTLIST file group-perm NMTOKEN #IMPLIED>
<!ATTLIST file other-perm NMTOKEN #IMPLIED>
<!ATTLIST file group CDATA #IMPLIED>
<!ATTLIST file owner CDATA #IMPLIED>
<!ATTLIST file xml:lang NMTOKEN #IMPLIED>

<!ELEMENT folder (#PCDATA) >
<!ATTLIST folder name CDATA #REQUIRED>
<!ATTLIST folder size CDATA #IMPLIED>
<!ATTLIST folder modified CDATA #IMPLIED>
<!ATTLIST folder created CDATA #IMPLIED>
<!ATTLIST folder accessed CDATA #IMPLIED>
```

```
<!ATTLIST folder user-perm NMTOKEN #IMPLIED>
<!ATTLIST folder group-perm NMTOKEN #IMPLIED>
<!ATTLIST folder other-perm NMTOKEN #IMPLIED>
<!ATTLIST folder group CDATA #IMPLIED>
<!ATTLIST folder owner CDATA #IMPLIED>
<!ATTLIST folder xml:lang NMTOKEN #IMPLIED>

<!ELEMENT parent-folder EMPTY>
```

9.2 Generic File Object

9.2.1 Introduction

The purpose of this chapter is to outline a method for the exchange of a basic file object [blob] from one device to another. While the OBEX protocol provides many useful headers and capabilities for exchanging objects. This section is designed to provide the reader with an overview of the basic structure used when exchanging generic file data. The file object is assumed to be sent by a generic client to the receiving device's inbox. In a manner consistent to the way QuickBeam and IrXfer applications work. Based on these assumptions, the file exchange should follow the guidelines shown here for headers used and response codes expected.

9.2.2 Commonly Used Headers

- **Name:** *(Required)* This header is used to convey the full name of the object to exchange. The **Name** header should not contain any path information. If it is necessary to specify a path, it should be done with the **SETPATH** command. All information in this header is interpreted as the objects name.
- **Length:** *(Recommended)* This header is used to convey the size of the file object in bytes. This can be used for verification of storage space requirements. The sum of the bytes passed in **Body** headers should be this many bytes (but is not guaranteed).
- **Body/End-of-Body:** *(Required)* **Body** headers are used to transfer the file data itself. They are sent repeatedly as long as file data exists to be exchanged. The **End-of-Body** header is used to indicate to the receiving application that this is the last piece of the file data. Applications frequently close the file and consider the transfer complete upon reception of this header. The data contained in the **Body/End-of-Body** headers should be concatenated to form the file.
- **Time:** *(Recommended)* This header is used to exchange the time that the file was last modified. Current file transfer applications use both the simple 4-byte **Time** header and the ISO format. It is recommended that all applications use the more robust ISO **Time** header. For backward compatibility it should be noted that the "IrXfer" application provided by Microsoft only accepts the 4-byte **Time** header. Attempts to send the ISO header will result in the rejection of the operation.
- **Type:** (O*ptional)* The MIME type of the file object. This is not often used but can be helpful in some cases to identify the application to associate with the file object.

9.2.3 Response Codes Commonly Used in File Exchange

The following is a list of the known OBEX Response codes that are used during File Object Exchange.

- CONTINUE
- SUCCESS
- BAD REQUEST
- UNAUTHORIZED
- NOT FOUND
- INTERNAL SERVER ERROR

9.2.4 Example Put Exchange

An application wishes to **PUT** a 2000 byte file named "**Test File Object**" with the time stamp 0x41a50016 (Jan 9, 1992, 11:02:00 UTC). The OBEX packet size is 512 bytes.

	Opcode or Response plus Headers	Final bit	Header Data	Header Length	Running Total of Data Exchanged
Request →	**PUT** **Name** **Length** **Time**		 "Test File Object" "2000" "19920109T110200Z"	 37 19	
Response ←	CONTINUE no headers	✓			
Request →	**PUT** **Body**		"start of file data.."	509	506 bytes
Response ←	CONTINUE no headers	✓			
Request →	**PUT** **Body**		"..continuation of file data.."	509	1012 bytes
Response ←	CONTINUE no headers	✓			
Request →	**PUT** **Body**		"..continuation of file data.."	509	1518 bytes
Response ←	CONTINUE no headers	✓			
Request →	**PUT** **Body**		"..final segment of file data"	485	2000 bytes
Response ←	CONTINUE no headers	✓			
Request →	**PUT** **End-of-Body**	✓	Empty end of body header.	3	2000 bytes
Response ←	SUCCESS no headers	✓			

9.3 The Capability Object

The syntax for the capability object is based on XML. XML was chosen because of its flexibility and ease with which elements can be categorized and described. The DTD of the Capability Object can be found in section 9.3.6 Capability Object DTD and an example Capability Object is shown in the section 9.3.7 Capability Object Example. The Capability Object is flexible enough that individual manufacturers can

include objects, services and extension elements that are of interest to them. The extension element (see 9.3.2.10 Extension (Ext)) permits the addition of user-defined parameters for specific application or manufacturer needs.

9.3.1 The Capability Container (Capability)

This element is the root element containing all other elements of the Capability Object. It has one attribute called **Version**. When the XML document conforms to this revision of the Capability Object specification, the Version attribute MUST have the value “1.0”.

The Capability container contains a General Information Container (see 9.3.2 General Information Container (General)), an optional Inbox Container (see 9.3.3 Inbox Container (Inbox)) and a list of services and applications (see 9.3.4 Service/Application List), which are supported by the device.

9.3.2 General Information Container (General)

The General Information Container is used to hold information that is specific to the device, which is hosting the Capability Object. This information may be used by a variety of services and applications and is therefore collected in the general section to make it easily available. The following is a description of the items that may be present in the general section.

9.3.2.1 Manufacturer (Manufacturer)

The manufacturer element is used to identify the vendor that builds the hardware device. This information can be handy when identifying the device to the user. It is a string value that shows the full name of the manufacturer. This value is **required**.

9.3.2.2 Model Name (Model)

The model name element is used to identify the model name and/or number assigned to the device by the manufacturer. Again, the information is handy when identifying the device to the user. This string value conveys the full model name of the device. This value is **required**.

9.3.2.3 Serial Number (SN)

The serial number is used to uniquely identify the device. This number should be assigned by the manufacturer in a globally unique method. The serial number does not have to be a UUID. If the manufacturer and model number are combined with the serial number you can get a universal unique ID. The format of the serial number is specified by the manufacturer, but it must be unique within a specific device model. This value is **optional**.

9.3.2.4 Original Equipment Manufacturer (OEM)

The OEM is used to specify the Original Equipment Manufacturer of the device. This value is **optional**.

9.3.2.5 Software Version (SW)

The software version specifies the software version of the device. The **Version** attribute specifies the version identifier and the **Date** attribute specifies the date of the SW version of the device. The date MUST be formatted as a complete representation, in the basic format of a [DATES] date or date and UTC time of day, for example, 19980114 or 19990714T133000Z. Only hours, minutes and seconds MUST be specified in the time component. Either **Version** or **Date** attribute should be present. This value is **optional**.

9.3.2.6 Firmware Version (FW)

Specifies the firmware version of the device. See 9.3.2.5 Software Version (SW) for more details. This value is **optional**.

9.3.2.7 Hardware Version (HW)

Specifies the hardware version of the device. See 9.3.2.5 Software Version (SW) for more details. This value is **optional**.

9.3.2.8 Currently Selected Language (Language)

Specifies the currently used language of the device.. The language MUST be represented by the two-letter lower-case language symbols defined by ISO [LANG], for example English would be represented as “en”. This value is **optional**.

9.3.2.9 Memory Record (Memory)

Describes the available memory types and sizes. Also may contain device specific memory limitations like maximum file name length. This element is **optional**.

MemType	Describes the memory type. Values are manufacturer specific. This value is **optional**.
Location	Describes the logical memory location (for example, drive “c:\” or path name). This value is **optional**.
Free	Defines the free user memory in bytes. This value is **optional**.
Used	Defines the used user memory in bytes. This value is **optional**.
Shared	Boolean value indicating whether the user memory is shared between applications. A missing element indicates the memory is not shared. This value is **optional**.
FileSize	Specifies the maximum accepted file size in bytes. This value is **optional**.
FolderSize	Specifies the maximum accepted folder size in bytes. This value is **optional**.
FileNLen	Specifies the maximum accepted file name length in characters. If the maximum folder name length element (FolderNLen) is not present the maximum file name length indicates the combined length (file name and path together). This value is **optional**.
FolderNLen	Specifies the maximum accepted folder name length in characters. This value is **optional**.
CaseSenN	Specifies whether the device supports case insensitivity or case sensitivity of file and folder names. A missing element indicates case insensitive names. This value is **optional**.
Ext	See 9.3.2.10 Extension (Ext). This value is **optional**.

9.3.2.10 Extension (Ext)

Specifies a non-standard, experimental or manufacturer specific extension supported by the device. The extension is specified in terms of the XML element type name and a value. Extension elements are **optional**.

XNam	Specifies the name of an extension element. This value is **required**.
XVal	Specifies the value of an extension element. This value is **optional**.

9.3.3 Inbox Container (Inbox)

The Inbox container contains all the types of objects that the device will accept in its inbox. An object type is identified by its MIME type or name extension listed in the Inbox object record. This list of objects can always include the object of type “ANY” to indicate that any object can be **PUT** to the Inbox. The Inbox can contain additional extension elements if needed, and is an **optional** element.

9.3.3.1 Object Record (Object)

An object record indicates the support for a particular object type. The Inbox container may contain zero or more objects. Each Inbox object must contain at least a **Type** or **Name-Ext** value. It is highly recommended that both values be present when available. The **Ext** element can be used to create additional attributes to the object. This element may contain one or more of the following attributes.

• **Type -** The MIME type of the object. This value is **conditionally required**.

• **Name-Ext -** The generally accepted filename extension used for this object format. This value is **conditionally required**. An object can contain zero or more Name-Ext elements.

• **Size -** The maximum accepted object size in bytes. This value is **optional**.

• **Ext -** Generic extension containing a name-value pair (see 9.3.2.10 Extension (Ext)). An object can contain zero or more Ext elements. This value is **optional.**

9.3.4 Service/Application List

The services and applications supported by the device are listed in the capability object. The service/application specific information of each service/application is presented inside of a service record. The service record contains all the service objects supported by the service/application.
Additionally, special information regarding application access methods is placed here. Applications are not required to have representation in the service/application list. The service/application list may contain zero or more service records.

9.3.4.1 Service/Application Record (Service)

One service record exists for each service that wishes to provide service specific information in the capability object. The service records can be identified by the **Name** of the service or its **UUID** but at least one of them has to be present. The information contained in each service record is governed by the service itself. This allows for the specification of service specific information. The following information may be present in any Service record.

• **Name -** The name of the service/application. This value is **conditionally required**.

• **UUID -** The UUID of the service/application. This value is **conditionally required**.

• **Version -** The version number of the service/application. This value is **optional**.

• **Object -** The format for a service object is the same as that of an inbox object in chapter 9.3.3.1 Object Record (Object). A service record may contain zero or more service objects. This value is **optional**.

• **Access -** The access method for the service. The attributes for this element are shown below. If not present, it can be assumed that the service is accessible via the standard targeted connection process used in OBEX. This value is **optional**.

Protocol	The transport protocol used to communicate with this

	service. Possible values are “TCP” and “IrDA”. This value is **required**.
Endpoint	Contains the endpoint to which the client can connect to communicate with this service. In IrDA, the endpoint is synonymous with a TinyTP LSAP-SEL. In TCP, it is a TCP Port number. This value is **required**.
Target	The **Target** header value used to establish a connection with this service. This value is **required**.
Ext	See (see 9.3.2.10 Extension (Ext)). This value is **optional.**

• **Ext -** Generic extension containing a name-value pair (see 9.3.2.10 Extension (Ext)). This value is **optional**.

9.3.5 Requesting the Capability Object

The capability object is requested by sending a **GET** request with the MIME type of the capability object provided in an OBEX **Type** header. The MIME type of the capability object is “x-obex/capability”. A successful response will contain one or more OBEX **Body** headers with the full capability object as their contents.

9.3.6 Capability Object DTD

```
<!ELEMENT Capability (General, Inbox?, Service*)>
        <!ATTLIST Capability Version CDATA "1.0">
<!ELEMENT General (Manufacturer, Model, SN?, OEM?, SW?, FW?, HW?, Language?, Memory*, Ext*)>
<!ELEMENT Manufacturer (#PCDATA)>
<!ELEMENT Model (#PCDATA)>
<!ELEMENT SN (#PCDATA)>
<!ELEMENT OEM (#PCDATA)>
<!ELEMENT SW EMPTY>
<!ATTLIST SW
        Version CDATA #IMPLIED
        Date CDATA #IMPLIED>
<!ELEMENT FW EMPTY>
<!ATTLIST FW
        Version CDATA #IMPLIED
        Date CDATA #IMPLIED>
<!ELEMENT HW EMPTY>
<!ATTLIST HW
        Version CDATA #IMPLIED
        Date CDATA #IMPLIED>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT Memory (MemType?, Location?, Free?, Used?, Shared?, FileSize?, FolderSize?,
FileNLen?, FolderNLen?, CaseSenN?, Ext*)>
<!ELEMENT MemType (#PCDATA)>
<!ELEMENT Location (#PCDATA)>
<!ELEMENT Free (#PCDATA)>
<!ELEMENT Used (#PCDATA)>
<!ELEMENT Shared EMPTY>
<!ELEMENT FileSize (#PCDATA)>
<!ELEMENT FolderSize (#PCDATA)>
<!ELEMENT FileNLen (#PCDATA)>
<!ELEMENT FolderNLen (#PCDATA)>
<!ELEMENT CaseSenN EMPTY>
```

```
<!ELEMENT Ext (XNam, XVal*)>
<!ELEMENT XNam (#PCDATA)>
<!ELEMENT XVal (#PCDATA)>
<!ELEMENT Object (((Type, Name-Ext*)|(Type?, Name-Ext+)), Size?, Ext*)>
<!ELEMENT Inbox (Object*, Ext*)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT Name-Ext (#PCDATA)>
<!ELEMENT Size (#PCDATA)>
<!ELEMENT Service (((Name?, UUID)|(Name, UUID?)), Version?, Object*, Access*, Ext*)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT UUID (#PCDATA)>
<!ELEMENT Version (#PCDATA)>
<!ELEMENT Access (Protocol? ,Endpoint?, Target?, Ext*)>
<!ELEMENT Protocol (#PCDATA)>
<!ELEMENT Endpoint (#PCDATA)>
<!ELEMENT Target (#PCDATA)>
```

9.3.7 Capability Object Example

```
<?xml version="1.0" ?>
<!-- OBEX Capability Object -->

<!DOCTYPE Capability SYSTEM "obex-capability.dtd">

<Capability Version="1.0">

        <!-- General Purpose information -->
        <General>

                <Manufacturer>Big Factory, Ltd.</Manufacturer>
                <Model>Mighty 4119</Model>
                <SN>1234567890</SN>
                <OEM>Jane´s Phones</OEM>
                <SW Version="2.0"/>
                <FW Version="2.0e" Date="19971031T231210"/>
        </General>

        <!-- Inbox Object Definitions -->
        <Inbox>
                <Object>
                        <Type>text/x-vMsg</Type>
                        <Name-Ext>vmg</Name-Ext>
            </Object>
                <Object>
                        <Type>text/x-vCard</Type>
                        <Name-Ext>vcf</Name-Ext>
            </Object>
                <Object>
                        <Type>image/jpeg</Type>
                        <Name-Ext>jpg</Name-Ext>
            </Object>
        </Inbox>

        <!-- Service Object Definitions -->
        <Service>
                <Name>IrMC</Name>
```

```
                    <UUID>IRMC-SYNC</UUID>
                    <Object>
                            <Type>x-irmc/info.log</Type>
                            <Name-Ext>log</Name-Ext>
                    </Object>
                    <Access>
                            <Protocol>IrDA</Protocol>
                            <Target></Target>
                    </Access>
                <Ext>
                     <XNam>PhoneBook/Support</XNam>
                     <XVal>4</XVal>
                </Ext>
                <Ext>
                     <XNam>PhoneBook/Optional</XNam>
                     <XVal>7</XVal>
                </Ext>
                    <Ext>
                     <XNam>PhoneBook/Version</XNam>
                     <XVal>1.0</XVal>
                </Ext>
                    <Ext>
                     <XNam>Messaging/Support</XNam>
                     <XVal>4</XVal>
                </Ext>
                    <Ext>
                     <XNam>Messaging/Version</XNam>
                     <XVal>1.0</XVal>
                </Ext>
            </Service>
        <Service>
                    <Name>Folder-Browsing</Name>
                    <UUID>F9EC7BC4-953C-11d2-984E-525400DC9E09</UUID>
                    <Object>
                            <Type>x-obex/folder-listing</Type>
                    </Object>
                    <Access>
                            <Protocol>IrDA</Protocol>
                            <Target>F9EC7BC4-953C-11d2-984E-525400DC9E09</Target>
                    </Access>
        </Service>
        <Service>
                    <Name>Image-X</Name>
                    <UUID>F9EC7BC7-953C-11d2-984E-525400DC9E09</UUID>
                    <Access>
                            <Protocol>IrDA</Protocol>
                            <Endpoint>7</Endpoint>
                            <Target>F9EC7BC7-953C-11d2-984E-525400DC9E09</Target>
                    </Access>
                <Access>
                            <Protocol>TCP</Protocol>
                            <Target>F9EC7BC7-953C-11d2-984E-525400DC9E09</Target>
                </Access>
            </Service>
</Capability>
```

9.4 The Object Profile Object

This section gives an overview on how to create object profile objects for your objects. Object profiles do not have to be in XML or any other common format. Each object profile developer is free to choose the most applicable format for describing the object. When complete, this format must be described in section 9 OBEX Objects.

9.4.1 Creating an Object Profile

Although all object profile objects have the same MIME type, they are not necessarily in the same format. Each object profile is defined in a format that best represents the object. For example, the vCard object profile syntax is very similar to the vCard syntax. This makes it easier for vCard oriented parsers to process the vCard object profile. The same holds true for the folder-listing object profile. This object profile is represented in XML format to match the folder-listing object format.

Whenever feasible the object profile syntax chosen should mirror the format and expressive capabilities of the object itself. In addition, a self-describing object format such as the format used by vCards and folder-objects is preferred. Whatever format is chosen, the author must add the specification of this format to this document. The MIME type of the object must also be specified to avoid possible ambiguity. The end of this document contains a section entitled Object Profiles. This is where all the object profiles available in the capability database are described.

9.4.2 Object Profiles

This section holds the definitions for each object profile supported by the capability object. Each definition must contain the MIME type and the name extension of the object. As well as, the format used by the profile object. The definitions are sorted alphabetically by object name.

9.4.2.1 Folder-Listing

MIME type	x-obex/folder-listing
Name extension	*None*
Profile format	XML, Refer to Folder-Listing Object Proposal.

9.4.2.2 vCalendar

MIME Type	text/x-vcalendar
Name extension	vcs
Profile format	TBD, Refer to the IrMC specification of X-IRMC-FIELDS definition.

9.4.2.3 vCard

MIME type	text/x-vcard
Name extension	vcf
Profile format	TBD, Refer to the IrMC specification of X-IRMC-FIELDS definition.

9.4.2.4 vMessage

MIME type	text/x-vmsg
Name extension	vmg
Profile format	TBD, Refer to the IrMC specification of X-IRMC-FIELDS definition.

9.4.2.5 UPF

MIME type	image/x-UPF
Name extension	upf
Profile format	TBD

9.4.3 Object Profile Example

This is an example of a vCard object profile. The vCard object profile is derived from the work of the IrMC group. This definition is similar to the X-IRMC-FIELDS extension property.

```
Begin: vCard-Profile
Version:
N:=20
UID:=4
ADR[1=20;2;6;7]
TEL;TYPE=HOME;WORK;
End: vCard-Profile
```

10. Session Capabilities in OBEX

A number of applications and services built on OBEX™ desire the ability to recover an OBEX session that has been broken due to loss of the underlying transport (e.g. IrLAP link loss). Examples include restarting a file transfer and resuming a sync session. The Infrared Financial Messaging (IrFM) group also has determined the need to maintain an OBEX session when the IrLAP link is lost. This section contains a proposal for providing reliable session support in OBEX. This proposal is aimed at meeting the needs of IrFM but should provide enough capabilities to meet the needs of other services and applications. A list of desired features for an OBEX session layer is given below.

- Allow an OBEX session to be resumed without loss of data when the underlying transport layer is lost.
- Allow OBEX applications and services to suspend a session and later resume it.
- Provide indications to applications and services about loss of the underlying transport connection.
- Provide the ability to automatically resume the session including restarting the lower level transports without burdening the applications or services.
- Allow existing services such as the Inbox, File transfer, IrMC, etc. to have session support.
- Allow multiple directed connections to be started and stopped within a single session.

A lower priority feature is listed below.

- Ability to start a session on one transport and move it to another transport (e.g. start the OBEX session on IrDA and move it to Bluetooth).

10.1 OBEX Session Overview

The OBEX specification divides OBEX into two parts: a model for representing objects and a session protocol. The session protocol uses a binary packet-based client/server request-response model. The session protocol defines the basic structure of an OBEX conversation and includes a set of commands to perform specific actions such as **PUT** and **GET**. The OBEX conversation occurs within the context of an OBEX connection. The OBEX session protocol runs on top of a transport layer such as TinyTP in IrDA, RFCOMM in Bluetooth or TCP. Currently, when the underlying transport is disconnected the OBEX session is lost. This section describes a solution, which allows OBEX to maintain the session when the underlying transport is lost. Before describing this solution it is necessary to review the current OBEX session protocol.

When the OBEX client creates an OBEX Transport Connection to the OBEX Server, an implicit OBEX session is created. A connection-oriented session can be created by sending an OBEX **CONNECT** command. An implicit OBEX session exists because it is possible to send directed commands to a service without creating a directed connection to it first. Connection-oriented sessions created using the OBEX **CONNECT** command are also called OBEX Connections. In the case of the Default OBEX Server, sending an OBEX **CONNECT** command without a **Target** header creates a connection to the Inbox service. This connection is called the Inbox connection and there are a number of services, which utilize this connection. A Directed Connection is made by sending an OBEX **CONNECT** command with a **Target** header containing the ID of the desired service or application.

The Default OBEX Server resides at a well-known location. The method for finding the Default OBEX Server is dependent on the underlying transport being used. OBEX applications can use Directed Connections on the Default OBEX Server or they can use there own OBEX server which runs over a different OBEX transport connection from the one used by the Default OBEX Server.

Currently, the implicit OBEX session created when an OBEX Client initiates an OBEX Transport connection is only reliable as long as the OBEX Transport connection exists. When the underlying

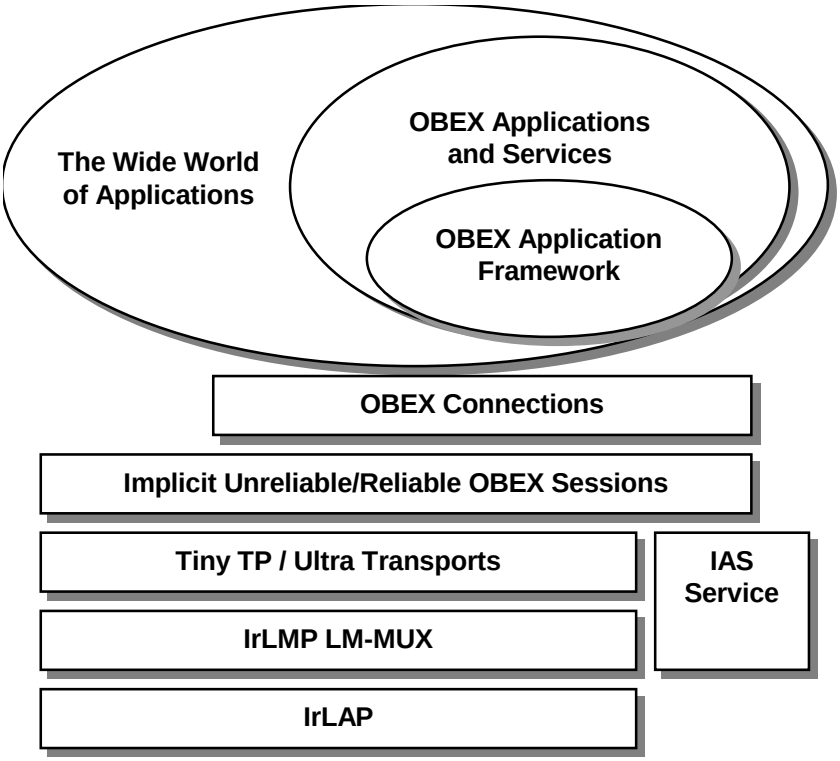
transport connection is disconnected the OBEX session fails. It is desired to have reliable OBEX sessions, which can be resumed when the underlying transport connection is broken. Given this context (broken OBEX Transport Connection), the implicit OBEX session is considered unreliable.

10.2 Reliable OBEX Session Overview

When the OBEX Client initiates the underlying transport connection to the OBEX Server it creates an unreliable implicit OBEX session. Any connection-oriented sessions made on top of this implicit session are also unreliable. This document proposes that reliable OBEX connections can be built by creating a reliable underlying session. The OBEX client can create a reliable session by sending a new OBEX command called **CREATESESSION**. In order to be able to reestablish the reliable session when the underlying transport breaks it is necessary for the session to have a Session ID that is known by both the OBEX Client and Server. This Session ID must be unique enough such that both the Client and the Server know they are resuming the same session. It is also necessary that the Client and Server know enough information about each other so that the underlying transport can be reestablished. In the case of IrDA this would be the 32-bit device address. Until a reliable session is established the unreliable implicit session is active. The list below outlines the concepts of a reliable OBEX Session.

- New OBEX commands are created for managing reliable sessions. These are **CREATESESSION**, **CLOSESESSION**, **SUSPENDSESSION, RESUMESESSION** and **SETTIMEOUT**.
- New OBEX headers are created to be used with the new commands and features. These are **Session-Parameters** and **Session-Sequence-Number**.
- When a reliable session is established a session ID is created that is known by both the Client and the Server. This ID must be unique enough such that the reliable session can be reestablished. When the underlying transport is broken under a reliable session, the session is considered suspended.
- The session context must be saved in order to resume a session when the underlying transport is broken.
- Applications/services must also maintain context in order to resume a session.
- Only one session can be active at a time per OBEX transport connection. When the OBEX transport connection is first established the unreliable implicit session is active. Multiple sessions can exist between devices but only one can be active at a time. The other sessions are considered suspended including the unreliable session. If the underlying transport is broken the unreliable session is broken as well.
- Any number of connection-oriented sessions (OBEX connections) can be made within a reliable session.
- The OBEX Capability Object can be accessed via any underlying session including the unreliable session.
- To facilitate reliability, the session layer must ensure reliable transfer of OBEX packets. To achieve this a header containing sequence numbers is added to each OBEX packet. If an OBEX packet is lost it will be retransmitted. This will occur only when a session is resumed since OBEX packets can only be lost when the underlying transport is lost (OBEX requires a reliable transport).
- Since OBEX packets are large, retransmission should only be used if the packet was actually lost. It is possible that the server received a command but the client did not receive the response. A method exists to allow the server to indicate the last packet received during session resume. The client can then send an empty packet (command without headers) to facilitate retransmission of the response.

The picture below shows how a reliable sessions fit into the architecture.


The Wide World of Applications
OBEX Applications and Services
OBEX Application Framework
OBEX Connections
Implicit Unreliable/Reliable OBEX Sessions
Tiny TP / Ultra Transports
IrLMP LM-MUX
IrLAP
IAS Service

10.3 Session Context

In order to resume a suspended session a context must be saved by both the client and the server. This context includes both session and OBEX information. The context contains the following items.

Item	Description
Session ID	This is the session ID created for this session. The session ID is generated from the information sent by both the client and server when the session was created so it does not actually have to be saved since it can be re-calculated.
Client Device Address	The device address sent by the client in the **CREATESESSION** command.
Client Nonce	The nonce sent by the client in the **CREATESESSION** command.
Server Device Address	The device address sent by the server in its response to the **CREATESESSION** command.
Server Nonce	The nonce sent by the server in its response to the **CREATESESSION** command.
Session Sequence Number	For the client this is the sequence number of the last packet sent. For the server this is the next sequence number expected.
Last OBEX packet	If the session was suspended unexpectedly because of loss of the underlying transport the last packet sent must be saved in case it needs to be transmitted. Both the client and the server must save the last packet sent. Also if the OBEX implementation buffers packets for the application then these must also be saved. It is possible to create an API between the OBEX layer and application such that the OBEX layer does not actually need to save the data. The API can allow the OBEX layer to ask the application to provide the data to be retransmitted.
Target Strings and Connection Ids	For all directed connections, which were created within the session, the target string and the corresponding Connection ID must be saved. Along with this is the need to save enough information to map the target back to the corresponding application. Also if the inbox connection has been created then this knowledge must also be saved.
OBEX Packet sizes	For each OBEX connection the maximum OBEX packet size must be saved.

Applications and services must also save context in order to resume a session. The OBEX layer does not know how commands such as PUT, GET, SETPATH, etc are used by the application/service so it cannot save context for them.

10.4 Persistence

If the OBEX layer and/or applications are shut down while a session is suspended then the context must be saved in a persistent data store if resuming the session later is desired.

10.5 Multiple Transport Support Proposal

This section describes a proposal for supporting sessions over different transports. This is not part of the errata but really just a method for capturing the idea. The actual method will be proposed as errata at a later date.

A possible method for supporting a session over different transports utilizes the OBEX Capability Object. The OBEX capability object must be accessible from any session via a connection to the Inbox service or the service that is accessed via the non-targeted OBEX connection. OBEX implementations that support sessions running over multiple transports will list the transport information in the Capability Object. This

includes the name of the transport and the device address used for that transport (e.g. IrDA is the 32-bit device address, Bluetooth is the 48-bit address, IP is the IP address, etc).

The proposal is to add a new section to the Capability Object listing the transports in which OBEX currently runs over. These are transports that allow an OBEX reliable session to be resumed independent of the transport the session was originally created or suspended. The section contains a list of transports where each transport contains the following attributes:

Protocols	The pertinent list of protocols on which OBEX resides. Items in the list are separated by commas. The first item in the list should be the main physical/link layer in the protocol stack (e.g. IrDA, Bluetooth, USB, Ethernet, Serial, etc). Typically this is the only protocol required. Other protocols are listed if needed.
Addresses	The list of addresses or port numbers needed to access the OBEX layer. Items in the list are separated by commas. There should be an item in the **Address List** corresponding to each item in the **Protocol List**. The last address in **the Address List** is the value used as the *Device Address* field in Session commands.

Both attributes are required. Below is an example Transports section with two entries.

```
<!—Transports Section -->
<Transports>
    <Transport Protocols =”IrDA” Addresses=”442356AE” />
    <Transport Protocols=”Bluetooth” Addresses=”00F349125E01” />
</Transports>
```

The method for moving a session to a different transport is to first suspend the session then resume the session over the new transport.

11. OBEX Testing Requirements

11.1 Symbols and Conventions

M	Mandatory support. Refers to capabilities that must be used in the OBEX test specification.
O	Optional support. Refers to capabilities that can be used in the OBEX test specification, but are not required.
X	Excluded. Refers to capabilities that may be supported by the device but shall not be exercised in this test specification.

11.2 OBEX Roles

OBEX Roles	Role
OBEX Client	O
OBEX Server	O

11.3 IrDA-Specific Features

IrDA-Specific Features	OBEX Client	OBEX Server
TinyTP connection	M	M
IAS Query procedure	M	M
IrLMP connection	M	M
IrLAP connection	M	M
Physical Layer	M	M
Physical Layer short-range option	O	O

11.4 OBEX Features

OBEX Features	OBEX Client	OBEX Server
Put operation	M	M
Get operation	O	O
SetPath operation	O	O
Connect operation	O	M
Disconnect operation	O	M
Session operation	O	O
Abort operation	O	O
Authentication	O	O
Directed Connections	O	M
Header support	M	M
User Defined operations	X	X

11.5 Required Behavior

Some behaviors are required of any device with an implementation of the OBEX protocol that wishes to be certified as an OBEX compliant device. The required behaviors are the subset of behaviors that are necessary to promote interoperability between various OBEX applications and devices.

Most of the required behaviors are intended to insure that any device that wishes to exchange an object in a protocol-compliant manner can do so. Therefore, some of the requirements affect only server behavior. Support for the **PUT** operation is required.

12. Appendices

12.1 Minimum level of service

Almost all elements of OBEX are optional to allow resource-constrained implementations to do the bare minimum while allowing reasonably rich interactions between more capable devices. Variations on this theme have been illustrated throughout this document. The only protocol requirements are that connection oriented versions of OBEX (such as those running over the IrDA TinyTP protocol) must use a **CONNECT** operation. Obviously at least one other operation will be needed to have a useful application.

A minimal OBEX server application would support **CONNECT** and either **PUT** or **GET**.

12.2 Extending OBEX

The headers, opcodes, flags, and constants defined for OBEX have ranges reserved for future use and ranges for user defined functions. Reserved ranges must not be used without official extensions to this specification. Please contact the authors or IrDA if you have proposed extensions with broad applicability. As a general rule, reserved flags or constants should be set to zero by senders and ignored by receivers.

User defined opcodes and headers are freely available for any kind of use. Obviously, they are only likely to make sense if both sides of the connection interpret them the same way, so the sides must somehow identify themselves to ensure compatibility. Recommended methods using the IrDA transport protocols are to use unique IAS class names, or optional headers (in particular the **Who** header) in the **CONNECT** packet.

12.3 Proposed Additions to OBEX

1. Add flexible acking, permitting the client to specify which packets it wants a response on. This permits faster operation by reducing the necessity to turn the IR link around frequently.
2. Add single byte encoding for **Type** header, covering common types
3. Add Offer operation - enables receiving side to preview and approve proposed objects
4. Add Negotiate operation - in-band negotiation
5. Add content negotiation - all sides to propose multiple options and have other side select
6. Add encryption
7. Add compression hooks

12.4 Known Target Identifiers

This section summarizes the UUID's known at the time when this document was last updated.

- **IrMC Service**
  - Unique identifier value "IRMC-SYNC".
  - Encoded as 9 bytes of ASCII text in a **Target** header.
- **Folder Browsing Service**
  - Unique Identifier value "F9EC7BC4-953C-11d2-984E-525400DC9E09".
  - Encoded as 16 bytes of hexadecimal in a **Target** header.
- SyncML Service
  - Unique identifier value "SYNCML-SYNC".
  - Encoded as 11 bytes of ASCII text in a **Target** header.

12.5 MD5 Algorithm for Authentication

This appendix contains code for a copyright and royalty free implementation of the MD5 algorithm. The code was copied from http://www.pbm.com/dice/rnd.txt.

```
/*
 * This code implements the MD5 message-digest algorithm.
 * The algorithm is due to Ron Rivest.  This code was
 * written by Colin Plumb in 1993, no copyright is claimed.
 * This code is in the public domain; do with it what you wish.
 *
 * Equivalent code is available from RSA Data Security, Inc.
 * This code has been tested against that, and is equivalent,
 * except that you don't need to include two pages of legalese
 * with every copy.
 *
 * To compute the message digest of a chunk of bytes, declare an
 * MD5Context structure, pass it to MD5Init, call MD5Update as
 * needed on buffers full of bytes, and then call MD5Final, which
 * will fill a supplied 16-byte array with the digest.
 */

typedef unsigned long word32;
typedef unsigned char byte;

struct xMD5Context {
        word32 buf[4];
        word32 bytes[2];
        word32 in[16];
};

void xMD5Init(struct xMD5Context *context);
void xMD5Update(struct xMD5Context *context, byte const *buf, int len);
void xMD5Final(byte digest[16], struct xMD5Context *context);
void xMD5Transform(word32 buf[4], word32 const in[16]);


/*
 * Shuffle the bytes into little-endian order within words, as per the
 * MD5 spec.  Note: this code works regardless of the byte order.
 */
void
byteSwap(word32 *buf, unsigned words)
{
        byte *p = (byte *)buf;

        do {
                *buf++ = (word32)((unsigned)p[3] << 8 | p[2]) << 16 |
                        ((unsigned)p[1] << 8 | p[0]);
                p += 4;
        } while (--words);
}

/*
 * Start MD5 accumulation.  Set bit count to 0 and buffer to mysterious
 * initialization constants.
 */
void
xMD5Init(struct xMD5Context *ctx)
{
        ctx->buf[0] = 0x67452301;
        ctx->buf[1] = 0xefcdab89;
        ctx->buf[2] = 0x98badcfe;
        ctx->buf[3] = 0x10325476;
```

```
        ctx->bytes[0] = 0;
        ctx->bytes[1] = 0;
}

/*
 * Update context to reflect the concatenation of another buffer full
 * of bytes.
 */
void
xMD5Update(struct xMD5Context *ctx, byte const *buf, int len)
{
        word32 t;

        /* Update byte count */

        t = ctx->bytes[0];
        if ((ctx->bytes[0] = t + len) < t)
                ctx->bytes[1]++;        /* Carry from low to high */

        t = 64 - (t & 0x3f); /* Space avail in ctx->in (at least 1) */
        if ((unsigned)t > len) {
                bcopy(buf, (byte *)ctx->in + 64 - (unsigned)t, len);
                return;
        }
        /* First chunk is an odd size */
        bcopy(buf,(byte *)ctx->in + 64 - (unsigned)t, (unsigned)t);
        byteSwap(ctx->in, 16);
        xMD5Transform(ctx->buf, ctx->in);
        buf += (unsigned)t;
        len -= (unsigned)t;

        /* Process data in 64-byte chunks */
        while (len >= 64) {
                bcopy(buf, ctx->in, 64);
                byteSwap(ctx->in, 16);
                xMD5Transform(ctx->buf, ctx->in);
                buf += 64;
                len -= 64;
        }

        /* Handle any remaining bytes of data. */
        bcopy(buf, ctx->in, len);
}

/*
 * Final wrapup - pad to 64-byte boundary with the bit pattern
 * 1 0* (64-bit count of bits processed, MSB-first)
 */
void
xMD5Final(byte digest[16], struct xMD5Context *ctx)
{
        int count = (int)(ctx->bytes[0] & 0x3f); /* Bytes in ctx->in */
        byte *p = (byte *)ctx->in + count;      /* First unused byte */

        /* Set the first char of padding to 0x80. There is always room.*/
        *p++ = 0x80;

        /* Bytes of padding needed to make 56 bytes (-8..55) */
        count = 56 - 1 - count;
```

```
        if (count < 0) {        /* Padding forces an extra block */
                bzero(p, count+8);
                byteSwap(ctx->in, 16);
                xMD5Transform(ctx->buf, ctx->in);
                p = (byte *)ctx->in;
                count = 56;
        }
        bzero(p, count+8);
        byteSwap(ctx->in, 14);

        /* Append length in bits and transform */
        ctx->in[14] = ctx->bytes[0] << 3;
        ctx->in[15] = ctx->bytes[1] << 3 | ctx->bytes[0] >> 29;
        xMD5Transform(ctx->buf, ctx->in);

        byteSwap(ctx->buf, 4);
        bcopy(ctx->buf, digest, 16);
        bzero(ctx,sizeof(ctx));
}


/* The four core functions - F1 is optimized somewhat */

/* #define F1(x, y, z) (x & y | ~x & z) */
#define F1(x, y, z) (z ^ (x & (y ^ z)))
#define F2(x, y, z) F1(z, x, y)
#define F3(x, y, z) (x ^ y ^ z)
#define F4(x, y, z) (y ^ (x | ~z))

/* This is the central step in the MD5 algorithm. */
#define MD5STEP(f,w,x,y,z,in,s) \
         (w += f(x,y,z) + in, w = (w<<s | w>>(32-s)) + x)

/*
 * The core of the MD5 algorithm, this alters an existing MD5 hash to
 * reflect the addition of 16 longwords of new data.  MD5Update blocks
 * the data and converts bytes into longwords for this routine.
 */
void
xMD5Transform(word32 buf[4], word32 const in[16])
{
        register word32 a, b, c, d;

        a = buf[0];
        b = buf[1];
        c = buf[2];
        d = buf[3];

        MD5STEP(F1, a, b, c, d, in[0] + 0xd76aa478, 7);
        MD5STEP(F1, d, a, b, c, in[1] + 0xe8c7b756, 12);
        MD5STEP(F1, c, d, a, b, in[2] + 0x242070db, 17);
        MD5STEP(F1, b, c, d, a, in[3] + 0xc1bdceee, 22);
        MD5STEP(F1, a, b, c, d, in[4] + 0xf57c0faf, 7);
        MD5STEP(F1, d, a, b, c, in[5] + 0x4787c62a, 12);
        MD5STEP(F1, c, d, a, b, in[6] + 0xa8304613, 17);
        MD5STEP(F1, b, c, d, a, in[7] + 0xfd469501, 22);
        MD5STEP(F1, a, b, c, d, in[8] + 0x698098d8, 7);
        MD5STEP(F1, d, a, b, c, in[9] + 0x8b44f7af, 12);
        MD5STEP(F1, c, d, a, b, in[10] + 0xffff5bb1, 17);
```

```
MD5STEP(F1, b, c, d, a, in[11] + 0x895cd7be, 22);
MD5STEP(F1, a, b, c, d, in[12] + 0x6b901122, 7);
MD5STEP(F1, d, a, b, c, in[13] + 0xfd987193, 12);
MD5STEP(F1, c, d, a, b, in[14] + 0xa679438e, 17);
MD5STEP(F1, b, c, d, a, in[15] + 0x49b40821, 22);

MD5STEP(F2, a, b, c, d, in[1] + 0xf61e2562, 5);
MD5STEP(F2, d, a, b, c, in[6] + 0xc040b340, 9);
MD5STEP(F2, c, d, a, b, in[11] + 0x265e5a51, 14);
MD5STEP(F2, b, c, d, a, in[0] + 0xe9b6c7aa, 20);
MD5STEP(F2, a, b, c, d, in[5] + 0xd62f105d, 5);
MD5STEP(F2, d, a, b, c, in[10] + 0x02441453, 9);
MD5STEP(F2, c, d, a, b, in[15] + 0xd8a1e681, 14);
MD5STEP(F2, b, c, d, a, in[4] + 0xe7d3fbc8, 20);
MD5STEP(F2, a, b, c, d, in[9] + 0x21e1cde6, 5);
MD5STEP(F2, d, a, b, c, in[14] + 0xc33707d6, 9);
MD5STEP(F2, c, d, a, b, in[3] + 0xf4d50d87, 14);
MD5STEP(F2, b, c, d, a, in[8] + 0x455a14ed, 20);
MD5STEP(F2, a, b, c, d, in[13] + 0xa9e3e905, 5);
MD5STEP(F2, d, a, b, c, in[2] + 0xfcefa3f8, 9);
MD5STEP(F2, c, d, a, b, in[7] + 0x676f02d9, 14);
MD5STEP(F2, b, c, d, a, in[12] + 0x8d2a4c8a, 20);

MD5STEP(F3, a, b, c, d, in[5] + 0xfffa3942, 4);
MD5STEP(F3, d, a, b, c, in[8] + 0x8771f681, 11);
MD5STEP(F3, c, d, a, b, in[11] + 0x6d9d6122, 16);
MD5STEP(F3, b, c, d, a, in[14] + 0xfde5380c, 23);
MD5STEP(F3, a, b, c, d, in[1] + 0xa4beea44, 4);
MD5STEP(F3, d, a, b, c, in[4] + 0x4bdecfa9, 11);
MD5STEP(F3, c, d, a, b, in[7] + 0xf6bb4b60, 16);
MD5STEP(F3, b, c, d, a, in[10] + 0xbebfbc70, 23);
MD5STEP(F3, a, b, c, d, in[13] + 0x289b7ec6, 4);
MD5STEP(F3, d, a, b, c, in[0] + 0xeaa127fa, 11);
MD5STEP(F3, c, d, a, b, in[3] + 0xd4ef3085, 16);
MD5STEP(F3, b, c, d, a, in[6] + 0x04881d05, 23);
MD5STEP(F3, a, b, c, d, in[9] + 0xd9d4d039, 4);
MD5STEP(F3, d, a, b, c, in[12] + 0xe6db99e5, 11);
MD5STEP(F3, c, d, a, b, in[15] + 0x1fa27cf8, 16);
MD5STEP(F3, b, c, d, a, in[2] + 0xc4ac5665, 23);

MD5STEP(F4, a, b, c, d, in[0] + 0xf4292244, 6);
MD5STEP(F4, d, a, b, c, in[7] + 0x432aff97, 10);
MD5STEP(F4, c, d, a, b, in[14] + 0xab9423a7, 15);
MD5STEP(F4, b, c, d, a, in[5] + 0xfc93a039, 21);
MD5STEP(F4, a, b, c, d, in[12] + 0x655b59c3, 6);
MD5STEP(F4, d, a, b, c, in[3] + 0x8f0ccc92, 10);
MD5STEP(F4, c, d, a, b, in[10] + 0xffeff47d, 15);
MD5STEP(F4, b, c, d, a, in[1] + 0x85845dd1, 21);
MD5STEP(F4, a, b, c, d, in[8] + 0x6fa87e4f, 6);
MD5STEP(F4, d, a, b, c, in[15] + 0xfe2ce6e0, 10);
MD5STEP(F4, c, d, a, b, in[6] + 0xa3014314, 15);
MD5STEP(F4, b, c, d, a, in[13] + 0x4e0811a1, 21);
MD5STEP(F4, a, b, c, d, in[4] + 0xf7537e82, 6);
MD5STEP(F4, d, a, b, c, in[11] + 0xbd3af235, 10);
MD5STEP(F4, c, d, a, b, in[2] + 0x2ad7d2bb, 15);
MD5STEP(F4, b, c, d, a, in[9] + 0xeb86d391, 21);

buf[0] += a;
buf[1] += b;
buf[2] += c;
```

```
        buf[3] += d;
}


void MD5(void *dest, void *orig, int len)
{
        struct xMD5Context context;

        xMD5Init(&context);
        xMD5Update(&context, orig, len);
        xMD5Final(dest, &context);
}
```

https://web.archive.org/web/20060901023650/http://nds2.nokia.com/files/support/nam/phones/guides/N91_US_en.PDF

NOKIA
N91

Nokia N91-1

Nokia N91 User Guide

Nota: La traducción en español de este manual inicia en la Página 125.

DECLARATION OF CONFORMITY
We, NOKIA CORPORATION declare under our sole responsibility that the product RM-43 is in conformity with the provisions of the following Council Directive: 1999/5/EC. A copy of the Declaration of Conformity can be found at **http://www.nokia.com/phones/declaration_of_conformity/**.

CE 434 !

The crossed-out wheeled bin means that within the European Union the product must be taken to separate collection at the product end-of-life. This applies to your device but also to any enhancements marked with this symbol. Do not dispose of these products as unsorted municipal waste.

Copyright © 2006 Nokia. All rights reserved.

Reproduction, transfer, distribution or storage of part or all of the contents in this document in any form without the prior written permission of Nokia is prohibited.

Nokia, Nokia Connecting People, and Pop-Port are trademarks or registered trademarks of Nokia Corporation. Other product and company names mentioned herein may be trademarks or tradenames of their respective owners.

Nokia tune is a sound mark of Nokia Corporation.

symbian

This product includes software licensed from Symbian Software Ltd © 1998-200(6). Symbian and Symbian OS are trademarks of Symbian Ltd.

Java™ and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc.

Java POWERED

Microsoft, Windows, and Media Player are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Stac ®, LZS ®, ©1996, Stac, Inc., ©1994-1996 Microsoft Corporation. Includes one or more U.S. Patents: No. 4701745, 5016009, 5126739, 5146221, and 5414425. Other patents pending.

Hi/fn ®, LZS ®,©1988-98, Hi/fn. Includes one or more U.S. Patents: No. 4701745, 5016009, 5126739, 5146221, and 5414425. Other patents pending.

Part of the software in this product is © Copyright ANT Ltd. 1998. All rights reserved. US Patent No 5818437 and other pending patents. T9 text input software Copyright © 1997-2005. Tegic Communications, Inc. All rights reserved.

This product is licensed under the MPEG-4 Visual Patent Portfolio License (i) for personal and noncommercial use in connection with information which has been encoded in compliance with the MPEG-4 Visual Standard by a consumer engaged in a personal and noncommercial activity and (ii) for use in connection with MPEG-4 video provided by a licensed video provider. No license is granted or shall be implied for any other use. Additional information, including that related to promotional, internal, and commercial uses, may be obtained from MPEG LA, LLC. See **http://www.mpegla.com**.

Nokia operates a policy of continuous development. Nokia reserves the right to make changes and improvements to any of the products described in this document without prior notice.

Under no circumstances shall Nokia be responsible for any loss of data or income or any special, incidental, consequential or indirect damages howsoever caused.

The contents of this document are provided 'as is'. Except as required by applicable law, no warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, are made in relation to the accuracy, reliability or contents of this document. Nokia reserves the right to revise this document or withdraw it at any time without prior notice.

The availability of particular products may vary by region. Please check with the Nokia dealer nearest to you.

Export Controls

This device may contain commodities, technology or software subject to export laws and regulations from the US and other countries. Diversion contrary to law is prohibited.

FCC / Industry Canada Notice

Your device may cause TV or radio interference (for example, when using a telephone in close proximity to receiving equipment). The FCC or Industry Canada can require you to stop using your telephone if such interference cannot be eliminated. If you require assistance, contact your local service facility. This device complies with part 15 of the FCC rules. Operation is subject to the condition that this device does not cause harmful interference.

Issue 1 EN, 9248320

Contents

For your safety 9

Your Nokia N91 12

Hard drive 12
View memory consumption 12
Memory low—free memory 12
Wireless LAN 13
Essential indicators 14
Nokia support on the Web 15
Help 15
Tutorial 16
Clock 16
- Clock settings 16
- World clock 17

Volume and loudspeaker control 17
File manager 17
Keypad lock (keyguard) 18

Music player 19

Play a song 19
Controls 19
Indicators 20
Play modes 20
- Shuffle play 20
- Repeat play 20

Modify sound 20
- Equalizer 20
- Audio settings 20

Music shop 21
- Go to music shop 21
- Music shop settings 21
- Find in music shop 21

Transfer music 22
- Transfer from PC 22
- Music transfer 23

Playlists 24
- Create a playlist 24
- Add songs to a playlist 24
- Reorder songs 24
- Remove songs from a playlist 24
- Add items from views to a playlist 25
- Playlist details 25
- Automatic playlists 25

Available options 25
- Options only in Now Playing view 26

Rename items 26
View and edit song details 27
Library details 27
Set a song as a ringing tone 27
Change album art 27

About encoded audio 28
Bitrates 28
Variable bitrate 28
Personalize your device 29
Profiles—Set tones 29
Offline profile 29
Change the look of your device 30
Active standby mode 31
Make calls 32
Voice calls 32
Make a conference call 32
1-touch dial a phone number 33
Voice dialing 33
Answer or decline a call 34
Call waiting (network service) 34
Options during a voice call 35
Contacts (Phonebook) 36
Save names and numbers 36
Default numbers and addresses 37
SIM directory and SIM services 37
Copy contacts 37
Add ringing tones for contacts 38
Create contact groups 38
Remove members from a group 38
Media applications 39
Camera 39
Take pictures 39
Viewfinder 41
Tips on taking good photographs 41
Record videos 42
Gallery 43
View images and video clips 44
Folders 44
Music player 44
RealPlayer™ 45
Play video 45
Stream content over the air 45
Receive RealPlayer settings 45
FlashPlayer 46
Organize flash files 46
Play flash files 47
Recorder 47
Radio 48
Listen to the radio 48
View visual content 49
Saved stations 49
Settings 49

Messaging 50
Write text 51
Traditional text input 51
Predictive text input—Predictive text 52
Copy text to clipboard 53
Write and send messages 54
Receive MMS and e-mail settings 56
Inbox—receive messages 56
Multimedia messages 57
Data and settings 57
Web service messages 57
My folders 58
Mailbox 58
Open the mailbox 58
Retrieve e-mail messages 58
Retrieve e-mail messages automatically 59
Delete e-mail messages 59
Disconnect from the mailbox 60
Outbox—messages waiting to be sent 60
View messages on a SIM card 61
Messaging settings 61
Text messages 61
Multimedia messages 62
E-mail 63
Web service messages 65
Cell broadcast 65
Other settings 65

My own 66
Web 66
Notes 66
Go to—add shortcuts 66
Delete a shortcut 67
IM—instant messaging (chat) 67
Receive IM settings 67
Connect to an IM server 67
Modify your IM settings 68
Search for IM groups and users 68
Join IM groups 68
Chat 68
View and start conversations 69
IM contacts 69
Manage IM groups 70
IM server settings 71
Push to talk (network service) 71
Define a push-to-talk access point 72
Push-to-talk settings 72
Log in to push to talk 73
Exit push to talk 73
Make a one-to-one call 73
Answer a one-to-one call 74
Make a group call 74
Subscribe login status of others 74
Send a callback request 74
Channels 74
Respond to a channel invitation 75
View the push-to-talk log 75

Log 76
Recent calls 76
Call timers 76
Packet data 76
Monitor all communication events 76

Calendar 78
Create calendar entries 78
Set a calendar alarm 79
Calendar views 79
To-do view 80
Remove calendar entries 80
Calendar settings 80

Services 81
Access point for Services 81
Bookmarks view 81
Add bookmarks manually 82
Send bookmarks 82
Make a connection 82
Connection security 82
Browse 83
View saved pages 84
Download and purchase items 84
End a connection 84
Empty the cache 84
Settings for Services 85

Connectivity 86
Bluetooth connectivity 86
Settings 86
Send data using Bluetooth connectivity 87
Pair devices 88
Receive data using Bluetooth connectivity 89
Switch off Bluetooth connectivity 89
Data cable 89
PC connections 89
CD-ROM 90
Your device as a modem 90
Connection manager 90
Data connections 90
View data connection details 91
Wireless Local Area Network (WLAN) 91
Wireless LAN 91
Synchronization 91
Create a new synchronization profile 92
Synchronize data 93
Device manager 93
Server profile settings 93

Tools 94
Calculator 94
Calculate percentages 94
Converter 94
Set base currency and exchange rates 94

Settings 95
Device settings 95
Call settings 96
Connection 97
Date & time 100
Security 101
Call forwarding 103
Call restriction 104
Network 104
Enhancement settings 105
Voice commands 105
Application manager 105
Install applications and software 106
Remove applications and software 107
Application settings 107
Memory-Hard drive 108
Tools 108
Digital Rights Management 109
Troubleshooting 110
Q&A 110
Product safety 113
Battery information 113
Charging and discharging 113
Operation times 114
Nokia battery authentication guidelines 114
Care and maintenance 115
Additional safety information 116
Technical Information 119
Index 120

Notes

FOR YOUR SAFETY

Read these simple guidelines. Not following them may be dangerous or illegal. Read the complete user guide for further information.

SWITCH ON SAFELY
Do not switch the device on when wireless phone use is prohibited or when it may cause interference or danger.

ROAD SAFETY COMES FIRST
Obey all local laws. Always keep your hands free to operate the vehicle while driving. Your first consideration while driving should be road safety.

INTERFERENCE
All wireless devices may be susceptible to interference, which could affect performance.

SWITCH OFF IN HOSPITALS
Follow any restrictions. Switch the device off near medical equipment.

SWITCH OFF IN AIRCRAFT
Follow any restrictions. Wireless devices can cause interference in aircraft.

SWITCH OFF WHEN REFUELING
Do not use the device at a refueling point. Do not use near fuel or chemicals.

SWITCH OFF NEAR BLASTING
Follow any restrictions. Do not use the device where blasting is in progress.

USE SENSIBLY
Use only in the normal position as explained in the product documentation. Do not touch the antenna unnecessarily.

QUALIFIED SERVICE
Only qualified personnel may install or repair this product.

ENHANCEMENTS AND BATTERIES
Use only approved enhancements and batteries. Do not connect incompatible products.

WATER-RESISTANCE
Your device is not water-resistant. Keep it dry.

BACK-UP COPIES
Remember to make back-up copies or keep a written record of all important information.

CONNECTING TO OTHER DEVICES
When connecting to any other device, read its user guide for detailed safety instructions. Do not connect incompatible products.

EMERGENCY CALLS
Ensure the phone function of the device is switched on and in service. Press the end key as many times as needed to clear the display and return to the standby mode. Enter the emergency number, then press the call key. Give your location. Do not end the call until given permission to do so.

About your device

The wireless device described in this guide is approved for use on the EGSM 900/1800/1900 and UMTS 2100 networks. Contact your service provider for more information about networks.

When using the features in this device, obey all laws and respect privacy and legitimate rights of others.

Copyright protections may prevent some images, music (including ringing tones), and other content from being copied, modified, transferred or forwarded.

When taking and using images or video clips, obey all laws and respect local customs as well as privacy and legitimate rights of others.

Warning: To use any features in this device, other than the alarm clock, the device must be switched on. Do not switch the device on when wireless device use may cause interference or danger.

Network Services

To use the phone you must have service from a wireless service provider. Many of the features in this device depend on features in the wireless network to function. These network services may not be available on all networks or you may have to make specific arrangements with your service provider before you can utilize network services. Your service provider may need to give you additional instructions for their use and explain what charges will apply. Some networks may have limitations that affect how you can use network services. For instance, some networks may not support all language-dependent characters and services.

Your service provider may have requested that certain features be disabled or not activated in your device. If so, they will not appear on your device menu. Your device may also have been specially configured. This configuration may include changes in menu names, menu order and icons. Contact your service provider for more information.

This device supports WAP 2.0 protocols (HTTP and SSL) that run on TCP/IP protocols. Some features of this device, such as MMS, browsing, e-mail and content downloading via browser or over MMS, require network support for these technologies.

For availability and information on using SIM card services, contact your SIM card vendor. This may be the service provider, network operator, or other vendor.

Enhancements, batteries, and chargers

Always switch the device off and disconnect the charger before removing the battery.

Removing the battery before switching off the device may corrupt the hard drive.

Check the model number of any charger before use with this device. This device is intended for use when supplied with power from a DC-4, AC-3 and AC-4 charger and from an AC-1, ACP-7, ACP-8, ACP-9, ACP-12U, LCH-8, LCH-9, or LCH-12 charger when used with the charger adapter CA-44.

The battery intended for use with this device is BL-5C.

Warning: Use only batteries, chargers, and enhancements approved by Nokia for use with this particular model. The use of any other types may invalidate any approval or warranty, and may be dangerous.

For availability of approved enhancements, please check with your dealer. When you disconnect the power cord of any enhancement, grasp and pull the plug, not the cord.

Your Nokia N91

Model number: Nokia N91-1

Hereinafter referred to as Nokia N91.

Congratulations on your purchase of the Nokia N91. Your device combines stereo sound, storage capacity for up to 3000 songs, over-the-air music downloads, access to your music collection and smartphone functionality with Wi-Fi and 2Mpix camera.

Hard drive

Your device is equipped with a 4 Gb hard drive providing high capacity storage room for different types of files. Sensors in the device will stop the hard drive if the operating temperature is too low or too high, if the device is dropped, or if the back cover of the device is removed. Hard drives are fragile components that must be handled with care. All impacts caused by shock, vibration, bumping, and dropping or other mishandling of the device, as well as extreme temperatures, humidity and liquids, may cause the hard drive to malfunction and data to be lost or corrupted.

Do not remove the back cover when the device is switched on.

View memory consumption

Leaving applications running in the background increases the demand on battery power and reduces the battery life. Press and hold to activate open applications and press **Options** > **Exit** to quit an application.

To view what types of data you have in the device and how much memory the different data types consume, press , and select **Tools** > **Hard drive**. Scroll down to **Free memory** to view the amount of free memory in the device.

Memory low—free memory

The device notifies you if the memory is getting low. If the device memory starts to get low when your browser is open, then the browser is closed automatically to free some memory space.

To free device memory, transfer data to the hard drive in **File mgr.**. Mark files to be moved, select **Options** > **Move to folder** > **Hard drive**, and select a folder.

Tip! To free memory from the device, use Image Store available in Nokia PC Suite to transfer sound files, images, and video clips to a compatible PC. See the CD-ROM supplied in the sales package.

To remove data to free memory, use **File mgr.**, or go to the respective application. For example, you can remove the following:

- Messages from **Inbox**, **Drafts**, and **Sent** folders in **Messag.**
- Retrieved e-mail messages from the phone memory
- Saved Web pages
- Saved images, videos, or sound files
- Contact information
- Calendar notes
- Downloaded applications. See also "Application manager," p. 105.
- Any other data that you no longer need

Wireless LAN

Your device is approved for use in a wireless local area network (wireless LAN). With wireless LAN, you can connect your device to the Internet.

Important: Always enable one of the available encryption methods to increase the security of your wireless LAN connection. Using encryption reduces the risk of unauthorized access to your data.

Some places, like France, have restrictions on the use of wireless LAN. Check with your local authorities for more information.

Features using wireless LAN, or allowing such features to run in the background while using other features, increase the demand on battery power and reduce the battery life.

Your device supports the following wireless LAN features:

- IEEE 802.11g standard
- Operation at a frequency of 2.4 GHz
- Encryption methods wired equivalent privacy (WEP) with keys up to 128 bits, Wi-Fi protected access (WPA), and 802.1x. These functions can be used only if they are supported by the network.

You can create an Internet access point (IAP) in a wireless LAN and use it for applications that need to connect to the Internet.

Create an Internet access point

1. Press , and select **Connect.** > **Conn mgr** > **Availab. WLAN**.
2. The device searches for wireless LANs within range. To create an Internet access point in a network, select **Options** > **Define access point**.
3. The device creates an Internet access point with default settings. To view or change the settings of the Internet access point, see "Access points," p. 98.

When an application asks you to select an access point, select the created access point, or **Search for WLAN**, to search for wireless LANs in range.

A wireless LAN connection is established when you create a data connection using a wireless LAN Internet access point. The active wireless LAN connection ends when you end the data connection. To end the connection, see "Connection manager," p. 90.

You can use wireless LAN during a voice call or when packet data is active. You can only be connected to one wireless LAN access point device at a time, but several applications can use the same wireless LAN connection.

When you activate the **Offline** profile, you can still use wireless LAN (if available). Remember to comply with any applicable safety requirements when establishing and using a wireless LAN connection.

There are two operating modes in wireless LAN: **Infrastructure** and **Ad-hoc**.

The infrastructure operating mode allows two kinds of communication: wireless devices are connected to each other through a wireless LAN access point device, or wireless devices are connected to a wired LAN through a wireless LAN access point device.

In the ad hoc operating mode, devices can send and receive data directly with each other. To create an Internet access point for the ad hoc network, see "Access points," p. 98.

Tip! To check the unique Media Access Control (MAC) address that identifies your device, enter *#62209526# in the standby mode.

Essential indicators

The phone is being used in a GSM network.

3G (network service)
The phone is being used in a UMTS network. See "Network," p. 104.

You have one or more unread messages in the **Inbox** folder in **Messag.**.

You have received new e-mail in the remote mailbox.

There are messages waiting to be sent in the **Outbox** folder. See "Outbox—messages waiting to be sent," p. 60.

You have missed calls. See "Voice calls," p. 32.

Shown if **Ringing type** is set to **Silent** and **Message alert tone**, **IM alert tone**, and **E-mail alert tone** are set to **Off**. See "Profiles—Set tones," p. 29.

The phone keypad is locked. See "Keypad lock (keyguard)," p. 18.

You have an active clock alarm. See "Clock," p. 16.

2 The second phone line is being used. See "Call settings," p. 96.

All calls to the phone are forwarded to another number. If you have two phone lines, the forward indicator for the first line is 1 and for the second 2.

A headset is connected to the phone.

The connection to a headset with Bluetooth connectivity has been lost.

A data call is active.

A GPRS or EDGE packet data connection is available.

A GPRS or EDGE packet data connection is active.

A GPRS or EDGE packet data connection is on hold. These icons are shown instead of the signal strength indicator (shown in the top left corner in the standby mode). See "Packet data connections in GSM and UMTS networks," p. 97.

A UMTS packet data connection is available.

A UMTS packet data connection is active.

A UMTS packet data connection is on hold.

Bluetooth connectivity is **On**.

Data is being transmitted using a Bluetooth connection. See "Bluetooth connectivity," p. 86.

You have set the device to scan for wireless LANs, and a wireless LAN is available. See "Wireless Local Area Network (WLAN)," p. 91.

A wireless LAN connection is active in a network that has encryption.

A wireless LAN connection is active in a network that does not have encryption.

A USB connection is active.

You have an e-mail that has been read, waiting for you to retrieve to your phone.

You have an e-mail that has not been read, waiting for you to retrieve to your phone.

Nokia support on the Web

Check http://www.nokia.com/support or your local Nokia Web site for the latest version of this guide, additional information, downloads, and services related to your Nokia product.

Help

Your device has context-sensitive help. You can access the help from an application or from the main menu.

When an application is open, to access help for the current view, select **Options** > **Help**.

When you are reading the instructions, to switch between help and the application that is open in the background, press and hold .

To open help from the main menu, select **Tools** > **Help**. Select the desired application to view the help topics. To search for help topics using keywords, select **Options** > **Search by keyword**.

When you are reading a help topic, to view other related topics, scroll left or right with the joystick.

Tutorial

The tutorial gives you information about some of the features of your device. To access the tutorial in the menu, press , and select **Tutorial** and the section you want to view.

Clock

Press , and select **Clock**.

To set a new alarm, select **Options** > **Set alarm**. Enter the alarm time, and select **OK**. When the alarm is active, the sign is shown.

To turn off the alarm, select **Stop**. When the alarm tone sounds, press any key, or select **Snooze** to stop the alarm for five minutes, after which it resumes. You can do this a maximum of five times.

If the alarm time is reached while the device is switched off, the device switches itself on and starts sounding the alarm tone. If you select **Stop**, the device asks whether you want to activate the device for calls. Select **No** to switch off the device or **Yes** to make and receive calls. Do not select **Yes** when wireless phone use may cause interference or danger.

To cancel an alarm, select **Clock** > **Options** > **Remove alarm**.

Clock settings

To change the clock settings, select **Clock** > **Options** > **Settings**.

To change the clock shown in the standby mode, scroll down, and select **Clock type** > **Analog** or **Digital**.

To allow the mobile phone network to update the time, date, and time zone information to your phone (network service), scroll down, and select **Network operator time**. For the **Network operator time** setting to take effect, the phone restarts.

To change the alarm tone, scroll down, and select **Clock alarm tone**.

World clock

Open **Clock**, and scroll right to open the world clock view. In the world clock view, you can view the time in different cities.

To add cities to the list, select **Options** > **Add city**. Enter the first letters of the city name. The search field appears automatically, and the matching cities are displayed. Select a city. You can add a maximum of 15 cities to the list.

To set your current city, scroll to a city, and select **Options** > **My current city**. The city is displayed in the clock main view, and the time in your phone is changed according to the city selected. Check that the time is correct and matches your time zone.

Volume and loudspeaker control

To increase or decrease the volume level, when you have an active call or are listening to a sound, press and hold or on the left side of the device.

The built-in loudspeaker allows you to speak and listen to the phone from a short distance without having to hold the phone to your ear, for example, having it on a table nearby. Sound applications use the loudspeaker by default.

Warning: Do not hold the device near your ear when the loudspeaker is in use, because the volume may be extremely loud.

To locate the loudspeaker, see 'Keys and parts' in the Quick start guide.

To use the loudspeaker during a call, start a call, and select **Options** > **Activate loudsp.**.

To turn off the loudspeaker when you have an active call or are listening to a sound, select **Options** > **Activate handset**.

File manager

Many features of the phone use memory to store data. These features include contacts, messages, images, ringing tones, calendar and to-do notes, documents, and downloaded applications. The free memory available depends on how much data is already saved in the phone memory.

To browse files and folders in the device memory or hard drive, press , and select **Tools** > **File mgr.**. The device memory view opens. Move the joystick right to view the folders.

To move or copy files to a folder, press and press the joystick at the same time to mark a file, and select **Options** > **Move to folder** or **Copy to folder**.

Many different file formats can be opened directly from File manager. However, to get the full functionality you should open the file in the corresponding application, not in File manager.

Icons in **File mgr.**:

Folder

Folder that has a subfolder

To find a file, select **Options** > **Find**, select the search location and enter the text that matches the file name.

Tip! You can use Nokia Phone Browser available in Nokia PC Suite to view the content of the device memory and hard drive of your phone. See the CD-ROM in the sales package.

Keypad lock (keyguard)

To lock: Move the lock switch on top of the device next to the power key, to the left. When you move the switch when the slide cover is closed, all the keys are locked. When you open the slide cover, the number keypad is active.

To unlock: Move the switch to the right.

When the keypad lock is on, calls still may be possible to the official emergency number programmed into your device. Enter the emergency number, and press .

Music player

Music player supports files with extensions AAC, AAC+, eAAC+, MP3, WMA, WAV, 3G2, 3GP, 3GPP, 3GPP2, MPEG, MP4, M4A, DCF, ODF, ASF, and M3U. **Music player** does not necessarily support all features of a file format or all the variations of file formats.

For more information on player, select **Options** > **Help** for context sensitive help.

Copyright protections may prevent some images, music (including ringing tones), and other content from being copied, modified, transferred or forwarded.

Play a song

1 Press , and select **Music player**.

2 Move the joystick to navigate to a preferred song, artist or other category and press the joystick to open category lists.

3 Press ▶II Play/Pause key to play the selected song or list of songs.

To open the **Now playing** view of the player from any other view, press the Music access key. To return to previous view, press it again.

To select more music to play in **Now playing** view, select **Options** > **Go to Music menu**.

Controls

Viewfinder indicators show the following:

Press to go to the next song. To fast forward, press and hold.

Press once to return to the beginning of the song. To skip to the previous song, press within two seconds after a song has started. To rewind through the song, press and hold.

Press to decrease volume. Located on the left side of the device.

Press to increase volume. Located on the left side of the device.

To lock all the keys of the device except the volume and power key, slide the lock switch on top of the device.

Tip! You can also use the keys on the remote to control the device. The lock switch on the side of the remote control only locks the keys of the remote control.

Indicators

Icons in **Music player**:

—Repeat one

—Repeat all

—Shuffle on

—Loudspeaker muted (displayed in volume pop-up)

Play modes

Shuffle play

1 To switch random play on or off, select **Options** > **Shuffle**.

2 Select **Off** or **On**. The **Now playing** view is displayed with or without the shuffle icon in top right corner.

Repeat play

1 To repeat the current song, all songs, or to switch repeat off, select **Options** > **Repeat**.

2 Select **Off**, **All**, or **One**. The **Now playing** view is displayed with or without the repeat all or repeat song icon in the top right corner.

Modify sound

Equalizer

To modify the tone of the music playback, select **Options** > **Equalizer**.

To use one of the preset equalizer settings, select it with the joystick, and select **Activate**. Each preset has different settings on a 8-band graphic equalizer.

Modify equalizer presets

1 To create a new preset, select **Options**> **New preset**. To modify an existing preset select **Options** > **Edit**.

2 To adjust the frequency settings, move the joystick to the left or right to select low, mid, or, high frequency and up or down to increase or decrease the volume of the selected frequency band.

3 To reset the settings to their original values, select **Options** > **Reset to defaults**.

To delete a preset, scroll to it, and press . The default presets cannot be deleted.

Audio settings

To adjust balance, change stereo image, or enhance bass, select **Options** > **Audio settings** and one of the options.

Music shop

In Music shop you can search, browse, and purchase music for downloading to your device. The variety, availability and appearance of Music shop services may vary.

Go to music shop

In Music shop you can search, browse, and purchase music for downloading to your device. The availability and appearance of this option may vary.

1 In **Music menu**, select **Options** > **Go to Music shop**. (This option is not available if music service Web address has not been defined. See "Music shop settings," p. 21)

2 Choose from the options on the screen to search, browse, or select ringing tones.

Music shop settings

The settings may also be predefined and not editable.

You must have valid Internet connections in order to use this service. For more details, see "Access points," p. 98.

The availability and appearance of music shop settings may vary.

You also must have the Web address of the music shop defined in music shop settings.

To change or define the Web address for the music shop, in Music menu, select **Options** > **Go to Music shop**. (Music shop may also be available as a separate shortcut in Operator folder.)

In Music shop, select **Options** > **Settings**. Define the following:

Address:—Enter the Web address of the music shop service.

Default access point:—Select whether to connect automatically or ask before connecting. Only available with WLAN.

User name:—Enter your user name for music shop. If this field is left empty, you may have to enter your user name at login.

Password:—Enter your password for music shop. If this field is left empty, you may have to enter your password at login.

Transfer music

You can transfer music from a compatible PC or other compatible devices using a USB cable or Bluetooth connection. For details about connecting your device, see "Connectivity," p. 86.

The requirements of the PC:

- Windows® XP operating system (or later)

- a compatible version of Windows Media® player application. You can get more detailed information about Windows Media® player compatibility from the N91 section of the Nokia Web site.
- Nokia PC Suite 6.7 or later. The sofware is available for installation on the sales package CD-ROM.

Connection method:

- USB data cable DKE-2 (miniUSB - USB)
- Bluetooth enabled compatible PC
- Active WLAN network

Transfer from PC

You can use three different methods to transfer files:

- To view your device on PC as an external hard drive where you can transfer any data files, make the connection with cable, Bluetooth connectivity or WLAN. If you are using the DKE-2 USB cable, select **USB** as the connection method. This method gives you total control of what you move and where, but you cannot see all the file information. You may not be able to transfer material that has been protected against copying. To refresh the Library after you have updated the song selection in your device, go to **Music menu**, and select **Options** > **Refresh Music library**.
- To synchronize music with Windows Media Player, attach the USB cable and select **Media player** as the connection method. This method is good if you want to synchronize all or some of the music files between a compatible PC and your device.
- To use Nokia Audio Manager in Nokia PC Suite, attach the DKE-2 USB cable and select **PC Suite** as the connection method.

USB cable connection

Connect the DKE-2 USB cable to a compatible PC and your device. If you are using Windows XP, your device is automatically recognized as an external hard drive. All the applications in your device are closed, and the phone goes offline.

To select the USB connection mode, use the joystick.

To change the default USB connection mode, press , select **Connect.** > **Data cbl.** and one of the options in **Data cable mode**.

Music transfer

The following instructions have been written based on information about the latest compatible Windows Media® player version available at the time of writing. However, the details of the instructions especially concerning the locations and naming of different options may not be

accurate and are subject to change. For more up-to-date information see the corresponding Windows Media Player guides and helps.

Manual synchronization

After connecting your device with a compatible PC, Windows Media player selects manual sync, if there is less than 4 GB of free storage room in your device or if there is not enough storage room to fit your entire library. With manual sync, you select the songs and playlists that you want to move, copy, or remove.

Enter a name that is used as the name of your device in Windows Media Player.

To transfer selected manually:

1. After your device is connected with Windows Media Player, select your device in the Navigation pane, if more than one device is connected.
2. Drag the songs or albums to the List pane for synchronisation. To remove songs or albums, select an item in the list and click **Remove from list**.
3. Check that the List pane contains the files you want to synchronize and that there is enough capacity on the device. Click **Start Sync.** to start the synchronization.

Automatic synchronization

To change the default file transfer option in Windows Media Player, click the arrow under **Sync** , select your device, and click **Set up Sync.** Clear or select the **Sync this device automatically** check box.

If the **Sync this device automatically** check box is selected and you connect your device, your N91 music library is automatically updated based on playlists you have selected in Windows Media Player.

If no playlists have been selected, the whole PC music library is selected for syncronization. Note that your PC library may contain more files that can fit to the hard drive of your Nokia N91. See Windows Media player help for more information.

The playlists on your device are not synchronized with the Windows Media player playlists.

Playlists

Create a playlist

To create a playlist of songs:

1. Select **Options** > **Create playlist** in **Playlists** view.
2. Enter a name for the playlist, and select **OK** or press the joystick.

3 Expand and collapse an artist title to find the songs you want to include in the playlist. Press the joystick to add items. To show the song list under an artist title, move the joystick to the right. To hide the song list, move the joystick to the left.

4 When you have made your selections, select **Done**.

Add songs to a playlist

You can add songs to an existing playlist or first create a new playlist.

1 In a song list, select **Options** > **Add songs**.

2 Expand and collapse an artist title to find the songs you want to include in the playlist. Press the joystick to add items. To show the song list under an artist title, move the joystick to the right. To hide the song list, move the joystick to the left.

3 When you have made your selections, select **Done**.

Tip! You can even add songs to a playlist that is currently playing. To search for a song start entering letters with the keypad.

Reorder songs

1 Select **Options** > **Reorder list** in a playlist view.

2 The selected song is ready to be moved. Scroll up or down to move it or leave it there.

3 Select **Drop**.

4 Scroll to select the song you want to move.

5 Select **Grab**.

6 Scroll to a new position.

7 Select **Drop**.

8 When you have reordered the list, select **Done**.

Tip! To grab and drop, you can also press the joystick.

Remove songs from a playlist

You can remove songs from a playlist.

1 Select a song in a playlist.

2 Select **Options** > **Remove from playlist**.

3 Select **Yes** to confirm. This deletion does not delete the song from the device; it only removes it from the playlist.

Add items from views to a playlist

You can add songs, albums, artists, genres, and composers from different views of the **Music menu** to existing or new playlists.

1 In a view opened from the **Music menu**, select an item.

2 Select **Options** > **Add to a playlist** > **Saved playlist** or **New playlist**.

3 If you selected **New playlist**, enter a name for the playlist, and select **OK**. If you selected **Saved playlist**, select the playlist and **OK**.

Tip! In some of the views you can select multiple songs to be added by selecting **Options** > **Mark**.

Tip! You can add a song that you are currently listening to to a playlist.

Playlist details

In **Playlists** view, Select **Options** > **Playlist details**. The following information is displayed:

Playlist name:—The name of the playlist

Songs—The current number of songs in the playlist

Location—The location of the playlist on the hard drive

Date—The date the playlist was last modified

Time—The time the playlist was last modified

Automatic playlists

The following playlists appear in the playlist view automatically:

Most played tracks—Includes the 40 most-played songs listed from the most recently played songs

Recent tracks—Includes 40 most recently played songs in reverse order, ending with the most recently played song.

Recent additions—Includes songs that you have added to the library within the past week.

Available options

The following items are available as options in different views of Music player.

Go to Now playing—Opens a view that displays information about the song that is currently being played

Go to Music shop—Opens wireless connection to a Web shop, where you can search for, browse, and purchase music

Refresh Music library—Updates the library list by scanning the hard drive for new songs and removing broken links

Song details/**Playlist details**/**Music library details**—Displays information about the selected item

Help—Opens the help application

New playlist—Opens a view where you can create a new playlist

Send—Allows you to send the selected item to another compatible device

Delete—Removes the selected item after confirmation. The song is permanently deleted from the hard drive.

Rename—Renames the selected item

Add to a playlist—Add the selected item to an existing playlist or new playlist

Album art—Opens a view which will show you album art currently attached to the selected album or song You can change or add album art using images on your phone.

Mark/Unmark—Allows you to select several items for action

Set as ringing tone—Selects the selected song as the ringing tone of the active profile. If you select songs from the hard drive to be used as ringing tones, they may not work with vibra. For optimal performance, store the ringing tones in the device memory.

Reorder list—Rearrange the songs of a playlist

Options only in Now Playing view

Go to Music menu—Opens the **Music menu** view

Shuffle—Switches the shuffle mode on or off. In shuffle mode the songs are played in random order.

Repeat—Repeats the song or all songs in the list that is currently being played.

Equalizer—Opens a view where you can adjust the sound frequency balance.

Audio settings—Opens a view where you can change the sound balance, stereo image and bass response.

Rename items

1. Select an **Artist**, **Album**, **Genre**, or **Composer** from the list.
2. Select **Options** > **Rename**.
3. Enter the new name, and select **OK**. All songs belonging to the selected artist, album, genre, or composer are updated.

View and edit song details

In a song view, Select **Options** > **Song details**. The following information is displayed and editable.

Song name—The name of the song

Artist—The name of the artist

Album—The name of the album to which this song belongs

Track number—The song position in the album to which this song belongs

Year—The year in which the album was released

Genre—The style category to which this song belongs

Composer—The name of the composer of this song

Comment—Any other information about this song

To view details like **Format**, **Duration**, and **Copyright**, select **Options** > **View file details** in **Song details** view.

Library details

In **Music menu** select **Options** > **Music library details**. The following information is displayed:

Songs—The current number of songs in the library

Duration—The length of current songs together

Used—The total size of the files in the music libray

Available—Available hard disk space in gigabytes

Refreshed—The date when the libraray was last modified

Set a song as a ringing tone

1 Select a song.

2 Select **Options** > **Set as ringing tone**. The selected song has been set as the ringing tone for the active profile.

Change album art

1 Select a song.

2 Select **Options** > **Album art**.

3 Select **Options** > **Change**.

4 Scroll throough images from your Gallery, and select **OK**.

5 To restore the original album art, select **Restore original**.

About encoded audio

Applications like Windows Media Player and Nokia Audio Manager encode audio files with codecs like WMA or MP3 to save storage space. The encoding is done by removing those parts of the audio signal that are not audible or only barely audible by human ear.

Regardless of the encoder and codec quality, the sound quality of the resulting file is not as good as the original.

The basic codecs supported by this device are AAC, WMA, and MP3. All of these have different variations.

Bitrates

When encoding audio, the quality of the resulting sound depends on the sample rate and bit rate used for the encoding. Sample rate is expressed as thousands of samples per second (kHz), and for CD music the sample rate is fixed at 44.1 kHz. Bit rate is expressed as kilobits per second (kb/s). The higher the bitrate, the better the sound quality.

The required quality level may depend on your demands, the headphones used, and the surrounding noise. For MP3s a bitrate between 128 and 192 kb/s usually gives results good enough for listening to pop music with your device. WMA or AAC generally gives a similar result as MP3 with one step lower bitrate (96-160 kb/s). Variations of AAC, such as AAC+ and enhanced AAC+ (eAAC+) provide CD

quality audio at bitrates as low as 48 kb/s. Classical music and music with nuances usually require a higher bitrate than basic pop music.

For speech or other sources where the quality of the sound is less important than saving storage space, you can use bitrates between 8 and 64 kb/s. At low bitrates WMA or AAC generally gives better results than MP3.

Variable bitrate

Your device also supports variable bitrate (VBR). Variable bitrate means that the bit rate of the encoding varies with the complexity of the encoded material. When using VBR, the encoding is optimized to preserve a constant audio quality throughout the track, rather than losing quality for complex music sections as with constant bitrate (CBR) encoding.

Personalize your device

Your service provider may have requested that certain features be disabled or not activated in your device. If so, they will not appear on your device menu. Your device may also have been specially configured for your network provider. This configuration may include changes in menu names, menu order, and icons. Contact your service provider for more information.

The open applications, and folders that have applications open, are indicated by a small icon in the main menu.

Profiles—Set tones

To set and customize the ringing tones, message alert tones, and other tones for different events, environments, or caller groups, press , and select **Tools** > **Profiles**. If the currently selected profile is other than **Normal**, the profile name is shown at the top of the display in the standby mode.

To change the profile, press in the standby mode. Scroll to the profile you want to activate, and select **OK**.

To modify a profile, press , and select **Tools** > **Profiles**. Scroll to the profile, and select **Options** > **Customize**.

Scroll to the setting you want to change, and press the joystick to open the choices. Tones stored on the hard drive are indicated with . You can scroll through the tone list and listen to each one before you make your selection. To stop the sound, press any key.

When you choose a tone, **Tone downloads** opens a list of bookmarks. You can select a bookmark and connect to a Web page to download tones.

If you want the caller's name to be said when your device rings, select **Options** > **Customize**, and set **Say caller's name** to **On**. The caller's name must be saved in **Contacts**.

To create a new profile, select **Options** > **Create new**.

If listen to music and **Music** profile is selected, you only hear a beep for incoming calls and all the other alert tones are switched off.

Offline profile

The **Offline** profile lets you use the device without connecting to the wireless network. When you activate the **Offline** profile, the connection to the wireless network is turned off, as indicated by in the signal strength indicator area. All wireless phone signals to and from the device are prevented. If you try to send messages, they are placed in the outbox to be sent later.

When you have activated the **Offline** profile, you can still use the wireless LAN (if available) to read your e-mail or browse on the Internet, for example. Remember to comply with any applicable safety requirements when establishing and using a wireless LAN connection.
See "Wireless LAN," p. 13.

Warning: In the Offline profile you cannot make (or receive) any calls, except make calls to certain emergency numbers, or use other features that require network coverage. To make calls, you must first activate the phone function by changing profiles. If the device has been locked, enter the lock code.

Warning: Your device must be switched on to use **Offline** profile. Do not switch the device on when wireless device use is prohibited or when it may cause interference or danger.

To leave the **Offline** profile, press , and select **Profiles**, select another profile, and select **Options** > **Activate** > **Yes**. The device re-enables wireless transmissions (providing there is sufficient signal strength). If Bluetooth connectivity is activated before entering the **Offline** profile, it will be deactivated. Bluetooth connectivity is automatically reactivated after leaving the **Offline** profile.
See "Settings," p. 86.

Change the look of your device

To change the look of the display, such as the wallpaper and icons, press , and select **Tools** > **Themes**. The active theme is indicated by . In **Themes** you can group together elements from other themes or select images from the **Gallery** to personalize themes further. The themes on the hard drive are indicated by .

To open a browser connection and download more themes, select **Theme downloads**.

To activate a theme, scroll to it, and select **Options** > **Apply**. To preview a theme before activating it, select **Options** > **Preview**.

To edit themes, scroll to a theme, and select **Options** > **Edit** to change the **Wallpaper** and **Power saver** settings.

To restore the selected theme to its original settings, select **Options** > **Restore orig. theme** when you edit a theme.

Active standby mode

Use your standby display for fast access to your most frequently used applications. By default, the active standby mode is on.

Press , select **Tools** > **Settings** > **Phone** > **Standby mode** > **Active standby**, and press the joystick to switch the active standby on or off.

The active standby display is shown with default applications across the top of the screen, and calendar, to-do, and player events are listed below.

Scroll to an application or event, and press the joystick.

The standard joystick shortcuts available in the standby mode cannot be used when the active standby mode is on.

To change the default application shortcuts, press , select **Tools** > **Settings** > **Phone** > **Standby mode** > **Active standby apps.**, and press the joystick.

Some shortcuts may be fixed, and you cannot change them.

Make calls

Voice calls

Tip! To increase or decrease the volume during a call, press the keys on the left side of the device. If you have set the volume to **Mute**, you cannot adjust the volume. To adjust the volume, select **Unmute**, then press the volume keys.

1 In the standby mode, enter the phone number, including the area code. Press to remove a number.

For international calls, press * twice for the + character (replaces the international access code), and enter the country code, area code (omit the leading zero if necessary), and phone number.

2 Press to call the number.

3 Press to end the call (or to cancel the call attempt).

Tip! You can also use the key on the remote control to answer and end a call.

Pressing always ends a call, even if another application is active.

To make a call from **Contacts**, press , and select **Contacts**. Scroll to the desired name, or enter the first letters of the name to the search field. Matching contacts are listed. Press to call.

You must copy the contacts from your SIM to **Contacts** before you can make a call this way. See "Copy contacts," p. 37.

To call your voice mailbox (network service), press and hold 1 in the standby mode. See also "Call forwarding," p. 103.

Tip! To change the phone number of your voice mailbox, press , and select **Tools** > **Voic. mail** > **Options** > **Change number**. Enter the number (obtained from your service provider), and select **OK**.

To call a recently dialed number in the standby mode, press to access a list of the 20 last numbers you called or attempted to call. Scroll to the number you want, and press to call the number.

Make a conference call

1 Make a call to the first participant.

2 To make a call to another participant, select **Options** > **New call**. The first call is automatically put on hold.

3 When the new call is answered, to join the first participant in the conference call, select **Options** > **Conference**.

To add a new person to the call, repeat step 2, and select **Options** > **Conference** > **Add to conference**.

Your device supports conference calls between a maximum of six participants, including yourself.

To have a private conversation with one of the participants, select **Options** > **Conference** > **Private**. Select a participant, and select **Private**. The conference call is put on hold on your device. The other participants can still continue the conference call. Once you have finished the private conversation, select **Options** > **Add to conference** to return to the conference call.

To drop a participant, select **Options** > **Conference**> **Drop participant**, scroll to the participant, and select **Drop**.

4 To end the active conference call, press .

1-touch dial a phone number

To activate 1-touch dialing, press , and select **Tools**> **Settings** > **Call** > **1-touch dialing** > **On**.

To assign a phone number to one of the 1-touch dialing keys (2 – 9), press , and select **Tools** > **Spd. dial**. Scroll to the key to which you want to assign the phone number, and select **Options** > **Assign**. 1 is reserved for the voice mailbox.

To call in the standby mode, press the speed dial key and .

Voice dialing

Your device supports enhanced voice commands. Enhanced voice commands are not dependent on the speaker's voice, so the user does not record voice tags in advance. Instead, the device creates a voice tag for the entries in contacts, and compares the spoken voice tag to it. The voice recognition in the device adapts to the main user's voice to recognize the voice commands better.

The voice tag for a contact is the name or nickname that is saved on the contact card. To listen to the synthesized voice tag, open a contact card, and select **Options** > **Play voice tag**.

Make a call with a voice tag

Note: Using voice tags may be difficult in a noisy environment or during an emergency, so you should not rely solely upon voice dialing in all circumstances.

When you use voice dialing, the loudspeaker is in use. Hold the device at a short distance away when you say the voice tag.

1 To start voice dialing, press and hold the right selection key. If you are using a compatible headset with the headset key, press and hold the headset key to start voice dialing.

2 A short tone is played, and **Speak now** is displayed. Say clearly the name or nickname that is saved on the contact card.

3 The device plays a synthesized voice tag for the recognized contact in the selected device language, and displays the name and number. After a timeout of 1.5 seconds, the device dials the number.

If the recognized contact was not correct, select **Next** to view a list of other matches, or **Quit** to cancel voice dialing.

If several numbers are saved under the name, the device selects the default number, if it has been set. Otherwise the device selects the first available number of the following: **Mobile**, **Mobile (home)**, **mobile (work)**, **Telephone**, **Tel. (home)**, and **Tel. (work)**.

Answer or decline a call

To answer the call, press or open the slide cover.

To mute the ringing tone when a call comes in, select **Silence**.

Tip! You can also use the key on the remote control to answer and end a call.

If you do not want to answer a call, press to decline it. The caller hears a line busy tone. If you have activated the **Call forward** > **If busy** function to forward calls, declining an incoming call also forwards the call. See "Call forwarding," p. 103.

When you decline an incoming call, you can also send a text message to the caller informing why you cannot answer the call. Select **Options** > **Send text message**. You can edit the text before sending it. To set up this option and write a standard text message, see "Call settings," p. 96.

Call waiting (network service)

You can answer a call while you have another call in progress if you have activated **Call waiting** in **Settings** > **Call** > **Call waiting**.

Tip! To change the device tones for different environments and events, for example, when you want your device to be silent, see "Profiles—Set tones," p. 29.

To answer the waiting call, press . The first call is put on hold.

To switch between the two calls, select **Swap**. Select **Options** > **Transfer** to connect an incoming call or a call on hold with an active call and to disconnect yourself from the calls. To end the active call, press . or close the slide cover. To end both calls, select **Options** > **End all calls**.

Options during a voice call

Many of the options that you can use during a voice call are network services. Select **Options** during a call for some of the following options: **Mute** or **Unmute**, **Answer**, **Decline**, **Swap**, **Hold** or **Unhold**, **Activate handset**, **Activate loudsp.**, or **Activate handsfree** (if a compatible headset with Bluetooth connectivity is attached), **End active call** or **End all calls**, **New call**, **Conference**, and **Transfer**. Select from the following:

Replace—To end an active call and replace it by answering the waiting call.

Send MMS (in UMTS networks only)—To send an image or a video in a multimedia message to the other participant of the call. You can edit the message and change the recipient before sending. Press ⌐ to send the file to a compatible device (network service).

Send touch tones—To send touch tone strings, for example, a password. Enter the touch tone string or search for it in **Contacts**. To enter a wait character (**w**) or a pause character (**p**), press * repeatedly. Select **OK** to send the tone.

Tip! You can add touch tones to the **Phone number** or **Touch tones** fields in a contact card.

Contacts (Phonebook)

Press , and select **Contacts**. In **Contacts** you can add a personal ringing tone or a thumbnail image to a contact card. You can also create contact groups, which allow you to send text messages or e-mail to many recipients at the same time. You can add received contact information (business cards) to contacts. See "Data and settings," p. 57. Contact information can only be sent to or received from compatible devices.

Tip! Regularly back up the device information. You can restore the information, such as contacts, to the device later. You can use Nokia PC Suite to back up your contacts to a compatible PC. See the CD-ROM supplied with your device.

Save names and numbers

1 Select **Options** > **New contact**.

2 Fill in the fields that you want, and select **Done**.

To edit contact cards in **Contacts**, scroll to the contact card you want to edit, and select **Options** > **Edit**.

To delete a contact card in **Contacts**, select a card, and press . To delete several contact cards at the same time, press and hold and press the joystick to mark each contact you want to delete. Press to delete the marked contacts.

Tip! To add and edit contact cards, use Nokia Contacts Editor available in Nokia PC Suite. See the CD-ROM supplied with your device.

To attach a small thumbnail image to a contact card, open the contact card, and select **Options** > **Edit** > **Options** > **Add thumbnail**. The thumbnail image is shown when the contact calls.

Tip! 1-touch dialing is a quick way to call frequently used numbers. You can assign 1-touch dialing keys to eight phone numbers. See "1-touch dial a phone number," p. 33.

Tip! To send contact information, select the card you want to send. Select **Options** > **Send** > **Via text message**, **Via multimedia**, or **Via Bluetooth**. See "Messaging," p. 50 and "Send data using Bluetooth connectivity," p. 87.

To add a contact to a group, select **Options** > **Add to group:** (shown only if you have created a group). See "Create contact groups," p. 38.

To view the amount of contacts, groups, and the available memory, select **Contacts** > **Options** > **Contacts info**.

Default numbers and addresses

You can assign default numbers or addresses to a contact card. In this way if a contact has several numbers or addresses, you can easily call or send a message to the contact at a certain number or address. The default number is also used in voice dialing.

1 In contacts, select a contact, and press the joystick.

2 Select **Options** > **Defaults**.

3 Select a default to which you want to add a number or an address, and select **Assign**.

4 Select a number or an address you want to set as a default.

The default number or address is underlined in the contact card.

SIM directory and SIM services

For availability and information on using SIM card services, contact your SIM card vendor. This may be the service provider, network operator, or other vendor.

Press , and select **Contacts** > **Options** > **SIM contacts** > **SIM directory** to see the names and numbers stored on the SIM card. In the SIM directory you can add, edit, or copy numbers to contacts, and you can make calls.

To view the list of fixed dialing numbers, select **Options** > **SIM contacts** > **Fixed dialing contacts**. This setting is only shown if supported by your SIM card.

To restrict calls from your device to selected phone numbers, select **Options** > **Activ. fixed dialing**. To add new numbers to the fixed dialing list, select **Options** > **New SIM contact**. You need the PIN2 code for these functions.

When you use **Fixed dialing**, packet data connections are not possible, except when sending text messages over a packet data connection. In this case, the message center number and the recipient's phone number must be included on the fixed dialing list.

When fixed dialing is activated, calls may be possible to the official emergency number programmed into your device.

Copy contacts

To copy names and numbers from a SIM card to your device, press , and select **Contacts** > **Options** > **SIM contacts** > **SIM directory**. Select the names you want to copy and **Options** > **Copy to Contacts**.

To copy contacts to your SIM card, press , and select **Contacts**. Select the names you want to copy and **Options** > **Copy** > **To SIM directory**. Only the contact card fields supported by your SIM card are copied.

To edit phone number to which calls from your device are allowed, press , and select **Contacts** > **Options** > **SIM contacts** > **Fixed dialing contacts**. Scroll to the contact and select **Options** > **Edit**. You may need your PIN2 code to edit your fixed dialing numbers.

Tip! You can synchronize your contacts to a compatible PC with Nokia PC Suite. See the CD-ROM supplied with your device.

Add ringing tones for contacts

When a contact or group member calls you, the device plays the chosen ringing tone (if the caller's telephone number is sent with the call and your device recognizes it).

1. Press the joystick to open a contact card, or go to the groups list, and select a contact group.
2. Select **Options** > **Ringing tone**. A list of ringing tones opens.
3. Select the ringing tone you want to use for the individual contact or the selected group.

To remove the ringing tone, select **Default tone** from the list of ringing tones.

Create contact groups

1. In **Contacts**, move the joystick right to open the groups list.
2. Select **Options** > **New group**.
3. Write a name for the group or use the default name **Group %N** (number), and select **OK**.
4. Open the group, and select **Options** > **Add members**.
5. Scroll to a contact, and press the joystick to mark it. To add multiple members at a time, repeat this action on all the contacts you want to add.
6. Select **OK** to add the contacts to the group.

To rename a group, select **Options** > **Rename**, enter the new name, and select **OK**.

Remove members from a group

1. In the groups list, open the group you want to modify.
2. Scroll to the contact, and select **Options** > **Remove from group**.
3. Select **Yes** to remove the contact from the group.

Tip! To check to which groups a contact belongs, scroll to the contact, and select **Options** > **Belongs to groups**.

Media applications

Camera

The Nokia N91 supports an image capture resolution of 1600 x 1200 pixels. The image resolution in these materials may appear different.

To activate the camera, press in the Standby view, and select **Camera**. The camera starts, and you can see the view to be captured.

If the camera is in **Video** mode, move the joystick to the left to start **Image** mode.

The images are automatically saved in the gallery in .jpg format. You can send images in a multimedia message as an e-mail attachment or by using Bluetooth connectivity.

Take pictures

To take a picture, press the joystick. Do not move the device before the image is saved. The image is saved automatically in the **Images** folder of **Gallery**. See "Gallery," p. 43.

Saving the captured image may take longer if you change the zoom, lighting, or color settings.

When taking a picture, note the following:

- To keep the camera still, use both hands.
- To adjust color before taking a picture, select **Options** > **Adjust** > **White balance** or **Color tone**. See "Adjust color," p. 40.
- The quality of a zoomed picture is lower than that of a nonzoomed picture.
- The camera goes into battery saving mode if there are no key presses for a moment. To continue taking pictures, press the joystick.

After the image is taken:

- To discard the image, select **Options** > **Delete**.
- To send the image **Via multimedia**, **Via e-mail**, or **Via Bluetooth**, select **Options** > **Send**. For more information, see "Messaging," p. 50 and "Bluetooth connectivity," p. 86. This option is not available during an active call.
- To send an image to the other party during an active call, select **Options** > **Send MMS**.

Exposure modes

- To set the camera to take six pictures in a sequence, select **Options** > **Sequence mode** . You can also use the sequence mode with the **Self-timer**. See "You in the picture—self-timer," p. 40. The pictures are automatically saved in **Gallery**.

  After you take the pictures, they are shown in a grid on the main display. To view a picture, press the joystick to open it.

- To set the camera to take pictures in low light conditions, select **Options** > **Night mode**

You in the picture—self-timer

Use the self-timer to delay the capture so that you can include yourself in the picture. To set the self-timer delay, select **Options** > **Self-timer** > **10 seconds**, **20 seconds**, or **30 seconds**. To activate the self-timer, select **Activate**. The self-timer indicator ( ) blinks and the device beeps when the timer is running. The camera takes the picture after the selected delay elapses. You can also use the **Self-timer** in the sequence mode. See "Exposure modes," p. 40.

Adjust color

To enable the camera to reproduce colors more accurately, or to add effects to your pictures or videos, select **Options** > **Adjust**, and select from the following options:

White balance—Select the current lighting condition from the list. This allows the camera to reproduce colors more accurately.

Color tone—Select a color effect from the list.

The screen display changes to match any settings made, showing you how the change affects the pictures or videos.

The settings return to the default settings when you close the camera.

Adjust still image camera settings

To change the main settings, select **Options** > **Settings**. These settings remain the same until you change them. Select from the following options:

Image quality—**High**, **Normal**, or **Basic**. The better the image quality, the more memory the image consumes.

Image resolution—**High**(resolution 1600x1200), **Normal** (resolution 800x600), or **Basic** (resolution 640x480). The higher the image resolution, the more memory the image consumes. The quality of a zoomed picture is lower than that of a nonzoomed picture.

Default image name— Set a name for the pictures taken. By default, the pictures are named in the format 'Image'. For example, if you set Beach as the default name, Camera will name the pictures you take Beach, Beach(01), Beach(02), and so on, until you change the setting again.

Show captured img.—Select **Yes** if you want to see the captured image after it is taken, or **No** if you want to continue taking pictures immediately.

Memory in use—Select where to store your images.

Viewfinder

Viewfinder shows the following:

The Memory in use indicator (1) shows where the images are currently stored: in device memory or in hard drive .

The images counter (2) on top of the viewfinder shows an estimate of how many images, depending on the selected image quality, can fit in the remaining memory of your device.

The zoom indicator (3) shows the zoom level. Move the joystick up and down to zoom in and zoom out.

The **Night mode** indicator (4) shows that the camera has been optimized for low-light photography.

The **Sequence mode** indicator (5) shows that the camera is in sequence mode.

The self-timer indicator (6) shows that the self-timer is activated. See "You in the picture—self-timer," p. 40.

Tips on taking good photographs

Picture quality

Use the appropriate picture quality. The camera has three picture quality modes (**Basic**, **Normal**, and **High**). Use the **High** setting to make sure that the camera produces the best picture quality available. Note however, that better picture quality requires more storage space. For MMS and e-mail attachments it may be necessary to use **Normal**, or **Basic** setting. You can define the quality in **Options** > **Settings**.

Background

Use a simple background. For portraits and other pictures with people, avoid having your subject in front of a cluttered, complex background that may distract attention from the subject. Move the camera, or the subject, when these conditions cannot be met. Move the camera closer to the object to take clearer portraits.

Depth

When shooting landscapes and sceneries, add depth to your pictures by placing objects in the foreground. If the foreground object is too close to the camera, it may be blurred.

Lighting conditions

Changing the source, amount, and direction of light can change photographs dramatically. Here are some typical lighting conditions:

- **Light source behind the subject.** Avoid placing your subject in front of a strong light source. If the light source is behind the subject or visible in the display, the resulting picture may have weak contrast, may be too dark, and may contain unwanted light effects.
- **Sidelit subject.** Strong side light gives a dramatic effect but may be too harsh, resulting in too much contrast.
- **Light source in front of the subject.** Harsh sunlight may cause the subjects to squint their eyes. The contrast may also be too high.
- **Optimal lighting** is found in situations where there is plenty of diffused, soft light available, for example, on a bright, partially cloudy day or on a sunny day in the shadow of trees.

Record videos

To activate the camera, press , and select **Camera** in the Standby view. The camera starts, and you can see the view to be captured.

If the camera is in **Image** mode, open the video recorder view by moving the joystick right.

To adjust color before recording a video, select **Options** > **Adjust** > **White balance** or **Color tone**. See "Adjust color," p. 40.

1. Press the joystick to start recording. The record icon is shown and a tone sounds, indicating to the subject that video is being recorded.
2. To pause recording at any time, select **Pause**. The pause icon blinks on the display. Video recording automatically stops if the recording is set to pause and there are no key presses within a minute.
3. Select **Continue** to resume recording.
4. Select **Stop**, or press the joystick to stop recording. The video clip is automatically saved to the **Video clips** folder of **Gallery**. See "Gallery," p. 43.

Viewfinder indicators show the following:

The microphone indicator shows that the microphone is muted.

The self-timer indicator shows that the self-timer is activated. See "You in the picture—self-timer," p. 40.

The zoom indicator shows the zoom level. Move the joystick up and down to zoom in and out.

After a video clip is recorded:

- To immediately play the video clip you just recorded, select **Options** > **Play**.
- To discard the video, select **Options** > **Delete**.
- To return to the viewfinder to record a new video, press the joystick.
- To send the video, select **Options** > **Send** > **Via multimedia**, **Via e-mail**, or **Via Bluetooth**. For more information, see "Messaging," p. 50 and "Bluetooth connectivity", p. 86. This option is not available during an active call.
- To send a video clip to the other party during an active call, select **Options** > **Send MMS**.

Video settings

There are two kinds of settings for the video recorder: temporary settings and default settings. To adjust contrast, brightness and color settings, see "Adjust color," p. 40. The temporaray settings return to the default settings after you close the camera, but the default settings remain the same until you change them again. To change the default settings, select **Options** > **Settings** and from the following:

Length: %U—To optimize video clip for MMS, select **Short**.

Video resolution—**High** (CIF, 352x288), **Normal** (QCIF 176x144), or **Basic** (SubQCIF 128x96). The higher the resolution, the more memory the video consumes.

Default video name:—Set a name for the video taken. Select **Date** or **Text**. If you select **Text**, the videos are named in the format Video by default. For example, if you set Beach as the default name, Camera will name the video clips you take Beach, Beach(01), Beach(02), and so on, until you change the setting again.

Gallery

To store and organise your images, video clips, sound clips, playlists, and streaming links, press , and select **Media** > **Gallery**. To open the gallery in the camcorder select **Options** > **Go to Gallery**.

Select **Images**, **Video clips**, **Tracks**, **Sound clips**, **Streaming links**, or **All files**, and press the joystick to open it.

You can browse and open folders, and mark, copy, and move items to folders. You can also create folders, and mark, copy, and add items to folders. See "Folders," p. 44.

Tip! You can transfer images from your device to a compatible PC with Nokia Phone Browser available in Nokia PC Suite. See the CD-ROM supplied with your device.

Press the joystick to open a file. Video clips, .ram files, and streaming links are opened and played in the RealPlayer application and sound clips in the Music Player application. See "RealPlayer™," p. 45. and "Music player," p. 19. See "View images and video clips," p. 44.

To rename a file, select a file and **Options** > **Rename**.

To download files into **Gallery** in one of the main folders using the browser, select **Graphic downl.** or **Video downl.**, **Track downl.**, or **Sound downl.**. The browser opens, and you can select a bookmark from which the site can download.

View images and video clips

Photos taken are stored in the **Images** folder and video clips are stored in the **Video clips** folder in **Gallery**. The quality of a video clip is indicated with one of the following icons: (High), (Normal), and (Sharing). Images and video clips can also be sent to you in a multimedia message, as an e-mail attachment, or through a Bluetooth connection. To be able to view a received image or video clip in the gallery or in the media player, you must save it.

Open the **Images**or **Video clips** folder in **Gallery**. To browse the files, scroll up or down and press the joystick to open a file for viewing.

To delete an image or a video clip, press .

Folders

With folders you can conveniently manage your images and video clips.

To create a new folder, select **Options** > **Organize** > **New folder**. Enter a name for the folder, and select **OK**.

To add a picture or a video clip to an folder in the gallery, scroll to a picture or video clip, and select **Options** > **Organize** > **Copy to hard drive** / **Copy to phone mem.** or **Move to hard drive** / **Move to phone mem.** depending on the current location of the file. A list of folders opens. Select the folder to which you want to add the picture or video clip, and press the joystick.

To remove a file from an folder, press . The file is not deleted from the **Images** folder or **Video clips** folder in **Gallery**.

Music player

See "Music player" on p. 19 for details.

RealPlayer™

Press , and select **Media** > **RealPlayer**. With **RealPlayer**, you can play video clips or stream media files over the air. You can activate a streaming link when you browse Web pages or store it in the hard drive.

RealPlayer supports files with extensions such as .awb or .m4a, but not all the variations of file formats. For example, some .mp4 files may include content that is not compliant with 3GPP standards and, therefore, cannot be opened.

Tip! You can transfer music files from your device to your hard drive with Nokia PC Suite. See the CD-ROM supplied with your device.

Play video

1 To play a media file stored in phone memory, or on the hard drive, select **Options** > **Open** and select:

Most recent clips—To play one of the last six files played in **RealPlayer**.

Saved clip—To play a file saved in **Gallery**. See "Gallery," p. 43.

2 Scroll to a file, and press the joystick to play the file.

Tip! To view a video clip in full screen mode, press **2**. Press again to change back to normal screen mode.

Stream content over the air

Many service providers require you to use an Internet access point (IAP) for your default access point. Other service providers allow you to use a WAP access point.

The access points may be configured when you first start your device.

Contact your service provider for more information.

In **RealPlayer**, you can only open an rtsp:// URL address. However, **RealPlayer** will recognize an http link to a .ram file.

To stream content over the air, select a streaming link saved in **Gallery**, on a Web page, or received in a text message or multimedia message. Before live content begins streaming, your device connects to the site and starts loading the content. The content is not saved in your device.

Receive RealPlayer settings

You may receive **RealPlayer** settings in a special text message from the network operator or service provider. See "Data and settings," p. 57. For more information, contact your network operator or service provider.

Change the RealPlayer settings

Select **Options** > **Settings** and from the following:

Video—To adjust the contrast of the video and to have **RealPlayer** automatically repeat video clips after they finish playing.

Audio—To select whether you want to repeat playing of track lists and play sound clips on a track list in random order.

Connection—To select whether to use a proxy server, change the default access point, and set the port range used when connecting. Contact your service provider for the correct settings.

Proxy sett.:

- **Use proxy**—Select **Yes**.
- **Proxy serv. address**—Enter the IP address of the proxy server.
- **Proxy port number**—Enter the port number of the proxy server.

Glossary: Proxy servers are intermediate servers between media servers and their users. Some service providers use them to provide additional security or speed up access to browser pages that contain sound or video clips.

Netw. sett.:

- **Default access point**—Scroll to the access point you want to use to connect to the Internet, and press the joystick.
- **Online time**—Set the time for **RealPlayer** to disconnect from the network when you have paused a media clip playing through a network link, select **User defined**, and press the joystick. Enter the time, and select **OK**.
- **Lowest UDP port**—Enter the lowest port number of the servers port range. The minimum value is 6970.
- **Highest UDP port**—Enter the highest port number of the servers port range. The maximum value is 32000.

  Select **Options** > **Advanced settings** to edit the bandwidth values for different networks.

FlashPlayer

With Flash player, you can view, play, and interact with flash files made for mobile devices.

Organize flash files

Press , and select **Media** >**Flash Player**. Scroll right.

To open a folder or play a flash file, scroll to it, and press the joystick.

To send a flash file to a compatible device, scroll to it, and press the call key.

To copy a flash file to another folder, select **Organize** > **Copy to folder**.

To move a flash file to another folder, select **Organize** > **Move to folder**

To create a folder to organize your flash files, select **Organize** > **New folder**.

The available options may vary.

To delete a flash file, scroll to it, and press C.

Play flash files

Press , and select **Media** > **Flash Player**. Scroll to a flash file and press the joystick.

Select **Options** and from the following:

Note: Options may not be available in all flash files.

- **Pause** to pause the playback.
- **Stop** to stop the playback.
- **Volume** to adjust the playback volume. To increase or decrease the volume scroll left or right.
- **Quality** to select the playback quality.

  If the playback appears uneven and slow, change the **Quality** setting to **Normal** or **Low**.
- **Full screen** to play the file using the entire display. To return to the normal screen, select **Normal screen**.

  Even though not visible in full screen, the key functions may still be available when you press either key below the display.
- **Fit to screen** to play the file in its original size after zooming it.
- **Pan mode on** to be able to move around the display with the joystick when you have zoomed in.

The available options may vary.

Recorder

Press , and select **Media** > **Recorder** to record telephone conversations and voice memos. If you are recording a telephone conversation, both parties hear a tone every 5 seconds during recording.

The recorder cannot be used when a data call or GPRS connection is active.

Radio

Press , and select **Radio**. You can use the application as a traditional FM radio with automatic tuning and saved stations, or with parallel visual information related to the radio program on the display, if you tune to stations that offer Visual Radio service. The Visual Radio service uses packet data (network service). You can listen to the FM radio while using other applications.

If you do not have access to the Visual Radio service, the operators and radio stations in your area do not necessarily support Visual Radio. The Visual Radio service may not be available in all areas and countries.

You can normally make a call or answer an incoming call while listening to the radio. The radio is muted when there is an active call.

The radio selects the used frequency band based on the country information received from the network. If this information is not available, you may be asked to select the region whereyou are located, or you can select the region in the Visual Radio settings.

Listen to the radio

Note that the quality of the radio broadcast depends on the coverage of the radio station in that particular area.

The FM radio depends on an antenna other than the wireless device antenna. A compatible headset or enhancement needs to be attached to the device for the FM radio to function properly.

Press , and select **Radio**. To start a station search, select or and press the joystick. To change the frequency manually, select **Options** > **Manual tuning**.

If you have previously saved radio stations, select or on the display and press the joystick to go to the next or previous saved station, or press the corresponding number key to select the memory location of a station.

To adjust the volume, press and hold or on the left side of the device. To listen to the radio using the loudspeaker, select **Options** > **Activate loudspeaker**.

Warning: Listen to music at a moderate level. Continuous exposure to high volume may damage your hearing. Do not hold the device near your ear when the loudspeaker is in use, because the volume may be extremely loud.

To view available stations based on location, select **Options** > **Station directory** (network service).

To save the current tuned station to your station list, select **Options** > **Save station**. To open the list of your saved stations, select **Options** > **Stations**.

To return to the standby mode and leave the FM radio playing in the background, select **Options** > **Play in background**.

View visual content

To check availability and costs and to subscribe to the service, contact your network operator or service provider.

To view available visual content for a tuned station, select or **Options** > **Start visual service**. If the visual service ID has not been saved for the station, enter it, or select **Retrieve** to search for it in the station directory (network service).

When the connection to the visual service is established, the display shows the current visual content.

To adjust display settings for the visual content view, select **Options** > **Display settings**.

Saved stations

You can save up to 20 radio stations in the radio. To open your station list, select **Options** > **Stations**.

To listen to a saved station, select **Options** > **Station** > **Listen**. To view available visual content for a station with Visual Radio service, select **Options** > **Station** > **Start visual service**.

To change station details, select **Options** > **Station** > **Edit**.

Settings

Select **Options** > **Settings** and from the following:

Start-up tone—Select whether a tone is played when the application is started.

Auto-start service—Select **Yes** to have the Visual Radio service start automatically when you select a saved station that offers visual service.

Access point—Select the access point used for the data connection. You do not need an access point to use the application as an ordinary FM radio.

Current region—Select the region you are currently located in. This setting is displayed only if there was no network coverage when the application was started.

Messaging

Press , and select **Messag.**. In **Messag.**, you can create, send, receive, view, edit, and organize text messages, multimedia messages, e-mail messages, and special text messages containing data. You can also receive messages and data using Bluetooth connectivity, receive Web service messages, cell broadcast messages, and send service commands.

Note: Only devices that have compatible features can receive and display multimedia messages. The appearance of a message may vary depending on the receiving device.

When you open **Messag.**, you can see the **New message** function and the following folders:

Inbox—Received messages, except e-mail and cell broadcast messages, are stored here. E-mail messages are stored in the **Mailbox**.

My folders—For organizing your messages into folders.

Mailbox—Connect to your remote mailbox to retrieve your new e-mail messages or view your previously retrieved e-mail messages offline. See 'E-mail', p. 63.

Drafts—Draft messages that have not been sent are stored here.

Sent—The last 20 messages that were sent, excluding messages sent using Bluetooth connectivity, are stored here. To change the number of messages to be saved, see 'Other settings', p. 65.

Outbox—Messages waiting to be sent are temporarily stored here.

Example: Messages are placed in the outbox when your device is outside network coverage. You can also schedule e-mail messages to be sent the next time you connect to your remote mailbox.

Reports—You can request the network to send you a delivery report of the text messages and multimedia messages you have sent (network service). Receiving a delivery report of a multimedia message that has been sent to an e-mail address may not be possible.

Tip! When you have opened any of the default folders, to switch between the folders, move the joystick to the right or left.

To enter and send service requests (also known as USSD commands), such as activation commands for network services, to your service provider, select **Options** > **Service command** in the main view of **Messag.**.

Cell broadcast (network service) allows you to receive messages on various topics, such as weather or traffic conditions from your service provider. For available topics and relevant topic numbers, contact your service provider. In the main view of **Messaging**, select **Options** > **Cell broadcast**. In the main view you can see the status of a topic, a topic number, name, and whether it has been flagged ( ) for follow-up.

Cell broadcast messages cannot be received in UMTS networks. A packet data connection may prevent cell broadcast reception.

Write text

ABC, **abc**, and **Abc** indicate the selected character mode. **123** indicates the number mode.

Traditional text input

The indicator is shown on the top right of the display when you are writing text using traditional text input.

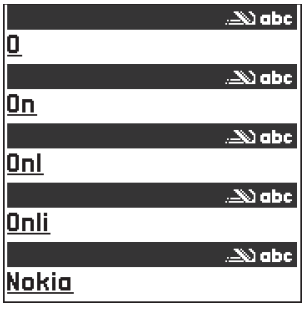
- Press a number key (**1** – **9**) repeatedly until the desired character appears. There are more characters available for a number key than are printed on the key.
- To insert a number, press and hold the number key.
- To switch between letter and number mode, press and hold **#**.
- If the next letter is located on the same key as the present one, wait until the cursor appears (or quickly move the joystick to the right to end the time-out period), and enter the letter.
- To erase a character, press . Press and hold to clear more than one character.
- The most common punctuation marks are available under **1**. Press **1** repeatedly to reach the desired punctuation mark.

  Press * to open a list of special characters. Use the joystick to move through the list, and press **Select** to select a character.
- To insert a space, press **0**. To move the cursor to the next line, press **0** three times.
- To switch between the different character modes, press **#**.

Predictive text input—Predictive text

You can enter any letter with a single key press. Predictive text input is based on a built-in dictionary to which you can also add new words. is shown on the top right of the display when you write text using predictive text input.

1 To activate predictive text input, press , and select **Predictive text on**. This activates predictive text input for all editors in the device.



2 To write the desired word, press the keys **2**–**9**. Press each key only once for one letter.
For example, to write 'Nokia' when the English dictionary is selected, press **6** for N, **6** for o, **5** for k, **4** for i, and **2** for a.

The word suggestion changes after each press.

3 When you finish writing the word correctly, scroll right to confirm it, or press **0** to add a space.

If the word is not correct, press * repeatedly to view each matching word, or press , and select **Predictive text** > **Matches**.

If the **?** character is shown after the word, the word is not in the dictionary. To add a word to the dictionary, select **Spell**, enter the word using traditional text input, and select **OK**. The word is added to the dictionary. When the dictionary is full, a new word replaces the oldest added word.

4 Start writing the next word.

Tip! To set predictive text input on or off, quickly press **#** twice.

Tips on text input

To insert a number in letter mode, press and hold the desired number key.

To switch between the different character modes, press **#**.

To erase a character, press . Press and hold to clear more than one character.

Press and hold * to open a list of special characters.

Tip! To insert several special characters at the same time, press **5** when you select each character.

The most common punctuation marks are available under **1**. Press **1**, then * repeatedly to search for the desired punctuation mark.

Tip! Predictive text input tries to guess which commonly used punctuation mark (**.,?!'**) is needed. The order and availability of the punctuation marks depend on the language of the dictionary.

Press * repeatedly to view the matching words the dictionary has found one by one.

Press , select **Predictive text**, and scroll right to select one of the following options:

Matches—To view a list of words that correspond to your key presses.

Insert word—To add a word to the dictionary by using traditional text input. When the dictionary becomes full, a new word replaces the oldest added word.

Edit word—To edit the word using traditional text input. This is available if the word is active (underlined).

Tip! When you press , the following options appear (depending on the editing mode): **Predictive text** (predictive text input), **Alpha mode** (traditional text input), **Number mode**, **Cut** (if text has been selected), **Copy** (if text has been selected), **Paste** (when text has been cut or copied first), **Insert number**, **Insert symbol**, and **Writing language:** (changes the input language for all editors in the phone).

Write compound words

Write the first half of a compound word; to confirm it, move the joystick right. Write the last part of the compound word. To complete the compound word, press 0 to add a space.

Turn predictive text input off

Press , and select **Predictive text** > **Off** to turn predictive text input off for all editors in the phone.

Copy text to clipboard

1. To select letters and words, press and hold . At the same time, move the joystick left or right. As the selection moves, text is highlighted.
2. To copy the text to the clipboard, while still holding , select **Copy**.
3. To insert the text into a document, press and hold , and select **Paste**, or press once, and select **Paste**
4. To select lines of text, press and hold . At the same time move the joystick up or down.
5. To remove the selected text from the document, press .

Write and send messages

The appearance of a multimedia message may vary, depending on the receiving device.

Copyright protections may prevent some images, music (including ringing tones), and other content from being copied, modified, transferred or forwarded.

Tip! You can start to create a message from any application that has the option **Send**. Select a file (image or text) to be added to the message, and select **Options** > **Send**.

Before you can create a multimedia message or write an e-mail, you must have the correct connection settings in place. See 'Receive MMS and e-mail settings', p. 56 and 'E-mail', p. 63.

The wireless network may limit the size of MMS messages. If the inserted picture exceeds this limit, the device may make it smaller so that it can be sent by MMS.

1 Select **New message** and one of the following:

Text msg.—to send a text message

Multimedia message—to send a multimedia message (MMS)

E-mail—to send an e-mail. If you have not set up your e-mail account, you are prompted to do so.

To start creating the e-mail settings with the mailbox guide, select **Start**.

2 In the **To** field, press the joystick to select recipients or groups from contacts, or enter the recipient's phone number or e-mail address. Press * to add a semicolon (**;**) that separates the recipients. You can also copy and paste the number or address from the clipboard.

Tip! Scroll to a contact, and press the joystick to mark it. You can mark several recipients at a time.

3 In the **Subject** field, enter the subject of a multimedia message or e-mail. To change the fields that are visible, select **Options** > **Address fields**.

4 Move the joystick down to move to the message field and write the message. To use a template, select **Options** > **Insert** or **Insert object** > **Template**.

5 To add a media object to a multimedia message, select **Options** > **Insert object** > **Image**, **Sound clip**, or **Video clip**. When sound is added, the icon is shown.

You cannot send video clips that are saved in the .mp4 format in a multimedia message. To change the format in which recorded videos are saved, see 'Video settings', p. 43.

6 To take a new picture or record sound or video for a multimedia message, select **Insert new** > **Image**, **Sound clip**, or **Video clip**. To insert a new slide to the message, select **Slide**.

To see what the multimedia message looks like, select **Options** > **Preview**.

7 To add an attachment to an e-mail, select **Options** > **Insert** > **Image**, **Sound clip**, **Video clip** or **Note**. E-mail attachments are indicated by .

Tip! To send files other than sounds and notes as attachments, open the appropriate application, and select **Send** > **Via e-mail**, if available.

8 To send the message, select **Options** > **Send**, or press .

Your device supports the sending of text messages beyond the character limit for a single message. Longer messages will be sent as a series of two or more messages. Your service provider may charge accordingly. Characters that use accents or other marks, and characters from some language options like Chinese, take up more space limiting the number of characters that can be sent in a single message. In the navigation bar, you can see the message length indicator counting backwards. For example, 10 (2) means that you can still add 10 characters for the text to be sent as two messages.

E-mail messages are automatically placed in **Outbox** before sending. If sending does not succeed, the e-mail is left in the **Outbox** with the **Failed** status.

Tip! In **Messaging** you can also create presentations and send them in a multimedia message. In the multimedia message editor view, select **Options** > **Create presentation** (shown only if **MMS creation mode** is set to **Guided** or **Free**). See 'Multimedia messages', p. 62.

Tip! When you send a multimedia message to an e-mail address or a device that supports the receiving of large images, use the larger image size. If you are not sure of the receiving device, or the network does not support sending large files, it is recommended that you use a smaller image size or a sound clip that is no longer than 15 seconds. To change the setting, select **Options** > **Settings** > **Multimedia message** > **Image size** in the **Messaging** main view.

Note: The indicators showing that the message has been sent indicate that the message has been sent by your device to the message center number programmed into your device. This is not an indication that the message has been received at the intended destination. For more details about messaging services, check with your service provider.

Receive MMS and e-mail settings

You may receive the settings in a text message from your network operator or service provider. See 'Data and settings', p. 57.

For availability of and subscription to data services, contact your network operator or service provider. Follow the instructions given by your service provider.

Enter the MMS settings manually:

1 Press , and select **Tools** > **Settings** > **Connection** > **Access points**, and define the settings for a multimedia messaging access point. See 'Connection', p. 97.

2 Press , and select **Messaging** > **Options** > **Settings** > **Multim. msg.** > **Access point in use** and the access point you created as the preferred connection. See also 'Multimedia messages', p. 62.

Before you can send, receive, retrieve, reply to, and forward e-mail, you must do the following:

- Configure an Internet access point (IAP) correctly. See 'Connection', p. 97.
- Define your e-mail settings correctly. If you select **Mailbox** in the **Messag.** main view and have not set up your e-mail account, you are prompted to do so. To start creating the e-mail settings with the mailbox guide, select **Start**. See 'E-mail', p. 63.

  You must have a separate e-mail account. Follow the instructions given by your remote mailbox and Internet service provider (ISP).

Inbox—receive messages

Inbox folder has the following indicators:

—Unread text message.

—Unread multimedia message.

—Data received through Bluetooth connectivity.

When you receive a message, and the text **1 new message** are shown in the standby mode. To open the message, select **Show**. To open a message in **Inbox**, scroll to it, and press the joystick.

To reply to a received message, select **Options** > **Reply**.

Multimedia messages

Important: Excercise caution when opening messages. Multimedia message objects may contain malicious software or otherwise be harmful to your device or PC.

You may receive a notification that a multimedia message is waiting in the multimedia message center. To start a packet data connection to retrieve the message to your device, select **Options** > **Retrieve**.

When you open a multimedia message ( ), you may see an image and a message. is shown if sound is included, or if video is included. To play the sound or the video, select the indicator.

To see what kinds of media objects have been included in the multimedia message, open the message, and select **Options** > **Objects**.

If the message includes a multimedia presentation, is displayed. To play the presentation, select the indicator.

Data and settings

Your device can receive many kinds of messages that contain data ( ):

Configuration message—You may receive settings from your network operator, service provider, or company information management department in a configuration message. To save the settings, select **Options** > **Save all**.

Business card—To save the information to **Contacts**, select **Options** > **Save business card**. Certificates or sound files attached to the business card are not saved.

Ringing tone—To save the ringing tone, select **Options** > **Save**.

Operat. logo—To have the logo shown in the standby mode instead of the network operator's own identification, select **Options** > **Save**.

Calen. entry—To save the invitation, select **Options** > **Save to Calendar**.

Web message—To save the bookmark to the bookmarks list in Web, select **Options** > **Add to bookmarks**. If the message contains both access point settings and bookmarks, to save the data, select **Options** > **Save all**.

E-mail notif.—The note tells you how many new e-mail messages you have in your remote mailbox. An extended notification may list more detailed information.

Web service messages

Web service messages ( ) are notifications (for example, news headlines) and may contain a text message or a link. For availability and subscription, contact your service provider.

My folders

In **My folders**, you can organize your messages into folders, create new folders, and rename and delete folders.

Tip! You can use texts in the templates folder to avoid rewriting messages that you send often.

Mailbox

If you select **Mailbox** and have not set up your e-mail account, you are prompted to do so. To start creating the e-mail settings with the mailbox guide, select **Start**. See also 'E-mail', p. 63.

When you create a new mailbox, the name you give to the mailbox replaces **Mailbox** in the **Messaging** main view. You can have up to six mailboxes.

Open the mailbox

When you open the mailbox, the device asks if you want to connect to the mailbox (**Connect to mailbox?**).

To connect to your mailbox and retrieve new e-mail headings or messages, select **Yes**. When you view messages online, you are continuously connected to a remote mailbox using a data connection. See also 'Connection', p. 97.

To view previously retrieved e-mail messages offline, select **No**. When you view e-mail messages offline, your device is not connected to the remote mailbox.

Retrieve e-mail messages

If you are offline, select **Options** > **Connect** to start a connection to a remote mailbox.

Important: Excercise caution when opening messages. E-mail messages may contain malicious software or otherwise be harmful to your device or PC.

1 When you have an open connection to a remote mailbox, select **Options** > **Retrieve e-mail** and one of the following:

New—to retrieve all new e-mail messages

Selected—to retrieve only the e-mail messages that have been marked

All—to retrieve all messages from the mailbox

To stop retrieving messages, select **Cancel**.

2 After you retrieve the e-mail messages, you can continue viewing them online, or select **Options** > **Disconnect** to close the connection and view the e-mail messages offline.

E-mail status indicators:

The new e-mail (offline or online mode) has not been retrieved to your device.

The new e-mail has been retrieved to your device.

The e-mail message has been read, and it has not been retrieved to your device.

The e-mail message has been read.

The e-mail heading has been read and the message content has been deleted from the device.

3 To open an e-mail message, press the joystick. If the e-mail message has not been retrieved and you are offline, you are asked if you want to retrieve this message from the mailbox.

To view e-mail attachments, open a message that has the attachment indicator, and select **Options** > **Attachments**. If the attachment has a dimmed indicator, it has not been retrieved to the device; select **Options** > **Retrieve**. In the **Attachments** view, you can retrieve, open, save, or remove attachments. You can also send attachments using Bluetooth connectivity.

Tip! If your mailbox uses the IMAP4 protocol, you can define how many messages to retrieve, and whether to retrieve the attachments. With the POP3 protocol, the options are **Headers only**, **Partially (kB)**, or **Msgs. & attachs.**.

Retrieve e-mail messages automatically

To retrieve messages automatically, select **Options** > **E-mail settings** > **Automatic retrieval** > **Header retrieval** > **Always enabled** or **Only in home net.**, and define when and how often the messages are retrieved.

Retrieving e-mail messages automatically may increase your call costs due to the data traffic.

Delete e-mail messages

To delete the contents of an e-mail message from the device while still retaining it in the remote mailbox, select **Options** > **Delete**. In **Delete msg. from:**, select **Phone only**.

The device mirrors the e-mail headings in the remote mailbox. Although you delete the message content, the e-mail heading stays in your device. If you want to remove the heading as well, you must first delete the e-mail message from your remote mailbox, then make a connection from your device to the remote mailbox again to update the status.

To delete an e-mail from the device and the remote mailbox, select **Options** > **Delete**. In **Delete msg. from:**, select **Phone and server**.

If you are offline, the e-mail is deleted first from your device. During the next connection to the remote mailbox, it is automatically deleted from the remote mailbox. If you are using the POP3 protocol, messages marked to be deleted are removed only after you have closed the connection to the remote mailbox.

To cancel deleting an e-mail from the device and server, scroll to an e-mail that has been marked to be deleted during the next connection ( ), and select **Options** > **Undelete**.

Disconnect from the mailbox

When you are online, select **Options** > **Disconnect** to end the packet data connection to the remote mailbox.

Tip! You can also leave your mailbox connection open and the new e-mails (**Headers only** as default) will be retrieved from the remote mailbox to your device automatically (only if the IMAP IDLE function is supported by your server). To leave the messaging application open in the background, press twice. Leaving the connection open may increase your call costs due to the data traffic.

View e-mail messages when offline

When you open **Mailbox** the next time and you want to view and read the e-mail messages offline, answer **No** to the **Connect to mailbox?** query. You can read the previously retrieved e-mail headings, the retrieved e-mail messages, or both. You can also write new, reply to, or forward e-mail to be sent the next time you connect to the mailbox.

Outbox—messages waiting to be sent

Outbox is a temporary storage place for messages waiting to be sent.

Status of the messages in **Outbox** are as follows:

Sending—A connection is being made, and the message is being sent.

Waiting/**Queued**—The message will be sent when previous messages of a similar type have been sent.

Resend at %U (time)—The device will try to send the message again after a time-out period. Select **Send** to restart the sending immediately.

Deferred—You can set documents to be on hold while they are in **Outbox**. Scroll to a message that is being sent, and select **Options** > **Defer sending**.

Failed—The maximum number of sending attempts has been reached. Sending has failed. If you were trying to send a text message, open the message, and check that the Sending options are correct.

Example: Messages are placed in the outbox, for example, when your device is outside network coverage. You can also schedule e-mail messages to be sent the next time you connect to your remote mailbox.

View messages on a SIM card

Before you can view SIM messages, you must copy them to a folder in your device.

1. In the **Messag.** main view, select **Options** > **SIM messages**.
2. Select **Options** > **Mark/Unmark** > **Mark** or **Mark all** to mark messages.
3. Select **Options** > **Copy**. A list of folders opens.
4. To start copying, select a folder and **OK**. To view the messages, open the folder.

Messaging settings

Fill in all fields marked with **Must be defined** or with a red asterisk. Follow the instructions given by your service provider. You may also obtain settings from your service provider in a configuration message.

Some or all message centers, access points or other settings may be preset for your device by your service provider or network operator, and you may not be able to change, create, edit, or remove them.

Text messages

Press , and select **Messag.** > **Options** > **Settings** > **Text msg.** and from the following:

Message centers—View a list of all text message centers that have been defined.

Msg. center in use—Select which message center is used to deliver text messages.

SMS e-mail—To send and receive text messages to and from e-mail addresses, enter the e-mail gateway and e-mail message center.

Character encoding— To use character conversion to another encoding system when available, select **Reduced support**.

Receive report—Select whether the network sends delivery reports on your messages (network service).

Message validity—Select how long the message center resends your message if the first attempt fails (network service). If the recipient cannot be reached within the validity period, the message is deleted from the message center.

Message sent as—Change this option only if you are sure that your message center is able to convert text messages into these other formats. Contact your network operator.

Preferred conn.—Select the connection used for sending text messages: GSM network or packet data, if supported by the network. See 'Connection', p. 97.

Reply via same ctr.—Select whether you want the reply message to be sent using the same text message center number (network service).

Add new and edit text message centers

Note that if you change your SIM card, the list of available text message centers is updated according to the centers on the new SIM card. Message centers may also be preset in your device by your service provider, and you may not be able to create, edit, or delete them.

1. Select **Message centers** > **Options** > **New msg. center** or **Edit**.
2. Press the joystick, write a name for the message center, and select **OK**.
3. Scroll down, press the joystick, and write the number of the text message center. You receive the number from your service provider.
4. Select **OK**.
5. To use the new settings, go back to the settings view. Scroll to **Msg. center in use**, and select the new message center.

Multimedia messages

Press , and select **Messag.** > **Options** > **Settings** > **Multimedia message** and from the following:

Image size—Define the size of the image in a multimedia message: **Original** (shown only when the **MMS creation mode** is set to **Guided** or **Free**), **Small**, or **Large**.

MMS creation mode—If you select **Guided**, the device informs you if you try to send a message that may not be supported by the recipient. Select **Restricted**; the device prevents you from sending messages that may not be supported.

Access point in use—Select which access point is used as the preferred connection for the multimedia message center.

Multimedia retrieval—Select how you want to receive multimedia messages. To receive multimedia messages automatically in your home network, select **Aut. in home network**. Outside your home network, you receive a notification that there is multimedia message that you can retrieve in the multimedia message center.

When you are outside your home network, sending and receiving multimedia messages may cost you more.

If you select **Multimedia retrieval** > **Always automatic**, your device automatically makes an active packet data connection to retrieve the message both in and outside your home network.

Allow anon. msgs.—Select whether you want to decline messages from an anonymous sender.

Receive ads—Define whether you want to receive multimedia message advertisements.

Receive report—Select whether you want the status of the sent message to be shown in the log (network service). Receiving a delivery report of a multimedia message that has been sent to an e-mail address may not be possible.

Deny report sending—Select whether you want to deny your device sending delivery reports of received multimedia messages.

Message validity—Select how long the message center resends your message if the first attempt fails (network service). If the recipient cannot be reached within the validity period, the message is deleted from the message center.

Tip! You may also obtain the multimedia and e-mail settings from your service provider in a configuration message. Contact your service provider for more information. See 'Data and settings', p. 57.

E-mail

Press , and select **Messag.** > **Options** > **Settings** > **E-mail** and from the following:

Mailboxes—Select a mailbox to change the following settings: **Connection settings**, **User settings**, **Retrieval settings**, and **Automatic retrieval**.

Mailbox in use— Select which mailbox you want to use for sending e-mail.

To create a new mailbox, select **Options** > **New mailbox** in the mailbox main view.

Connection settings

To edit the settings for the e-mail you receive, select **Incoming e-mail** and from the following:

User name—Enter your user name, given to you by your service provider.

Password—Enter your password. If you leave this field blank, you are prompted for the password when you try to connect to your remote mailbox.

Incoming mail serv.—Enter the IP address or host name of the mail server that receives your e-mail.

Access point in use—Select an Internet access point (IAP). See 'Connection', p. 97.

Mailbox name—Enter a name for the mailbox.

Mailbox type—Defines the e-mail protocol that your remote mailbox service provider recommends. The options are **POP3** and **IMAP4**. This setting can be selected only once and cannot be changed if you have saved or exited from the mailbox settings. If you use the POP3 protocol,

e-mail messages are not updated automatically when you are online. To see the latest e-mail messages, you must disconnect and make a new connection to your mailbox.

Security (ports)—Select the security option used to secure the connection to the remote mailbox.

Port—Define a port for the connection.

APOP secure login (for POP3 only)—Use with the POP3 protocol to encrypt the sending of passwords to the remote e-mail server while connecting to the mailbox.

To edit the settings for the e-mail you send, select **Outgoing e-mail** and from the following:

My e-mail address—Enter the e-mail address given to you by your service provider. Replies to your messages are sent to this address.

Outgoing mail serv.—Enter the IP address or host name of the mail server that sends your e-mail. You may only be able to use the outgoing server of your network operator. Contact your service provider for more information.

Access point in use—Select an Internet access point (IAP). See 'Connection', p. 97.

The settings for **User name**, **Password**, **Security (ports)**, and **Port** are similar to the ones in **Incoming e-mail**.

User settings

My name—Enter your own name. Your name replaces your e-mail address in the recipient's device when the recipient's device supports this function.

Send message—Define how e-mail is sent from your device. Select **Immediately** for the device to connect to the mailbox when you select **Send message**. If you select **During next conn.**, e-mail is sent when the connection to the remote mailbox is available.

Send copy to self—Select whether you want to save a copy of the e-mail to your remote mailbox and to the address defined in **My e-mail address**.

Include signature—Select whether you want to attach a signature to your e-mail messages.

New e-mail alerts—Select whether you want to receive the new e-mail indications, a tone and a note, when new mail is received to the mailbox.

Retrieval settings

E-mail to retrieve—Define which parts of the e-mail are retrieved: **Headers only**, **Partially (kB)**, **Messages** (for IMAP4 only), or **Msgs. & attachs.**.

Retrieval amount—Define how many new e-mail messages are retrieved to the mailbox.

IMAP4 folder path (for IMAP4 only)—Define the folder path for folders to be subscribed.

Folder subscriptions (for IMAP4 only)—Subscribe to other folders in the remote mailbox and retrieve content from those folders.

Automatic retrieval

Header retrieval—Select whether you want the device to retrieve new e-mail automatically. You can define when, and how often, the messages are retrieved.

Retrieving e-mail messages automatically may increase your call costs due to the data traffic.

Web service messages

Press , and select **Messaging** > **Options** > **Settings** > **Service message**. Select whether you want to receive service messages. If you want to set the device to automatically activate the browser and start a network connection to retrieve content when you receive a service message, select **Download messages** > **Automatically**.

Cell broadcast

Check the available topics and related topic numbers with your service provider. Press , and select **Messaging** > **Options** > **Settings** > **Cell broadcast** and from the following:

Reception—Select whether you want to receive cell broadcast messages.

Language—Select the languages in which you want to receive messages: **All**, **Selected**, or **Other**.

Topic detection—Select whether the device automatically searches for new topic numbers, and saves the new numbers without a name to the topic list.

Other settings

Press , and select **Messaging** > **Options** > **Settings** > **Other** and from the following:

Save sent messages—Select whether you want to save a copy of the text messages, multimedia messages, or e-mail that you send to the **Sent** folder.

No. of saved msgs.—Define how many sent messages are saved to the **Sent** folder at a time. The default limit is 20 messages. When the limit is reached, the oldest message is deleted.

Memory in use—Select the memory where you want to save your messages: **Phone memory** or **Hard drive**.

My own

Web

Press , and select **My Own** > **Web** to open a supplementary web browser optimized for browsing web sites that are not originally formatted for mobile browsing.

Web preserves the original page layout.

To easily scroll within a page, scroll to any direction on a page. A smaller image of the page is displayed with a red frame that indicates the currently shown area.

To view the thumbnail image of the current page you can also select **Options** > **Tools** > **Page overview**. Move the red frame with the joystick and press the joystick to return to normal view.

To view recently visited pages as thumbnails, select **Options** > **Navigation options**.

To block pop-up windows, select **Options** > **Settings** > **Block pop-up windows**.

To search for a text in real time, select **Options** > **Find** and start entering the search string. The matching string is displayed as you enter the text.

Web feeds: You can subscribe to news and blog feeds hosted by many web sites. The Web feeds feature of the browser will also alert the user when feed links are available for a web site.

For information on other features and options in Web, see 'Services', p. 81.

Notes

Press , and select **My Own** > **Notes** to write notes. You can send notes to other compatible devices and save plain text files (.txt format) that you receive to **Notes**.

Go to—add shortcuts

To store shortcuts (links to your favorite images, notes, bookmarks, and so on), press , and select **My Own** > **Go to**. Shortcuts are added only from the individual applications. Not all applications have this function.

1 Select an item from an application to which you want to add a shortcut.

2 Select **Add to 'Go to'**. A shortcut in **Go to** is automatically updated if you move the item to which it is pointing (for example, from one folder to another). To change the identifier in the lower left corner of the shortcut icon, select **Options** > **Shortcut icon**.

Delete a shortcut

Select the shortcut, and press C. The default shortcuts **Notes**, **Calendar**, and **Inbox** cannot be deleted.

When you remove an application or a document which has a shortcut in **Go to**, the shortcut icon of the removed item is dimmed in the **Go to** view. The shortcut can be deleted next time you start **Go to**.

IM—instant messaging (chat)

Press , and select **My Own** > **IM**.

Instant messaging (network service) allows you to converse with other people using instant messages and join discussion forums (IM groups) with specific topics. Various service providers maintain IM servers that you can access after you subscribe to an IM service. Service providers may differ in their support of features.

Select **Conversations** to start or continue a conversation with an IM user; **IM contacts** to create, edit, or view the online status of your IM contacts; **IM groups** to start or continue a group conversation with multiple IM users; or **Record. chats** to view a previous instant messaging session that you saved.

Check the availability of services, pricing, instructions, and fees with your network operator or service provider.

Receive IM settings

You must save the settings to access the service that you want to use. You may receive the settings in a special text message from the network operator or service provider that offers the IM service. See 'Data and settings', p. 57. You can also enter the settings manually. See 'IM server settings', p. 71.

Connect to an IM server

1 To connect to the IM server in use, open **IM**, and select **Options** > **Login**. To change the IM server in use and save new IM servers, see 'IM server settings', p. 71.

2 Enter your user ID and password, and press the joystick to log in. You obtain the user ID and password for the IM server from your service provider.

3 To log out, select **Options** > **Logout**.

Modify your IM settings

Select **Options** > **Settings** > **IM settings** and from the following:

Use screen name (shown only if IM groups are supported by the server)—To enter a nickname, select **Yes**.

Allow messages from—To allow messages from all, select **All**.

Allow invitations from—To allow invitations only from your IM contacts, select **IM contacts only**. IM invitations are sent by IM contacts who want you to join their groups.

Msg. scrolling speed—Select the speed at which new messages are displayed.

Sort IM contacts—Select how your IM contacts are sorted: **Alphabetically** or **By online status**.

Availabil. reloading—To select how to update information about whether your IM contacts are online or offline, select **Automatic** or **Manual**.

Offline contacts—Select whether IM contacts with an offline status are shown in the IM contacts list.

Search for IM groups and users

To search for groups, in the **IM groups** view, select **Options** > **Search**. You can search by **Group name**, **Topic**, and **Members** (user ID).

To search for users, in the **IM contacts** view, select **Options** > **New IM contact** > **Search from server**. You can search by **User's name**, **User ID**, **Phone number**, and **E-mail address**.

Join IM groups

The **IM groups** view shows a list of the IM groups that you have saved or are currently joined to.

To join a saved IM group, press the joystick.

To join an IM group that is not on the list, but for which you know the group ID, select **Options** > **Join new group**.

To leave the IM group, select **Options** > **Leave IM group**.

Chat

After you join an IM group, you can view the messages that are exchanged there, and send your own messages.

To send a message, write the message in the message editor field, and press the joystick.

To send a private message to a participant, select **Options** > **Send private msg.**.

To reply to a private message sent to you, select the message and **Options** > **Reply**.

To invite IM contacts who are online to join the IM group, select **Options** > **Send invitation**.

To prevent receiving messages from certain participants, select **Options** > **Blocking options** and the desired option.

Record chats

To record the messages that are exchanged during a conversation or while you are joined in a IM group, select **Options** > **Record chat**. To stop recording, select **Options** > **Stop recording**. To view the recorded chats, in the main view, select **Record. chats**.

View and start conversations

The **Conversations** view shows a list of the individual conversation participants that you have an ongoing conversation with. Ongoing conversations are automatically closed when you exit **IM**.

To view a conversation, scroll to a participant, and press the joystick.

To continue the conversation, write your message, and press the joystick.

To return to the conversations list without closing the conversation, select **Back**. To close the conversation, select **Options** > **End conversation**.

To start a new conversation, select **Options** > **New conversation**.

To save a conversation participant to your IM contacts, select **Options** > **Add to IM contacts**.

To send automatic replies to incoming messages, select **Options** > **Set auto reply on**. You can still receive messages.

IM contacts

In the **IM contacts** view, you can retrieve IM contact lists from the server, or add a new IM contact to a contact list. When you log in to the server, the previously used IM contact list is retrieved from the server automatically.

To create a new contact, select **Options** > **New IM contact** > **Enter manually**. Fill in the **Nickname** and **User ID** fields, and select **Done**.

To move a contact from a list on the server to the retrieved list, select **Options** > **New IM contact** > **Move from other list**.

To change the IM contact list, select **Options** > **Change contact list**.

Select from the following options:

Open conversation—To start a new conversation or continue an ongoing conversation with the contact.

Switch tracking on—To have the device notify you every time the IM contact goes online or offline.

Belongs to groups—To see which groups the IM contact has joined.

Reload users' availab.—To update information about whether the contact is online or offline. The online status is shown by an indicator next to the contact name. This option is not available if you have set the **Availabil. reloading** to **Automatic** in **IM settings**.

Manage IM groups

Select **IM groups** view to see a list of the IM groups that you have either saved or are currently joined to.

Scroll to a group, select **Options** > **Group** and one of the following:

Save—To save an unsaved group that you are currently joined to.

View participants—To see who is currently joined to the group.

Details—To see the group ID, topic, participants, administrators of the group (shown only if you have the editing rights), the list of blocked participants (shown only if you have the editing rights), and whether private conversations are allowed in the group.

Settings—To view and edit the IM group settings. See 'Create a new IM group', p 70.

Create a new IM group

Select **IM groups** > **Options** > **Create new group**. Enter the settings for the group.

You can edit the settings for an IM group if you have administrator rights to the group. The user who creates a group is automatically given administrator rights to it.

Group name, **Group topic**, and **Welcome note**—To add details that the participants see when they join the group.

Group size—To define the maximum number of members allowed to join the group.

Allow search—To define if others can find the IM group by searching.

Editing rights—To define the IM group participants to whom you want to give rights to invite contacts to join the IM group and edit the group settings.

Group members—See 'Add and remove group members', p. 71.

Banned list—Enter the participants that are not allowed to join the IM group.

Allow private msgs.—To allow messaging between selected participants only.

Group ID—The group ID is created automatically and cannot be changed.

Add and remove group members

To add members to a group, select **IM groups**, scroll to an IM group, and select **Options** > **Group** > **Settings** > **Group members** > **Selected only** or **All**.

To remove a member from the IM group, scroll to the member, and select **Options** > **Remove**. To remove all members, select **Options** > **Remove all**.

IM server settings

Select **Options** > **Settings** > **Server settings**. You may receive the settings in a special text message from the network operator or service provider that offers the chat service. You obtain the user ID and password from your service provider when you subscribe for the service. If you do not know your user ID or password, contact your service provider.

Servers—View a list of all defined IM servers.

Default server—Change the IM server to which you want to connect.

IM login type—To log in automatically when you start **IM**, select **On app. start-up**.

To add a new server to your list of IM servers, select **Servers** > **Options** > **New server**. Enter the following settings:

Server name—the name for the IM server

Access point in use—the access point you want to use for the server

Web address—the URL address of the IM server

User ID—your user ID

Password—your login password

Push to talk (network service)

Push to talk (PTT) is a real-time voice over IP service implemented over a GSM/GPRS network. Push to talk provides direct voice communication connected with the push of a key. Use push to talk to have a conversation with one person or with a group of people.

Before you can use push to talk, you must define the push-to-talk access point and push-to-talk settings. You may receive the settings in a special text message from the service provider that offers the push-to-talk service.

Press , and select **My Own** > **Push to talk**.

In push-to-talk communication, one person talks while the others listen through the built-in loudspeaker. Speakers take turns responding to each other. Because only one group member can talk at any time, the maximum duration of a speech turn is limited. The maximum duration is usually set to 30 seconds. For details of the speech turn

duration for your network, contact your network operator or service provider.

Warning: Be sure to hold the device in front of you and not to your ear. The loudspeaker is used to listen. Continuous exposure to high volume may damage your hearing.

Phone calls always take priority over push-to-talk activities.

Define a push-to-talk access point

Many service providers require you to use an Internet access point (IAP) for your default access point. Other service providers allow you to use a WAP access point. Contact your service provider for more information.

If you do not have a WAP connection, you may need to contact your service provider for assistance with the first-time connection or visit **www.nokia.com/phonesettings**.

Push-to-talk settings

Press , and select **My Own** > **Push to talk** > **Options** > **PTT settings**. Enter the following information:

User settings:

Incoming PTT calls—Select **Notify** if you want to see a notification of incoming calls. Select **Auto-accept** if you want PTT calls to be answered automatically. Select **Not allowed** if you want PTT calls to be declined automatically.

PTT call alert tone—Select **Set by profile** if you want the incoming call alert setting for push to talk to follow your profile settings. For example, if your profile is silent, push to talk is set to do not disturb (DND) and, except for callback requests, you are not available to others using push to talk.

Callback request tone—Select a tone for callback requests.

Application start-up—Select if you want to log in to the Push-to-talk service when you switch on your device.

Default nickname—Enter your default nickname (20 characters maximum) that is displayed to other users. Your service provider may have disabled this option in your device. For more information, contact your network operator or service provider.

Show my PTT address—Select **In 1 to 1 calls**, **In group calls**, **In all calls**, or **Never**. Your service provider may have disabled this option in your device. For more information, contact your network operator or service provider.

Show my login status—Select **Yes** whether you want your status to be shown or **No** if you want your status to be hidden.

Connection:

Domain—Enter the domain name obtained from your service provider.

Access point name—Enter the push-to-talk access point name. The access point name is needed to establish a connection to the GSM/GPRS network.

Server address—Enter the IP address or domain name of the push-to-talk server obtained from your service provider.

User name—Enter your user name obtained from your service provider.

PTT password—Enter a password, if required, to make a data connection. The password is usually provided by the service provider and is often case-sensitive.

Log in to push to talk

Press , and select **My Own** > **Push to talk**. Push to talk automatically logs in to the service when started.

When log in is successful, push to talk automatically connects to groups that were active when the application was last closed. If the connection is lost, the device automatically tries to log back in until you choose to exit push to talk.

Exit push to talk

Select **Options** > **Exit**. **Switch Push to talk off after exiting application?** is displayed. Select **Yes** to keep the application active on the background or **No** to log out and close the service.

If several applications are open and you want to switch from one application to another, press and hold .

Make a one-to-one call

Select **Options** > **PTT contacts**.

Select the contact you want to talk to from the list and select **Options** > **Talk 1 to 1**.

Press and hold the entire time you are talking. When you have finished talking, release .

You must end the one-to-one call before you can create a new call. Select **Disconnect** or press .

Tip! Remember to hold the device in front of you during a push-to-talk call so you can see the display. Speak towards the microphone, and do not cover the loudspeaker with your hands.

Answer a one-to-one call

Press ⌐ to start a one-to-one call or ¬ to dismiss the call.

Tip! You can make a one-to-one or group call also from Contacts. Select **Options** > **Talk 1 to 1** or **Make PTT group call**.

Make a group call

To call a group, select **Options** > **PTT contacts**, select **Options** > **Make PTT group call**, mark the contacts you want to call, and press ⌐.

Subscribe login status of others

To subscribe or unsubscribe to the push-to-talk login status of others, select **Options** > **PTT contacts**, select the contact, select **Options** > **Show login status** or **Hide login status**.

Send a callback request

In **PTT contacts**, scroll to the desired name, and select **Options** > **Send callback request**.

Respond to a callback request

When someone sends you a callback request, **1 new callback request** is displayed in standby mode. Press **Show** to open **Callback inbox**. Select a contact and select **Options** > **Talk 1 to 1** to start a one-to-one call.

Channels

You can join public preconfigured channels. Public channels are open to anyone who knows the URL for the channel.

When you are connected to a channel and talk, all members joined to the channel hear you talking. Up to five channels can be active at the same time. When more than one channel is active, use **Swap** to change the channel you want to talk to.

Select **Options** > **PTT channels**.

To connect to a public preconfigured channel for the first time, select **Options** > **New channel** > **Add existing**. When you join a preconfigured channel, you must fill in an URL for the channel.

Create your own channels

To create a new preconfigured channel select **Options** > **New channel** > **Create new**.

You can create your own public channels, choose your own channel name, and invite members. Those members can invite more members to the public channel.

You may also set up private channels. Only users invited by the host are allowed to join and use private channels.

For each channel, define the following: **Channel name**, **Channel privacy**, **Nickname in channel**, and **Channel thumbnail** (optional).

When you have successfully created a channel, you are asked if you want to send channel invitations. Channel invitations are text messages.

Talking to a channel

To talk to a channel after you log in to the push-to-talk service, press . You hear a tone indicating that access is granted.

Continue to press and hold the entire time you are talking. When you finish talking, release .

If you try to respond to a channel by pressing while another member is talking, **Wait** is displayed. Release , wait for the other person to finish talking, and press again when the speaker has finished. Alternatively, keep pressed and wait for **Talk** to be displayed.

When you are talking in a channel, the first person to press when someone stops talking talks next.

When you have finished the push-to-talk call, select **Disconnect** or press .

To view currently active members of a channel when you have an active call to the channel, select **Options** > **Active members**.

You can also invite new members to a channel when you are connected to it, select **Options** > **Send invitation** to open the invitation view. You can only invite new members when you are the host of a private channel, or when the channel is a public channel. Channel invitations are text messages.

Respond to a channel invitation

To save a received channel invitation, select **Options** > **Save PTT channel**. The channel is added to your PTT contacts, channels view.

After you save the channel invitation, you are asked if you want to connect to the channel. Select **Yes** to open the push-to-talk sessions view. Your device logs in to the service, if you are not logged in already.

If you reject or clear the invitation, the invitation is stored in your messaging inbox. To join the channel later, open the invitation message, and accept.

View the push-to-talk log

Press , and select **My Own** > **Push to talk** > **Options** > **PTT log**. The log contains the following folders: **Missed calls**, **Received PTT calls**, and **Created calls**.

You can initiate a one-to-one call from **PTT log** by pressing .

Log

Recent calls

To monitor the phone numbers of missed, received, and dialed calls, press , and select **My Own** > **Log** > **Recent calls**. The device logs missed and received calls only if the network supports these functions, the device is switched on, and within the network service area.

Tip! When you see a note in the standby mode about missed calls, select **Show** to access the list of missed calls. To call back, scroll to a name or number, and press .

To clear all recent call lists, select **Options** > **Clear recent calls** in the recent calls main view. To clear one of the call logs, open the log you want to erase, and select **Options** > **Clear list**. To clear an individual event, open a log, scroll to the event, and press .

Call timers

To monitor the approximate duration of your incoming and outgoing calls, press , and select **Log** > **Call timers**.

To set the display of the call duration while a call is active, select **My Own** > **Log** > **Options** > **Settings** > **Show call duration**. Select **Yes** or **No**.

Note: The actual time invoiced for calls by your service provider may vary, depending on network features, rounding off for billing, and so forth.

To clear call duration timers, select **Options** > **Clear timers**. For this you need the lock code, see 'Security', 'Phone and SIM', p. 101.

Packet data

To check the amount of data sent and received during packet data connections, press , and select **My Own** > **Log** > **Packet data**. For example, you may be charged for your packet data connections by the amount of data sent and received.

Monitor all communication events

Icons in **Log**:

Incoming

Outgoing

Missed communication events

To monitor all voice calls, text messages, or data connections logged by the device, press , select **My Own** > **Log**, and press the joystick to the right to open the general log. For each communication event, you can see the sender or recipient name, phone number, name of the service provider, or access point. You can filter

the general log to view just one type of event and create new contact cards based on the log information.

Tip! If you want to see the duration of a voice call on the main display during an active call, select **Options** > **Settings** > **Show call duration** > **Yes**.

Tip! To view a list of sent messages, press , and select **Messaging** > **Sent**.

Subevents, such as a text message sent in more than one part and packet data connections, are logged as one communication event. Connections to your mailbox, multimedia messaging center, or Web pages are shown as packet data connections.

To filter the log, select **Options** > **Filter** and a filter.

To erase the contents of the log, recent calls log, and messaging delivery reports permanently, select **Options** > **Clear log**. Select **Yes** to confirm.

To set the **Log duration**, select **Options** > **Settings** > **Log duration**. The log events remain in the phone memory for a set number of days after which they are automatically erased to free memory. If you select **No log**, all the log contents, recent calls log, and messaging delivery reports are permanently deleted.

To view the details of a communication event, in the general log view scroll to an event, and press the joystick.

Tip! In the details view you can copy a phone number to the clipboard, and paste it to a text message, for example. Select **Options** > **Copy Number**.

Packet data counter and connection timer: To view how much data, measured in kilobytes, is transferred and how long a certain packet data connection lasts, scroll to an incoming or outgoing event indicated by **Pack.**, and select **Options** > **View details**.

Calendar

Tip! Regularly back up the phone information with Nokia PC Suite. You can restore the information, such as calendar entries, to the phone later.

Create calendar entries

Shortcut: In the day, week, or month calendar view, press any key (**1** – **0**). An appointment entry opens, and the characters you enter are added to the **Subject** field. In the to-do view, a to-do note entry opens.

1 Press , and select **My own** > **Calendar** > **Options** > **New entry** and one of the following:

Meeting—To remind you of an appointment that has a specific date and time.

Memo—To write a general entry for a day.

Anniversary—To remind you of birthdays or special dates. Anniversary entries are repeated every year.

To-do—to remind you of a task that needs doing by a specific date

2 Fill in the fields.

Alarm (meetings and anniversaries)—Select **On**, and scroll down to fill in the **Alarm time** and **Alarm date** fields. in the day view indicates an alarm.

Repeat—Scroll right to change the entry to be repeating (is shown in the day view).

Repeat until—You can set an end date for the repeated entry.

Synchronization:

Private—After synchronisation the calendar entry can be seen only by you and it will not be shown to others with online access to view the calendar.

Public—The calendar entry is shown to others who have access to view your calendar online.

None—The calendar entry will not be copied to your PC when you synchronize.

3 To save the entry, select **Done**.

When the calendar alarms for a note, select **Silence** to turn off the calendar alarm tone. The reminder text stays on the screen. To end the calendar alarm, select **Stop**. To set the alarm to snooze, select **Snooze**.

To send a calendar note to a compatible phone, select **Options** > **Send** > **Via text message**, **Via multimedia**, or **Via Bluetooth**.

Tip! You can move calendar and to-do data from many different Nokia phones to your phone or synchronize your calendar and to-do to a compatible PC using Nokia PC Suite. See the CD-ROM supplied with the phone.

Set a calendar alarm

You can set an alarm to remind you of meetings and anniversaries.

1. Open an entry in which you want to set an alarm, and select **Alarm** > **On**.
2. Set the **Alarm time** and **Alarm date**.
3. Scroll down to **Repeat**, and scroll right to select how often you want the alarm to be repeated.
4. Select **Done**.

To delete a calendar alarm, open the entry in which you want to delete an alarm, and select **Alarm** > **Off**.

Calendar views

Tip! Select **Options** > **Settings** to change the starting day of the week or the view that is shown when you open the calendar.

To open Month view or To-do view, select **Options** > **Month view** / **To-do view**.

In the month view, dates that have calendar entries are marked with a small triangle at the right bottom corner. In the week view, memos and anniversaries are placed before 8 o'clock. Press * to switch between the month view, the week view, and the day view.

Icons in the day and week views:

Memo

Anniversary

There is no icon for **Meeting**.

To go to a certain date, select **Options** > **Go to date**, write the date, and select **OK**.

Press # to jump to today.

To-do view

Press , and select **Options** > **To-do view** to write notes and maintain a task list.

To add a note, press any key to start to write the task in the **Subject** field.

To set the due date for the task, scroll to the **Due date** field, and enter a date.

To set the priority for the **To-do** note, scroll to the **Priority** field, and scroll right to select the priority. The priority icons are ! (**High**) and ↓ (**Low**). There is no icon for **Normal**.

To mark a task as completed, scroll to it in the **To-do** list, and select **Options** > **Mark as done**.

To restore a task, scroll to it in the **To-do** list, and select **Options** > **Mark as not done**.

Remove calendar entries

Removing past entries in **Calendar** saves space in your phone memory.

To remove more than one event at a time, go to the month view, and select **Options** > **Delete entry** and one of the following:

Before date—Deletes all calendar entries that take place before a certain date you define

All entries—Deletes all calendar entries

Calendar settings

Scroll to the following settings and press the joystick to edit:

Calendar alarm tone—To change the alarm tone, scroll to the new tone and press the joystick.

Default view—To change the view that is displayed when you open Calendar, scroll to the desired view and press the joystick.

Week starts on—To change the first day of the week, scroll to the desired day and press the joystick.

Week view title—Select either **Week number** or **Week dates**. If you select any day other than Monday in **Week starts on**, the above setting is not available.

To save your settings, press **Back**.

Services

Various service providers maintain pages specifically designed for mobile devices. These pages use the wireless markup language (WML), extensible hypertext markup language (XHTML), or hypertext markup language (HTML).

Check the availability of services, pricing, and fees with your network operator or service provider. Service providers will also give you instructions on how to use their services.

Press , and select **Services**.

Shortcut: To start a connection, press and hold **0** in the standby mode.

Access point for Services

To access pages, you need service settings for the page that you want to use. You may receive the settings in a special text message from the network operator or service provider that offers the Web page. See 'Data and settings', p. 57. For more information, contact your network operator or service provider.

Tip! Settings may be available on the Web site of a network operator or service provider.

Enter the settings manually

1. Press , and select **Settings** > **Connection** > **Access points**, and define the settings for an access point. Follow the instructions given to you by your service provider. See 'Connection', p. 97.
2. Press , and select **Services** > **Options** > **Bookmark manager** > **Add bookmark**. Enter a name for the bookmark and the address of the page defined for the current access point.
3. To set the created access point as the default access point, select **Options** > **Settings** > **Access point**.

Bookmarks view

The bookmarks view opens when you open **Services**.

Glossary: A bookmark consists of an Internet address (mandatory), bookmark title, access point, and if the Web page requires, a user name and password.

Your device may have some bookmarks loaded for sites not affiliated with Nokia. Nokia does not warrant or endorse these sites. If you choose to access them, you should take the same precautions, for security or content, as you would with any Internet site.

Icons in the bookmark view:

The starting page defined for the default access point. If you use another default access point for browsing, the starting page is changed accordingly.

The automatic bookmarks folder contains bookmarks ( ) that are collected automatically when you browse pages. The bookmarks in this folder are automatically organized according to domain.

Any bookmark showing the title or Internet address of the bookmark.

Add bookmarks manually

1 In the bookmarks view, select **Options** > **Bookmark manager** > **Add bookmark**.

Start to fill in the fields. Only the URL address must be defined. The default access point is assigned to the bookmark if no other one is selected. Press * to enter special characters such as /, ., :, and @. Press to clear characters.

2 Select **Options** > **Save** to save the bookmark.

Send bookmarks

Scroll to a bookmark, and select **Options** > **Send** > **Via text message**. Press to send. It is possible to send more than one bookmark at the same time.

Make a connection

Once you have stored all the required connection settings, you can access the pages.

1 Select a bookmark or enter the address in the field ( ). When you enter the address, matching bookmarks are shown above the field. Scroll up to select a matching bookmark.

2 Press the joystick to start to download the page.

Connection security

If the security indicator is displayed during a connection, the data transmission between the device and the Internet gateway or server is encrypted.

The security icon does not indicate that the data transmission between the gateway and the content server (or place where the requested resource is stored) is secure. The service provider secures the data transmission between the gateway and the content server.

Select **Options** > **Details** > **Security** to view details about the connection, encryptions status, and information about server and user authentication.

Security features may be required for some services, such as banking services. For such connections you need security certificates. For more information, contact your service provider. See also 'Certificate management', p. 102.

Browse

Important: Use only services that you trust and that offer adequate security and protection against harmful software.

To download a page, select a bookmark, or enter the address in the field ( ).

On a browser page, new links appear underlined in blue and previously visited links in purple. Images that act as links have a blue border around them.

To open a link, to check boxes, and make selections, press the joystick.

Shortcut: Use # to jump to the end of a page and * to the beginning of a page.

To go to the previous page while browsing, select **Back**. If **Back** is not available, select **Options** > **Navigation options** > **History** to view a chronological list of the pages you have visited during a browsing session. The history list is cleared each time a session is closed.

To retrieve the latest content from the server, select **Options** > **Navigation options** > **Reload**.

To save a bookmark, select **Options** > **Save as bookmark**.

Tip! To access the bookmarks view while browsing, press and hold down the joystick. To return to the browser view again, select **Options** > **Back to page**.

To save a page while browsing, select **Options** > **Advanced options** > **Save page**. You can save pages either to the device memory or on the hard drive, and browse them when offline. To access the pages later, scroll right in the bookmarks view to open the **Saved pages** view.

To enter a new URL address, select **Options** > **Navigation options** > **Go to web address**.

To open a list of commands or actions specific for the currently open page, select **Options** > **Service options**, if available.

You can download files that cannot be shown on the page, such as ringing tones, images, operator logos, themes, and video clips. To download an item, scroll to the link, and press the joystick.

When you start a download, a list of ongoing, paused, and completed downloads from the current browsing session is displayed. You can also view the list by selecting **Options** > **Tools** > **Downloads**. In the list, scroll to an item, and select **Options** to pause, resume, or cancel ongoing downloads, or open, save, or delete completed downloads.

Downloaded items are handled by their respective applications. For example, a downloaded image is saved in **Gallery**.

Important: Only install and use applications and other software from sources that offer adequate security and protection against harmful software.

Tip! The browser collects bookmarks automatically while you are browsing pages. The bookmarks are stored to the automatic bookmarks folder ( ) and automatically organized according to domain. See also 'Settings for Services', p. 85.

View saved pages

If you regularly browse pages containing information that does not change very often, you can save and browse them when offline. In the saved pages view you can also create folders to store your saved pages.

To open the saved pages view, scroll right in the bookmarks view. In the saved pages view, press the joystick to open a saved page ( ).

To save a page while browsing, select **Options** > **Advanced options** > **Save page**.

To start a connection to the browser service and to download the latest version of the page, select **Options** > **Reload**. The device stays online after you reload the page.

Download and purchase items

You can download items such as ringing tones, images, operator logos, themes, and video clips. These items can be provided free, or you can purchase them. Downloaded items are handled by their respective applications, for example, a downloaded photo is saved in **Gallery**.

Important: Only install and use applications and other software from sources that offer adequate security and protection against harmful software.

1. To download the item, scroll to the link, and press the joystick.
2. Select the appropriate option to purchase the item.
3. Carefully read all the information provided. To continue the download, select **Accept**. To cancel the download, select **Cancel**.

For information on downloading music, see 'Go to music shop', p. 21.

Copyright protections may prevent some images, music (including ringing tones), and other content from being copied, modified, transferred or forwarded.

End a connection

Select **Options** > **Advanced options** > **Disconnect** to end the connection and view the browser page offline, or to end the connection and close the browser, select **Options** > **Exit**.

Empty the cache

The information or services you have accessed are stored in the cache memory of the device.

A cache is a memory location that is used to store data temporarily. If you have tried to access or have accessed confidential information requiring passwords, empty the

cache after each use. The information or services you have accessed are stored in the cache. To empty the cache,select **Options** > **Advanced options** > **Clear cache**.

Your device may have some bookmarks loaded for sites not affiliated with Nokia. Nokia does not warrant or endorse these sites. If you choose to access them, you should take the same precautions, for security or content, as you would with any Internet site.

Settings for Services

Select **Options** > **Settings** and the following settings:

Access point—To change the default access point, press the joystick to open a list of available access points. See 'Connection', p. 97. Some or all access points may be preset for your device by your service provider or network operator, and you may not be able to change, create, edit, or remove them.

Load imgs. & sounds—Select whether you want to load images while browsing. If you select **No**, to load images later during browsing, select **Options** > **Show images**.

Font size—To choose the text size.

Default encoding—If text characters are not shown correctly, you may choose another encoding according to language.

Auto. bookmarks—To disable automatic bookmark collecting, select **Off**. If you want to continue collecting automatic bookmarks but hide the folder from the bookmarks view, select **Hide folder**.

Screen size—To select what is shown when you are browsing. Select **Select. keys only** or **Full screen**.

Homepage—To define the home page.

Search page—To define a Web page that is downloaded when you select **Navigation options** > **Open search page** in the bookmarks view, or when browsing.

Volume—If you want the browser to play sounds embedded on Web pages, select a volume level.

Rendering—If you want the page layout shown as accurately as possible when in **Small screen** mode, select **By quality**. If you do not want external cascading style sheets to be downloaded, select **By speed**.

Cookies—To enable or disable the receiving and sending of cookies.

Glossary: Cookies are a means of content providers to identify users and their preferences for frequently used content.

Java/ECMA script—To enable or disable the use of scripts.

Security warnings—To hide or show security notifications.

Confirm touch tones—Choose whether you want to confirm before the device sends touch tones during a voice call. See also 'Options during a voice call', p. 35.

Connectivity

Bluetooth connectivity

You can connect wirelessly to other compatible devices with wireless Bluetooth technology. Compatible devices may include mobile phones, computers, and enhancements such as headsets and car kits. You can use Bluetooth connectivity to send images, video clips, music and sound clips, and notes; connect wirelessly to your compatible PC (for example, to transfer files).

Since devices with Bluetooth technology communicate using radio waves, your device and the other device do not need to be in direct line-of-sight. The two devices only need to be within 10 meters (33 feet) of each other, although the connection may be subject to interference from obstructions such as walls or from other electronic devices.

This device is compliant with Bluetooth Specification 1.2 supporting the following profiles: Basic Printing Profile, Generic Access Profile, Serial Port Profile, Dial-up Networking Profile, Headset Profile, Handsfree Profile, Generic Object Exchange Profile, Object Push Profile, File Transfer Profile, Basic Imaging Profile, and SIM Access profile. To ensure interoperability between other devices supporting Bluetooth technology, use Nokia approved enhancements for this model. Check with the manufacturers of other devices to determine their compatibility with this device.

Glossary: A profile corresponds to a service or a function, and defines how different devices connect. For example, the Handsfree Profile is used between the handsfree device and the phone. For devices to be compatible, they must support the same profiles.

There may be restrictions on using Bluetooth technology in some locations. Check with your local authorities or service provider.

Features using Bluetooth technology, or allowing such features to run in the background while using other features, increase the demand on battery power and reduce the battery life.

When the device is locked, you cannot use Bluetooth connectivity. See 'Security', p. 101 for more information on locking the device.

Settings

Press , and select **Connect.** > **Bluetooth**. When you open the application for the first time, you are asked to define a name for your device. After you set Bluetooth connectivity on and change **My phone's visibility** to **Shown to all**, your

device and this name can be seen by other users with devices using Bluetooth technology.

Select from the following:

Bluetooth—Select **On** or **Off**. To connect wirelessly to another compatible device, first set Bluetooth connectivity **On**, then establish a connection.

My phone's visibility—To allow your device to be found by other devices with Bluetooth technology, select **Shown to all**. To hide it from other devices, select **Hidden**.

My phone's name—Edit the name of your device.

Remote SIM mode—To enable another device, such as a car kit enhancement, to use the SIM card in your device to connect to the network, select **On**.

The remote SIM mode

To use the remote SIM mode, set Bluetooth connectivity on, and enable the use of the remote SIM mode with your device. Before the mode can be activated, the two devices must be paired and the pairing initiated from the other device. When pairing, use a 16-digit passcode, and set the other device as authorized. See 'Pair devices', p. 88. The remote SIM mode is activated from the other device.

When remote SIM mode is on in your Nokia N91, **Remote SIM** is displayed in the standby mode. The connection to the wireless network is turned off, as indicated by ✕ in the signal strength indicator area, and you cannot use SIM card services or features requiring cellular network coverage. However, a wireless LAN connection remains active during the remote SIM mode.

Warning: In the remote SIM mode, you cannot make or receive any calls with your device, except make calls to certain emergency numbers, or use other features that require network coverage. To make calls, you must first leave the remote SIM mode. If the device has been locked, enter the lock code.

To leave the remote SIM mode, press the power key, and select **Exit rem. SIM mode**.

Security tips

When you are not using Bluetooth connectivity, select **Bluetooth** > **Off** or **My phone's visibility** > **Hidden**.

Operating the phone in hidden mode is a safer way to avoid malicious software.

Do not pair with an unknown device. Do not accept Bluetooth connectivity from sources you do not trust.

Send data using Bluetooth connectivity

Several Bluetooth connections can be active at a time. For example, if you are connected to a headset, you can also transfer files to another compatible device at the same time.

Bluetooth connectivity indicators

- When is shown in the standby mode, Bluetooth connectivity is on.
- When is blinking, your device is trying to connect with another device.
- When is shown continuously, data is being transmitted using Bluetooth connectivity.

Tip! To send text using Bluetooth connectivity, open **Notes**, write the text, and select **Options** > **Send** > **Via Bluetooth**.

1 Open an application or folder where the item you want to send is stored. For example, to send an image to another compatible device, open **Gallery**.

2 Select the item and **Options** > **Send** > **Via Bluetooth**. The device icon, the name of the device, the device type, or a short name for devices with Bluetooth technology within range start to appear on the display.

Device icons: computer, phone, audio or video device, and other device.

To interrupt the search, select **Stop**.

3 Select the device with which you want to connect.

4 If the other device requires pairing before data can be transmitted, a tone sounds, and you are asked to enter a passcode. See 'Pair devices', p. 88.

5 When the connection has been established, **Sending data** is shown.

The **Sent** folder in **Messag.** does not store messages sent using Bluetooth connectivity.

Tip! When searching for devices, some devices may show only the unique address (device address). To find the unique address of your device, enter the code ***#2820#** in the standby mode.

Pair devices

To open the paired devices view ( ), in the **Bluetooth** application main view, scroll right.

Before pairing, create your own passcode (1—16 digits), and agree with the owner of the other device to use the same code. Devices that do not have a user interface have a factory-set passcode. The passcode is used only once.

To pair with a device, select **Options** > **New paired device**. Devices with wireless Bluetooth technology within range start to appear on the display. Select the device, and enter the passcode. The same passcode must be entered on the other device as well. After pairing, the device is saved to the paired devices view.

Paired devices are indicated by in the device search.

To set a device as authorized or unauthorized, scroll to a device, and select from the following options:

Set as authorized—Connections between your device and this device can be made without your knowledge. No separate acceptance or authorization is needed. Use this status for your own devices, such as your compatible headset or PC, or devices that belong to someone you trust. indicates authorized devices in the paired devices view.

Set as unauthorized—Connection requests from this device must be accepted separately every time.

To cancel a pairing, scroll to the device, and select **Options** > **Delete**. If you want to cancel all pairings, select **Options** > **Delete all**.

Tip! If you are currently connected to a device and delete the pairing with that device, pairing is removed immediately, and the connection is switched off.

Receive data using Bluetooth connectivity

When you receive data through Bluetooth connectivity, a tone sounds, and you are asked if you want to accept the message. If you accept, is shown, and the item is placed in the **Inbox** folder in **Messag.**. Messages received through Bluetooth connectivity are indicated by .
See 'Inbox—receive messages', p. 56.

Switch off Bluetooth connectivity

To switch off Bluetooth connectivity, select **Bluetooth** > **Off**.

Data cable

You can connect to a compatible PC with a USB data cable. With data cable connectivity, you can use Nokia PC Suite and transfer music or other data, such as image files, between your device and the PC. See 'Transfer music', p. 21 for more information on transferring music.

Press , and select **Connect.** > **Data cbl.** > **Data cable mode**. Select **Media player** to transfer music, **PC Suite** to use Nokia PC Suite, or **USB** to transfer other data. To have the device ask the purpose of the connection each time the cable is connected, select **Ask on connection**.

PC connections

You can use your device with a variety of PC connectivity and data communications applications. With Nokia PC Suite you can, for example, synchronize contacts, calendar and To-do notes, and transfer images between your device and a compatible PC. Because of the high capacity of the 4 GB hard drive you can store hundreds of images and songs in the device.

Always create the connection from the PC to synchronize with your device.

For further information on how to install Nokia PC Suite (compatible with Windows 2000 and Windows XP), see the User Guide for Nokia PC Suite and the Nokia PC Suite help in the 'Install' section on the CD-ROM supplied in the sales package.

CD-ROM

The CD-ROM should launch after you have inserted it into the CD-ROM drive of a compatible PC. If not, proceed as follows: Open Windows Explorer, right-click the CD-ROM drive where you inserted the CD-ROM, and select Autoplay.

Your device as a modem

You can use your device as a modem to send and receive e-mail, or to connect to the Internet with a compatible PC by using a Bluetooth connection or a data cable. Detailed installation instructions can be found in the User Guide for Nokia PC Suite in Modem options on the CD-ROM.

Tip! When using Nokia PC Suite for the first time, to connect your device to a compatible PC and to use Nokia PC Suite, use the Get Connected wizard available in Nokia PC Suite. See the CD-ROM supplied in the sales package.

Connection manager

You may have multiple data connections active at the same time when you are using your device in GSM and UMTS networks. Press , and select **Connect.** > **Conn mgr** To view the status of data connections or end connections in GSM and UMTS network and wireless LAN, select **Act. data conn.**. To search for wireless LANs available within range, select **Availab. WLAN**.

Data connections

In the active connections view, you can see the open data connections: data calls ( ), packet data connections (or), and wireless LAN connections ( ).

Note: The actual time invoiced for calls by your service provider may vary, depending on network features, rounding off for billing, and so forth.

To end a connection, select **Options** > **Disconnect**. To close all open connections, select **Options** > **Disconnect all**.

To view the details of a connection, select **Options** > **Details**. The details shown depend on the connection type.

View data connection details

To view the details of a connection, scroll to a connection, and select **Options** > **Details**.

Name—The name of the Internet access point (IAP) in use, or the modem connection name if the connection is a dial-up connection.

Bearer—The type of data connection: **Data call**, **High speed (GSM)**, or **Packet data**.

Status—The current status of the connection: **Connecting**, **Conn. (inactive)**, **Conn.(active)**, **On hold**, **Disconnecting**, or **Disconnected**.

Recvd.—The amount of data, in bytes, received to your device.

Sent—The amount of data, in bytes, sent from your device.

Duration—The length of time that the connection has been open.

Speed—The current speed of both sending and receiving data in kilobytes per second.

Dial-up—The dial-up number used.

Access pt.—Access point name used.

Shared (not shown if the connection is not shared)—The number of applications using the same connection.

Wireless Local Area Network (WLAN)

Wireless LAN

The available wireless LAN view shows a list of wireless LANs within range, their network mode (**Infrastructure** or **Ad-hoc**), and signal strength indicator. is shown for networks with encryption, and if your device has an active connection in the network.

To view the details of a network, select **Options** > **Details**.

To create an Internet access point in a network, select **Options** > **Define access point**. For more information on WLAN, see 'Wireless LAN' on p.13.

Synchronization

Press , and select **Connect.** > **Sync**. **Sync** enables you to synchronize your notes, calendar, and contacts with various calendar and address book applications on a compatible PC or on the Internet.

The synchronization application uses SyncML technology for synchronization. For information on SyncML compatibility, contact the supplier of the calendar or address book application you want to synchronize your phone data with.

You may receive syncronization settings in a special text message. See 'Data and settings', p. 57.

Create a new synchronization profile

1 If no profiles have been defined, your device asks if you want to create a new profile. Select **Yes**.

To create a new profile in addition to existing ones, select **Options** > **New sync profile**. Choose whether you want to use the default setting values or copy the values from an existing profile to be used as the basis for the new profile.

2 Define the following:

Sync profile name—Write a descriptive name for the profile.

Applications—Select the applications to synchronize with this profile, **Contacts** and **Calendar**, for example. Scroll to each application and press the joystick to define its synchronization settings:

- **Include in sync**—Select **Yes** if you want to synchronize the application.
- In **Remote database**, enter a correct path to the remote calendar, address book, or database on the server.
- Select **Synchronization type**: **Normal** (two-way synchronization), **To server only**, or **To phone only**.

Connection settings—Define the following. Contact your service provider or system administrator for the correct values.

- **Server version**—Select the SyncML version the synchronization server uses. You must select **1.2** to synchronize e-mail.
- **Server ID**—Enter the server ID for the e-mail server (shown only if **Server version** is set to **1.2**).
- **Data bearer**—Select the connection type: **Internet**, or **Bluetooth**.
- **Access point** (shown only if **Data bearer** is set to **Internet**)—Select an access point to use for the data connection.
- **Host address**—The IP address of the host server.
- **Port**—The server port number (shown only if **Data bearer** is set to **Internet**).
- **User name**—Your user ID for the synchronization server.
- **Password**—Your password for the synchronization server.
- **Allow sync requests**—Select **Yes** if you want to allow the server to start a synchronization.
- **Accept all sync reqs.**—Select **No** if you want your device to ask you before a synchronization initialized by the server is started.
- **Network authentic.** (shown only if **Data bearer** is set to **Internet**)—Select **Yes** to enter a network user name and password. Scroll down to view the user name and password fields.

3 Select **Back** to save the settings and return to the main view.

Synchronize data

In the **Sync** main view, you can see the different synchronization profiles and the kind of data to be synchronized.

1 Select a synchronization profile and **Options** > **Synchronize**. The status of the synchronization is shown at the bottom of the screen.

To cancel synchronization before it finishes, select **Cancel**.

2 You are notified when the synchronization is complete. After synchronization is complete, select **Options** > **View log** to open a log file showing the synchronization status (**Complete** or **Incomplete**) and how many calendar or contact entries, or notes have been added, updated, deleted, or discarded (not synchronized) in your device or on the server.

Device manager

Press , and select **Connect.** > **Dev. mgr.**. You may receive server profiles and different configuration settings from your network operator, service provider, or company information management department. These configuration settings may include access point settings for data connections and other settings used by different applications in your device.

To connect to a server and receive configuration settings for your device, select **Start configuration**.

To allow or deny receiving of configuration settings from a service provider, select **Enable config.** or **Disable config.**.

Server profile settings

Contact your service provider for the correct settings.

Server name—Enter a name for the configuration server.

Server ID—Enter the unique ID to identify the configuration server.

Server password—Enter a password to identify your device to the server.

Session mode— Select **Bluetooth** or **Internet**.

Access point—Select an access point to be used when connecting to the server.

Host address—Enter the URL address of the server.

Port—Enter the port number of the server.

User name and **Password**—Enter your user name and password.

Allow configuration—To receive configuration settings from the server, select **Yes**.

Auto-accept all reqs.—If you want your device to ask for confirmation before accepting a configuration from the server, select **No**.

Tools

Calculator

To add, substract, multiply, divide, calculate square roots and percentages, press , and select **Tools** > **Calculator**.

Note: This calculator has limited accuracy and is designed for simple calculations.

To save a number in the memory (indicated by **M**), select **Options** > **Memory** > **Save**. To retrieve a number in memory, select **Options** > **Memory** > **Recall**. To clear a number in memory, select **Options** > **Memory** > **Clear**.

Calculate percentages

1. Enter a number for which you want to calculate a percentage.
2. Select |×|, |÷|, |−|, or |+|.
3. Enter the percentage.
4. Select |%|.

Converter

To convert measures such as **Length** from one unit (**Yards**) to another (**Meters**), press , and select **Tools** > **Converter**.

Note that **Converter** has limited accuracy, and rounding errors may occur.

1. Scroll to the **Type** field, and press the joystick to open a list of measures. Scroll to the measure you want to use, and select **OK**.
2. Scroll to the first **Unit** field, and press the joystick. Select the unit from which you want to convert and **OK**. Scroll to the next **Unit** field, and select the unit to which you want to convert.
3. Scroll to the first **Amount** field, and enter the value you want to convert. The other **Amount** field changes automatically to show the converted value.

   Press **#** to add a decimal and ***** for the **+**, **–** (for temperature), and **E** (exponent) symbols.

Tip! To change the conversion order, enter the value in the second **Amount** field. The result is shown in the first **Amount** field.

Set base currency and exchange rates

Before you can make currency conversions, you need to choose a base currency and add exchange rates. The rate of the base currency is always 1. The base currency determines the conversion rates of the other currencies.

1 Select **Converter** > **Options** > **Currency rates**. A list of currencies opens, and you can see the current base currency at the top.

Tip! To rename a currency, go to the currency rates view, scroll to the currency, and select **Options** > **Rename currency**.

2 To change the base currency, scroll to the currency, and select **Options** > **Set as base curr.**.

3 Add exchange rates. Scroll to the currency, and enter a new rate, that is, how many units of the currency equal one unit of the base currency you have selected.

After you have inserted all the necessary exchange rates, you can make currency conversions.

Note: When you change base currency, you must enter the new rates because all previously set exchange rates are set to zero.

Settings

To change settings, press , and select **Tools** > **Settings**. Scroll to a setting group, and press the joystick to open it. Scroll to a setting you want to change, and press the joystick.

Device settings

General

Phone language—Changing the language of the display texts in your device also affects the format used for date and time and the separators used, for example, in calculations. **Automatic** selects the language according to the information on your SIM card. After you change the display text language, the device restarts.

Changing the settings for **Phone language** or **Writing language** affects every application in your device, and the change remains effective until you change these settings again.

Writing language—Changing the language affects the characters and special characters available when writing text and the predictive text dictionary used.

Predictive text—You can set the predictive text input **On** or **Off** for all editors in the device. The predictive text dictionary is not available for all languages.

Welcome note / logo—The welcome note or logo is displayed briefly each time you switch on the device. Select **Default** to use the default image, **Text** to write a welcome note (up to 50 letters), or **Image** to select a photo or picture from **Gallery**.

Orig. phone settings—You can reset some of the settings to their original values. To do this, you need the lock code. See 'Security', 'Phone and SIM', p. 102. After resetting, the device may take a longer time to power on. Documents and files are not affected.

Standby mode

Active standby—Use shortcuts to applications in the standby mode on the display. See 'Active standby mode', p. 31.

Left selection key—To assign a shortcut to the left selection key ( ) in the standby mode, select an application from the list.

Right selection key—To assign a shortcut to the right selection key ( ) in the standby mode, select an application from the list.

Active standby apps.—To select the application shortcuts you want to appear in the active standby mode, select the applications from the list. This setting is only available if **Active standby** is on.

You can also assign shortcuts to different movements of the joystick. Select the application for each movement from a list. The joystick shortcuts are only available if **Active standby** is off.

Operator logo—This setting is visible only if you have received and saved an operator logo. You can choose whether you want the operator logo to show or not.

Display

Brightness—You can change the brightness of the display to be lighter or darker.

Power saver time-out—The power saver is activated when the time-out period is over.

Light time-out—Select a time-out after which the backlight of the display is switched off.

Call settings

Send my caller ID (network service)—You can set your phone number to be displayed to (**Yes**) or hidden from (**No**) the person to whom you are calling, or the value may be set by your network operator or service provider when you make a subscription (**Set by network**).

Call waiting (network service)—If you have activated call waiting, the network notifies you of a new incoming call while you have a call in progress. Select **Activate** to request the network to activate call waiting, **Cancel** to request the network to deactivate call waiting, or **Check status** to check if the function is active or not.

Reject call with SMS—Select **Yes** to send a text message to a caller informing why you could not answer the call. See 'Answer or decline a call', p. 34.

Message text—Write a text to be sent in a text message when you decline a call.

Automatic redial—Select **On**, and your device makes a maximum of ten attempts to connect the call after an unsuccessful call attempt. To stop automatic redialing, press .

Summary after call—Activate this setting if you want the device to briefly display the approximate duration of the last call.

1-touch dialing—Select **On**, and the numbers assigned to the 1-touch dialing keys (**2** - **9**) can be dialed by pressing and holding the key. See also '1-touch dial a phone number', p. 33.

Anykey answer—Select **On**, and you can answer an incoming call by briefly pressing any key under the slide cover.

Line in use (network service)—This setting is shown only if the SIM card supports two subscriber numbers, that is, two phone lines. Select which phone line you want to use for making calls and sending text messages. Calls on both lines can be answered irrespective of the selected line. If you select **Line 2** and have not subscribed to this network service, you will not be able to make calls. When line 2 is selected, **2** is shown in the standby mode.

Tip! To switch between the phone lines, press and hold **#** in the standby mode.

Line change (network service)—To prevent line selection, select **Line change** > **Disable** if supported by your SIM card. To change this setting, you need the PIN2 code.

Connection

Data connections and access points

Your device supports packet data connections ( ), such as GPRS in the GSM network.

Glossary: General packet radio service (GPRS) uses packet data technology where information isbsent in short packets of data over the mobile network.

To establish a data connection, an access point is required. You can define different kinds of access points, such as:

- MMS access point to send and receive multimedia messages
- Access point for the Web application to view WML or XHTML pages
- Internet access point (IAP) to send and receive e-mail

Check the type of access point you need with your service provider for the service you wish to access. For availability and subscription to packet data connection services, contact your network operator or service provider.

Packet data connections in GSM and UMTS networks

When you use your device in GSM and UMTS networks, multiple data connections can be active at the same time and access points can share a data connection. In the UMTS network, data connections remain active during voice calls. To check the active data connections, see 'Connection manager', p. 90.

The following indicators may be displayed below the signal indicator, depending on which network you use:

GSM network, packet data is available in the network.

GSM network, packet data connection is active, data is being transferred.

GSM network, multiple packet data connections are active.

GSM network, packet data connection is on hold. (This can happen during a voice call, for example.)

UMTS network, packet data is available in the network.

UMTS network, packet data connection is active, data is being transferred.

UMTS network, multiple packet data connections are active.

UMTS network, packet data connection is on hold.

Access points

Some or all access points may be preset for your device by your service provider or network operator, and you may not be able to change, create, edit, or remove them. indicates a protected access point. You may also receive access point settings in a text message from a service provider. See 'Data and settings', p. 57.

Tip! See also 'Receive MMS and e-mail settings', p. 56, 'E-mail', p. 63, and 'Access point for Services', p. 81.

indicates a packet data access point, and a wireless LAN access point.

To create a new access point, press , and select **Tools** **Settings** > **Connection** > **Access points** > **Options** > **New access point**.

Follow the instructions given to you by your service provider.

Connection name—Give a descriptive name for the connection.

Data bearer—Depending on what data connection you select, only certain setting fields are available. Fill in all fields marked with **Must be defined** or with a red asterisk. Other fields can be left empty, unless you have been instructed otherwise by your service provider.

To be able to use a data connection, the network service provider must support this feature, and if necessary, activate it for your SIM card.

Packet data

Follow the instructions from your service provider.

Access point name (for packet data only)—The access point name is needed to establish a connection to the packet data and UMTS networks. You obtain the access point name from your network operator or service provider.

User name—The user name may be needed to make a data connection, and it is usually provided by the service provider. The user name is often case-sensitive.

Prompt password—Select **Yes** if you must enter a new password every time you log in to a server, or if you do not want to save your password in your device.

Password—A password may be needed to make a data connection, and it is usually provided by the service provider. The password is often case-sensitive.

Authentication—Select **Normal** or **Secure**.

Homepage—Depending on what you are setting up, write the Web address or the address of the multimedia messaging center. Complete the **Advanced settings** to further define your access details:

Network type—Select the Internet protocol type to use: **IPv4 settings** or **IPv6 settings**. The other settings depend on the selected network type.

Phone IP address (for IPv4)—Enter the IP address of your device.

DNS address— **Primary name serv.:**, **Second. name server:** Contact your Internet service provider to obtain these addresses.

Glossary: The domain name service (DNS) is an Internet service that translates domain names such as **www.nokia.com** into IP addresses like 192.100.124.195.

Proxy serv. address—Define the address for the proxy server.

Proxy port number—Enter the proxy port number.

Wireless LAN

Follow the instructions from your service provider.

WLAN netw. name—Select **Enter manually** or **Search for netw.**. If you select an existing network, **WLAN netw. mode** and **WLAN security mode** are determined by the settings of its access point device.

WLAN netw. mode—Select **Ad-hoc** to create an ad-hoc network and to allow devices to send and receive data directly; a wireless LAN access point device is not needed.

WLAN security mode—Select the encryption used: **WEP**, **802.1x** (not for ad hoc networks), or **WPA/WPA2**. If you select **Open network**, no encryption is used. The WEP, 802.1x, and WPA functions can be used only if the network supports them.

WLAN security sett.—Enter the settings for the selected security mode:

Security settings for **WEP**:

Wep key in use—Select the WEP key number. You can create up to four WEP keys. The same settings must be entered on the wireless LAN access point device.

Authentication type—Select **Open** or **Shared** for the authentication type between your device and the wireless LAN access point device.

WEP key settings—Enter **WEP encryption** (length of the key), **WEP key format** (**ASCII** or **Hexadecimal**), and **WEP key** (the WEP key data in the selected format).

Security settings for **802.1x** and **WPA/WPA2**:

WPA mode—Select the means of authentication: **EAP** to use an extensible authentication protocol (EAP) plug-in, or **Pre-shared key** to use a password. Complete the appropriate settings:

EAP plug-in settings (only for **EAP**)—Enter the settings as your service provider instructs.

Pre-shared key (only for **Pre-shared key**)—Enter a password. The same password must be entered on the wireless LAN access point device.

TKIP encryption—Select whether Temporal Key Integrity Protocol (TKIP) is used.

Homepage—Define a home page.

Select **Options** > **Advanced settings** and from the following:

IPv4 settings: **Phone IP address** (the IP address of your device), **Subnet mask** (the subnet IP address), **Default gateway** (the gateway), and **DNS address**—enter the IP addresses of the primary and secondary DNS server. Contact your Internet service provider for these addresses.

IPv6 settings > **DNS address**—Select **Automatic**, **Well-known**, or **User defined**.

Ad-hoc channel (only for **Ad-hoc**)—To enter a channel number (1-11) manually, select **User defined**.

Proxy serv. address—Enter the address for the proxy server.

Proxy port number—Enter the proxy port number.

Packet data

The packet data settings affect all access points using a packet data connection.

Packet data conn.—If you select **When available** and you are in a network that supports packet data, the device registers to the packet data network. Also, starting an active packet data connection is quicker (for example, to send and receive e-mail). If you select **When needed**, the device uses a packet data connection only if you start an application or action that needs it. If there is no packet data coverage and you select **When available**, the device periodically tries to establish a packet data connection.

Access point—The access point name is needed to use your device as a packet data modem to your PC.

Configurations

You may receive trusted server settings from your network operator or service provider in a configuration message, or the settings may be stored on your SIM or USIM card. You can save these settings to your device, view, or delete them in **Configurations**.

Date & time

See 'Clock settings', p. 16.

See also the language settings in 'General', p. 95.

Security

Phone and SIM

PIN code request—When active, the code is requested each time the device is switched on. Deactivating the personal identification number (PIN) code request may not be allowed by some SIM cards. See 'Glossary of PIN and lock codes', p. 101.

PIN code, **PIN2 code**, and **Lock code**—You can change the lock code, PIN code, and PIN2 code. These codes can only include the numbers from **0** to **9**. See 'Glossary of PIN and lock codes', p. 101.

Avoid using access codes similar to the emergency numbers to prevent accidental dialing of the emergency number.

If you forget any of these codes, contact your service provider.

Autolock period—You can set an autolock period, a timeout after which the device automatically locks and can be used only if the correct lock code is entered. Enter a number for the timeout in minutes, or select **None** to turn off the autolock period.

To unlock the device, enter the lock code.

When the device is locked, calls still may be possible to the official emergency number programmed into your device.

Tip! To lock the device manually, press . A list of commands opens. Select **Lock phone**. For information on Lock switch see 'Keypad lock (keyguard)', p. 18.

Lock if SIM changed—You can set the device to ask for the lock code when an unknown SIM card is inserted into your device. The device maintains a list of SIM cards that are recognized as the owner's cards.

Closed user group (network service)—You can specify a group of people to whom you can call and who can call you.

When calls are limited to closed user groups, calls still may be possible to the official emergency number programmed into your device.

Confirm SIM services (network service)—You can set the device to display confirmation messages when you are using a SIM card service.

Glossary of PIN and lock codes

If you forget any of these codes, contact your service provider.

Personal identification number (PIN) code—This code protects your SIM card against unauthorized use. The PIN code (4 to 8 digits) is usually supplied with the SIM card. After three consecutive incorrect PIN code entries, the PIN code is blocked, and you need to unblock it before you can use the SIM card again. See the information about the PUK code in this section.

UPIN code—This code may be supplied with the USIM card. The USIM card is an enhanced version of the SIM card and is supported by UMTS mobile phones. The UPIN code protects the USIM card against unauthorized use.

PIN2 code—This code (4 to 8 digits) is supplied with some SIM cards, and is required to access some functions in your device.

Lock code (also known as security code)—This code (5 digits) can be used to lock the device to avoid unauthorized use. The factory setting for the lock code is **12345**. To avoid unauthorized use of your device, change the lock code. Keep the new code secret and in a safe place separate from your device.

Personal unblocking key (PUK) code and PUK2 code—These codes (8 digits) are required to change a blocked PIN code or PIN2 code, respectively. If the codes are not supplied with the SIM card, contact the operator whose SIM card is in your device.

UPUK code—This code (8 digits) is required to change a blocked UPIN code. If the code is not supplied with the USIM card, contact the operator whose USIM card is in your device.

Certificate management

Digital certificates do not guarantee safety; they are used to verify the origin of software.

Glossary: Digital certificates are used to verify the origin of the XHTML or WML pages and installed software. However, they can only be trusted if the origin of the certificate is known to be authentic.

In the certificate management main view, you can see a list of authority certificates that are stored in your device. Scroll right to see a list of personal certificates, if available.

Digital certificates should be used if you want to connect to an online bank or another site or remote server for actions that involve transferring confidential information. They should also be used if you want to reduce the risk of viruses or other malicious software and be sure of the authenticity of software when downloading and installing software.

Important: Even if the use of certificates makes the risks involved in remote connections and software installation considerably smaller, they must be used correctly in order to benefit from increased security. The existence of a certificate does not offer any protection by itself; the certificate manager must contain correct, authentic, or trusted certificates for increased security to be available. Certificates have a restricted lifetime. If **Expired certificate** or **Certificate not valid yet** is shown even if the certificate should be valid, check that the current date and time in your device are correct.

View certificate details—check authenticity

You can only be sure of the correct identity of a server when the signature and the period of validity of a server certificate have been checked.

You are notified on the device display if the identity of the server is not authentic or if you do not have the correct security certificate in your device.

To check certificate details, scroll to a certificate, and select **Options** > **Certificate details**. When you open certificate details, the validity of the certificate is checked, and one of the following notes may be displayed:

- **Certificate not trusted**—You have not set any application to use the certificate. See 'Change the trust settings', p. 103.
- **Expired certificate**—The period of validity has ended for the selected certificate.
- **Certificate not valid yet**—The period of validity has not yet begun for the selected certificate.
- **Certificate corrupted**—The certificate cannot be used. Contact the certificate issuer.

Change the trust settings

Before changing any certificate settings, you must make sure that you really trust the owner of the certificate and that the certificate really belongs to the listed owner.

Scroll to an authority certificate, and select **Options** > **Trust settings**. Depending on the certificate, a list of the applications that can use the selected certificate is shown. For example:

- **Symbian installation**: **Yes**—The certificate is able to certify the origin of a new Symbian operating system application.
- **Internet**: **Yes**—The certificate is able to certify servers.
- **App. installation**: **Yes**—The certificate is able to certify the origin of a new Java application.

Select **Options** > **Edit trust setting** to change the value.

Track protect. object

Some copyright-protected music files may include a transaction identifier. Select whether you allow the identifier to be sent with the music file if you send or forward the file. The provider of the music file can use the identifier to track the distribution of the files.

EAP plug-ins

You can view the EAP plug-ins currently installed in your device (network service). Some Internet access points that use wireless LAN as a data bearer and WPA security mode use the plug-ins for authentication.

Call forwarding

Call forwarding allows you to forward your incoming calls to your voice mailbox or another phone number. For details, contact your service provider.

1 Press , and select **Tools** > **Settings** > **Call forwarding**.

2 Select which calls you want to forward: **Voice calls**, **Data calls**, or **Fax calls**.

3 Select the desired forwarding option. To forward voice calls when your number is busy or when you decline incoming calls, select **If busy**.

4 Set the forwarding option on (**Activate**) or off (**Cancel**), or check whether the option is activated (**Check status**). Several forwarding options can be active at the same time.

When all calls are forwarded, is shown in the standby mode.

Call restrictions and call forwarding cannot be active at the same time.

Call restriction

Call restriction (network service) allows you to restrict the calls that you make or receive with the device. To change the settings, you need the restriction password from your service provider. Select the desired restricting option, and set it on (**Activate**) or off (**Cancel**), or check whether the option is active (**Check status**). **Call restriction** affects all calls, including data calls.

Call restrictions and call forwarding cannot be active at the same time.

When calls are restricted, calls still may be possible to certain official emergency numbers.

Network

Your device can automatically switch between the GSM and UMTS networks. The GSM network is indicated with Y in the standby mode. The UMTS network is indicated with **3G**.

Network mode (shown only if supported by the operator)—Select which network to use. If you select **Dual mode**, the device uses the GSM or UMTS network automatically, according to the network parameters and the roaming agreements between the network operators. Contact your network operator for more details.

Operator selection—Select **Automatic** to set the device to search for and select one of the available networks, or **Manual** to manually select the network from a list of networks. If the connection to the manually selected network is lost, the device sounds an error tone and asks you to reselect a network. The selected network must have a roaming agreement with your home network, that is, the operator whose SIM card is in your device.

Glossary: A roaming agreement is an agreement between two or more network service providers to enable the users of one service provider to use the services of other service providers.

Cell info display—Select **On** to set the device to indicate when it is used in a cellular network based on micro-cellular network (MCN) technology and to activate cell info reception.

Enhancement settings

Indicators shown in the standby mode:

—A headset is connected.

—A headset is unavailable, or a Bluetooth connection to a headset is lost.

Select **Headset** or **Bluetooth handsfree** and the following options are available:

Default profile—To set the profile that you want activated each time you connect a certain enhancement to your device. See 'Profiles—Set tones', p. 29.

Automatic answer—To set the device to answer an incoming call automatically after 5 seconds. If the Ringing type is set to **Beep once** or **Silent**, automatic answer is disabled.

Voice commands

You can use voice commands to control your device. For more information about the enhanced voice commands supported by your device, see 'Voice dialing', p. 33.

To activate enhanced voice commands for starting applications or profiles, you must open the **Voice comms.** application and its **Profiles** folder. Press , and select **Tools** > **Voice cm.** > **Profiles**. The device creates voice commands for the applications and profiles. You can now use enhanced voice commands by pressing and holding the right selection key (—) and saying a voice command in the standby mode. The voice command is the name of the application or profile displayed in the list.

To add more applications to the list, select **Options** > **New application**. To add a second voice command that can be used to start the application, scroll to it, select **Options** > **Change command**, and enter the new voice command as text. Avoid very short names, abbreviations, and acronyms.

To change voice command settings, select **Options** > **Settings**. To switch off the synthesiser that plays recognized voice tags and commands in the selected device language, select **Synthesizer** > **Off**. To reset voice recognition learning, for example, when the main user of the device has changed, select **Reset voice adapts.**.

Application manager

Press , and select **Tools** > **App. mgr.**. You can install two types of applications and software to your device:

- J2ME™ applications based on Java™ technology with the extension .jad or .jar ().
- Other applications and software suitable for the Symbian operating system (). The installation files have the .sis extension. Only install software specifically designed for your Nokia N91. Software providers will often refer to the official model number of this product: the Nokia N91-1.

Installation files may be transferred to your device from a compatible PC, downloaded during browsing, or sent to you in a multimedia message, as an e-mail attachment, or using Bluetooth connectivity. You can use Nokia Application Installer in Nokia PC Suite to install an application to your device. If you use Microsoft Windows Explorer to transfer a file, save the file to the hard drive of your device (local disk).

Example: If you have received the installation file as an e-mail attachment, go to your mailbox, open the e-mail, open the attachments view, scroll to the installation file, and press the joystick to start installation.

Install applications and software

Tip! You can also use Nokia Application Installer available in Nokia PC Suite to install applications. See the CD-ROM supplied with the device.

Application icons are as follows:

—.sis application

—Java application

—application is not fully installed.

Important: Only install and use applications and other software from sources that offer adequate security and protection against harmful software.

Before installation, note the following:

- To view the application type, version number, security certificate details, and the supplier or manufacturer of the application, select **Options** > **View details**.
- If you install a file that contains an update or repair to an existing application, you can only restore the original application if you have the original installation file or a full back-up copy of the removed software package. To restore the original application, remove the application, and install the application again from the original installation file or the back-up copy.
- The .jar file is required for installing Java applications. If it is missing, the device may ask you to download it. If there is no access point defined for the application, you are asked to select one. When you are downloading the .jar file, you may need to enter a user name and password to access the server. You obtain these from the supplier or manufacturer of the application.

To install applications and software complete these steps:

1 Open **App. mgr.**, and scroll to an installation file. Alternatively, search the device memory or the hard drive in **File mgr.**, or open a message in **Messag.** > **Inbox** that contains an installation file.

2 Select the application, and press the joystick to start the installation.

Tip! While browsing, you can download an installation file, and install it without closing the connection.

During installation, the device shows information about the progress of the installation. If you are installing an application without a digital signature or certification, the device displays a warning. Continue installation only if you are sure of the origin and contents of the application.

To start an installed application, locate it in the menu, and press the joystick.

To start a network connection and to view extra information about the application, scroll to it, and select **Options** > **Go to web address**, if available.

To see what software packages were installed or removed and when, select **Options** > **View log**.

To send your installation log to a help desk so that they can see what was installed or removed, select **Options** > **Send log** > **Via text message** or **Via e-mail** (available only if the correct e-mail settings are in place).

Remove applications and software

Scroll to a software package, and select **Options** > **Remove**. Select **Yes** to confirm.

If you remove software, you can only reinstall it if you have the original software package or a full backup of the removed software package. If you remove a software package, you may no longer be able to open documents created with that software.

If another software package depends on the software package that you removed, the other software package may stop working. Refer to the documentation of the installed software package for details.

Application settings

Select **Options** > **Settings** and from the following:

Online certif. check—Select to check the online certificates before installing an application.

Default web address—Set the default address used when checking online certificates.

Some Java applications may require a phone call, a message to be sent, or a network connection to a specific access point for downloading extra data or components. In the **App. mgr.** main view, scroll to an application, and select **Options** > **Suite settings** to change settings related to that specific application.

Memory-Hard drive

To monitor the amount of used and free disk space on the device hard drive, press , and select **Tools** > **Hard drive**. To see how much disc space the different applications and data types use, select **Options** > **Memory details**.

To change the name of the hard drive, select **Options** > **Hard drive name**.

Tools

Use a charger when scanning, defragmenting or formatting the hard drive.

Scan hard drive

You can scan your hard drive to find and fix possible errors. The problems are fixed automatically.

To scan your hard drive, select **Options** > **Tools** > **Scan hard drive**. A progress bar is displayed.

After the scanning is completed you will be informed of the results.

Defragment hard drive

It is recommendable to scan the hard drive before defragmenting it.

To optimize your hard drive, select **Options** > **Tools** > **Defragment hard drive**.

You will be informed when the defragmentation is completed.

Format hard drive

When the hard drive is formatted, all data on the drive is permanently lost. Back up data you want to keep before formatting the hard drive. You can use Nokia PC Suite to back up data to a compatible PC. If your device needs to be repaired or replaced, a back-up copy is required to restore data protected by digital rights management technologies (DRM).

It may happen that the hard drive becomes corrupted and needs to be formatted to be fully functional again. Formatting removes all the data from the hard drive. Formatting the hard drive increases the demand on battery power. It is advisable not to use your device for anything else and connect it to a charger and keep the charger connected while your device formats the hard drive.

To format the hard drive, select **Options** > **Quick format** or **Full format**. **Full format** scans the drive for errors before formatting. **Quick format** deletes all the information from the disk without scanning the disk first. Use this option for a previously formatted disk and if you are sure the disk is not damaged.

Digital rights management technologies may prevent some back-up data from being restored. Contact your service provider for more information about DRM used for your content.

Digital Rights Management

Content owners may use different types of digital rights management technologies (DRM) to protect their intellectual property including copyrights. This device uses applicable DRM software to access DRM protected content. Ilf certain DRM software fails to protect the content, content owners may ask the software's ability to access the new content protected by such DRM to be revoked. Revocation does not affect the use of content that is already in your device. With this device you can access content protected with OMA 1.0 and OMA 2.0.

To view activation keys for DRM protected content stored in your device, press , and select **Tools** > **Actv. keys** and from the following:

Valid keys—View keys that are connected to one or more media files and keys whose validity period has not started yet.

Invalid keys—View keys that are not valid; the time period for using the media file is exceeded or there is a protected media file in the device but no connected activation key.

Not in use—View keys that have no media files connected to them in the device.

To buy more usage time or extend the usage period for a media file, select an invalid activation key and **Options** > **Get activation key**. Activation keys cannot be updated if Web service message reception is disabled. See 'Web service messages', p. 57.

To view detailed information, such as the validity status and ability to send the file, scroll to an activation key, and press the joystick.

Troubleshooting

Q&A

Bluetooth connectivity

Q: Why can't I find my friend's device?

A: Check that both devices have activated Bluetooth connectivity.

Check that the distance between the two devices is not over 10 metres and that there are no walls or other obstructions between the devices.

Check that the other device is not in hidden mode.

Check that both devices are compatible.

Q: Why can't I end a Bluetooth connection?

A: If another device is connected to your phone, you can either end the connection using the other device or by deactivating Bluetooth connectivity. Select **Connect.** > **Bluetooth** > **Off**.

Multimedia messaging

Q: What should I do when the phone cannot receive a multimedia message because memory is full?

A: The amount of memory needed is indicated in the error message: **Not enough memory to retrieve message. Delete some data first.** To view what kind of data you have and how much memory the different data groups consume, select **File manager** > **Options** > **Memory details**.

Q: The note **Retrieving message** is shown briefly. What is happening?

A: The phone is trying to retrieve a multimedia message from the multimedia messaging center.

Check that the settings for multimedia messaging are defined correctly and that there are no mistakes in phone numbers or addresses. Select **Messaging** > **Options** > **Settings** > **Multim. msg.**.

Messaging

Q: Why can't I select a contact?

A: The contact card does not have a phone number or an e-mail address. Add the missing information to the contact card in **Contacts**.

Camera

Q: Why do images look smudgy?

A: Ensure that the camera lens protection window is clean.

Calendar

Q: Why are the week numbers missing?

A: If you have changed the calendar settings so that the week starts on a day other than Monday, the week numbers are not shown.

Browser services

Q: What do I do if the following message is displayed: **No valid access point defined. Define one in Services settings.**?

A: Insert the correct browser settings. Contact your service provider for instructions.

Log

Q: Why does the log appear empty?

A: You may have activated a filter, and no communication events fitting that filter have been logged. To see all events, select **Connect.** > **Log** > **Options** > **Filter** > **All communication**.

Q: How do I delete my log information?

A: Select **Connect.** > **Log** > **Options** > **Clear log** or go to **Settings** > **Log duration** > **No log**. These will erase the log contents, recent calls log, and message delivery reports permanently.

PC connectivity

Q: Why do I have problems in connecting the phone to my PC?

A: Make sure that Nokia PC Suite is installed and running on your PC. See the User Guide for Nokia PC Suite on the CD-ROM. If Nokia PC Suite is installed and running, you can use Nokia Get connected wizard available in Nokia PC Suite to connect to your PC. For further information on how to use Nokia PC Suite, see the help function on Nokia PC Suite or visit the support pages at **www.nokia.com**.

Access codes

Q: What is my password for the lock, PIN, or PUK codes?

A: The default lock code is **12345**. If you forget or lose the lock code, contact your phone dealer.

If you forget or lose a PIN or PUK code, or if you have not received such a code, contact your network service provider.

For information about passwords, contact your access point provider, for example, a commercial Internet service provider (ISP), service provider, or network operator.

Application not responding

Q: How do I close an application that is not responding?

A: Open the application switching window by pressing and holding . Then scroll to the application, and press to close the application.

Phone display

Q: Why do missing, discolored, or bright dots appear on the screen every time I turn on my phone?

A: This is a characteristic of this type of display. Some displays may contain pixels or dots that remain on or off. This is normal, not a fault.

Memory low

Q: What can I do if my phone memory is low?

A: You can delete the following items regularly to avoid low memory:

- Messages from **Inbox**, **Drafts**, and **Sent** folders in **Messag.**
- Retrieved e-mail messages from the phone memory
- Saved browser pages
- Images and photos in **Gallery**

To delete contact information, calendar notes, call timers, call cost timers, game scores, or any other data, go to the respective application to remove the data. If you are deleting multiple items and any of the following notes are shown: **Not enough memory to perform operation. Delete some data first.** or **Memory low. Delete some data.**, try deleting items one by one (starting from the smallest item).

Q: How can I save my data before deleting it?

A: Save your data using one of the following methods:

- Use Nokia PC Suite to make a back-up copy of all data to a compatible computer.
- Send images to your e-mail address, then save the images to your computer.
- Send data using a Bluetooth connection to a compatible device.
- Move data from the device memory to the hard drive of your device.

Product safety

Battery information

Charging and discharging

Your device is powered by a rechargeable battery. The full performance of a new battery is achieved only after two or three complete charge and discharge cycles. The battery can be charged and discharged hundreds of times but it will eventually wear out. When the talk and standby times are noticeably shorter than normal, replace the battery. Use only Nokia approved batteries, and recharge your battery only with Nokia approved chargers designated for this device.

Unplug the charger from the electrical plug and the device when not in use. Do not leave fully charged battery connected to a charger, since overcharging may shorten its lifetime. If left unused, a fully charged battery will lose its charge over time.

If the battery is completely discharged, it may take a few minutes before the charging indicator appears on the display or before any calls can be made.

Use the battery only for its intended purpose. Never use any charger or battery that is damaged.

Do not short-circuit the battery. Accidental short-circuiting can occur when a metallic object such as a coin, clip, or pen causes direct connection of the positive (+) and negative (-) terminals of the battery. (These look like metal strips on the battery.) This might happen, for example, when you carry a spare battery in your pocket or purse. Short-circuiting the terminals may damage the battery or the connecting object.

Leaving the battery in hot or cold places, such as in a closed car in summer or winter conditions, will reduce the capacity and lifetime of the battery. Always try to keep the battery between 15°C and 25°C (59°F and 77°F). A device with a hot or cold battery may not work temporarily, even when the battery is fully charged. Battery performance is particularly limited in temperatures well below freezing.

Do not dispose of batteries in a fire as they may explode. Batteries may also explode if damaged. Dispose of batteries according to local regulations. Please recycle when possible. Do not dispose as household waste.

Operation times

The BL-5C battery provides up to 3-4 hours of talk time and up to 7-8 days of standby time when operating on a GSM network. The BL-5C battery provides up to 2.5-3 hours of talk time and up to 7 days of standby time when operating on a WCDMA network. The BL-5C battery provides up to 10 hours of music or radio playback. Operation times are estimates and may vary depending on network conditions, charging, and device use. Charging the battery with the AC-4 charger takes up to 2 hours while the device is in standby mode.

Nokia battery authentication guidelines

Always use original Nokia batteries for your safety. To check that you are getting an original Nokia battery, purchase it from an authorized Nokia dealer, look for the Nokia Original Enhancements logo on the packaging and inspect the hologram label using the following steps:

Successful completion of the four steps is not a total assurance of the authenticity of the battery. If you have any reason to believe that your battery is not an authentic original Nokia battery, you should refrain from using it and take it to the nearest authorized Nokia service point or dealer for assistance. Your authorized Nokia service point or dealer will inspect the battery for authenticity. If authenticity cannot be verified, return the battery to the place of purchase.

Authenticate hologram

1 When looking at the hologram on the label, you should see the Nokia connecting hands symbol from one angle and the Nokia Original Enhancements logo when looking from another angle.

2 When you angle the hologram left, right, down, and up, you should see 1, 2, 3, and 4 dots on each side respectively.

3 Scratch the side of the label to reveal a 20-digit code, for example 12345678919876543210. Turn the battery so that the numbers are facing upwards. The 20-digit code reads starting from the number at the top row followed by the bottom row.

4 Confirm that the 20-digit code is valid by following the instructions at **www.nokia.com/batterycheck**.

What if your battery is not authentic?

If you cannot confirm that your Nokia battery with the hologram on the label is an authentic Nokia battery, please do not use the battery. Take it to the nearest authorized Nokia service point or dealer for assistance. The use of a battery that is not approved by the manufacturer may be dangerous and may result in poor performance and damage to your device and its enhancements. It may also invalidate any approval or warranty applying to the device.

To find out more about original Nokia batteries, visit **www.nokia.com/battery**.

Care and maintenance

Your device is a product of superior design and craftsmanship and should be treated with care. The suggestions below will help you protect your warranty coverage.

- Keep the device dry. Precipitation, humidity and all types of liquids or moisture can contain minerals that will corrode electronic circuits. If your device does get wet, remove the battery and allow the device to dry completely before replacing it.
- Do not use or store the device in dusty, dirty areas. Its moving parts and electronic components can be damaged.
- Do not store the device in hot areas. High temperatures can shorten the life of electronic devices, damage batteries, and warp or melt certain plastics.
- Do not store the device in cold areas. When the device returns to its normal temperature, moisture can form inside the device and damage electronic circuit boards.
- Do not attempt to open the device other than as instructed in this guide.
- Do not drop, knock, or shake the device. Rough handling can break internal circuit boards and fine mechanics.
- Do not use harsh chemicals, cleaning solvents, or strong detergents to clean the device.
- Do not paint the device. Paint can clog the moving parts and prevent proper operation.
- Use a soft, clean, dry cloth to clean any lenses (such as camera, proximity sensor, and light sensor lenses).
- Use only the supplied or an approved replacement antenna. Unauthorized antennas, modifications, or attachments could damage the device and may violate regulations governing radio devices.
- Use chargers indoors.
- Always create a backup of data you want to keep (such as contacts and calendar notes) before sending your device to a service facility.

All of the above suggestions apply equally to your device, battery, charger, or any enhancement. If any device is not working properly, take it to the nearest authorized service facility for service.

Additional safety information

Operating environment

Remember to follow any special regulations in force in any area and always switch off your device when its use is prohibited or when it may cause interference or danger. Use the device only in its normal operating positions. This device meets RF exposure guidelines when used either in the normal use position against the ear or when positioned at least 1.5 cm (5/8 in) away from the body. When a carry case, belt clip or holder is used for body-worn operation, it should not contain metal and should position the device the above-stated distance from your body.

In order to transmit data files or messages, this device requires a good quality connection to the network. In some cases, transmission of data files or messages may be delayed until such a connection is available. Ensure the above separation distance instructions are followed until the transmission is completed.

Parts of the device are magnetic. Metallic materials may be attracted to the device, and persons with a hearing aid should not hold the device to the ear with the hearing aid. Do not place credit cards or other magnetic storage media near the device, because information stored on them may be erased.

Medical devices

Operation of any radio transmitting equipment, including wireless phones, may interfere with the functionality of inadequately protected medical devices. Consult a physician or the manufacturer of the medical device to determine if they are adequately shielded from external RF energy or if you have any questions. Switch off your device in health care facilities when any regulations posted in these areas instruct you to do so. Hospitals or health care facilities may be using equipment that could be sensitive to external RF energy.

Pacemakers

Pacemaker manufacturers recommend that a minimum separation of 15.3 cm (6 in) be maintained between a wireless phone and a pacemaker to avoid potential interference with the pacemaker. These recommendations are consistent with the independent research by and recommendations of Wireless Technology Research. Persons with pacemakers should:

- always keep the device more than 15.3 cm (6 in) from their pacemaker;
- not carry the device in a breast pocket; and
- hold the device to the ear opposite the pacemaker to minimize the potential for interference.

If you suspect interference, switch off your device and move the device away.

Hearing aids
Some digital wireless devices may interfere with some hearing aids. If interference occurs, consult your service provider.

Vehicles

RF signals may affect improperly installed or inadequately shielded electronic systems in motor vehicles such as electronic fuel injection systems, electronic antiskid (antilock) braking systems, electronic speed control systems, air bag systems. For more information, check with the manufacturer or its representative of your vehicle or any equipment that has been added.

Only qualified personnel should service the device, or install the device in a vehicle. Faulty installation or service may be dangerous and may invalidate any warranty that may apply to the device. Check regularly that all wireless device equipment in your vehicle is mounted and operating properly. Do not store or carry flammable liquids, gases, or explosive materials in the same compartment as the device, its parts, or enhancements. For vehicles equipped with an air bag, remember that air bags inflate with great force. Do not place objects, including installed or portable wireless equipment in the area over the air bag or in the air bag deployment area. If in-vehicle wireless equipment is improperly installed and the air bag inflates, serious injury could result.

Using your device while flying in aircraft is prohibited. Switch off your device before boarding an aircraft. The use of wireless teledevices in an aircraft may be dangerous to the operation of the aircraft, disrupt the wireless telephone network, and may be illegal.

Potentially explosive environments

Switch off your device when in any area with a potentially explosive atmosphere and obey all signs and instructions. Potentially explosive atmospheres include areas where you would normally be advised to turn off your vehicle engine. Sparks in such areas could cause an explosion or fire resulting in bodily injury or even death. Switch off the device at refuelling points such as near gas pumps at service stations. Observe restrictions on the use of radio equipment in fuel depots, storage, and distribution areas, chemical plants or where blasting operations are in progress. Areas with a potentially explosive atmosphere are often but not always clearly marked. They include below deck on boats, chemical transfer or storage facilities, vehicles using liquefied petroleum gas (such as propane or butane), and areas where the air contains chemicals or particles such as grain, dust or metal powders.

Emergency calls

Important: Wireless phones, including this device, operate using radio signals, wireless networks, landline networks, and user-programmed functions. Because of this, connections in all conditions cannot be guaranteed. You should never rely solely on any wireless device for essential communications like medical emergencies.

To make an emergency call:

1 If the device is not on, switch it on. Check for adequate signal strength.
Some networks may require that a valid SIM card is properly inserted in the device.

2 Press as many times as needed to clear the display and ready the device for calls.

3 Enter the official emergency number for your present location. Emergency numbers vary by location.

4 Press the key.

When making an emergency call, give all the necessary information as accurately as possible. Your wireless device may be the only means of communication at the scene of an accident. Do not end the call until given permission to do so.

Warning: In the offline profile you cannot make (or receive) any calls, except make calls to certain emergency numbers, or use other features that require network coverage.

Certification information (SAR)

This mobile device meets guidelines for exposure to radio waves.

Your mobile device is a radio transmitter and receiver. It is designed not to exceed the limits for exposure to radio waves recommended by international guidelines. These guidelines were developed by the independent scientific organization ICNIRP and include safety margins designed to assure the protection of all persons, regardless of age and health.

The exposure guidelines for mobile devices employ a unit of measurement known as the Specific Absorption Rate or SAR. The SAR limit stated in the ICNIRP guidelines is 2.0 watts/kilogram (W/kg) averaged over 10 grams of tissue. Tests for SAR are conducted using standard operating positions with the device transmitting at its highest certified power level in all tested frequency bands. The actual SAR level of an operating device can be below the maximum value because the device is designed to use only the power required to reach the network. That amount changes depending on a number of factors such as how close you are to a network base station. The highest SAR value under the ICNIRP guidelines for use of the device at the ear is 0.48 W/kg.

Use of device accessories and enhancements may result in different SAR values. SAR values may vary depending on national reporting and testing requirements and the network band. Additional SAR information may be provided under product information at **www.nokia.com**.

Your mobile device is also designed to meet the requirements for exposure to radio waves established by the Federal Communications Commission (USA) and Industry Canada. These requirements set a SAR limit of 1.6 W/kg averaged over 1 gram of tissue. The highest SAR value reported under this standard during product certification for use at the ear is 0.74 W/kg and when properly worn on the body is 0.58 W/kg. Information about this device model can be found at **http://www.fcc.gov/oet/fccid** by searching the equipment authorization system using FCC ID: QEYRM-43.

Technical Information

Feature	Specification
Weight	164 grams (5.78 oz) with BL-5C Li-Ion Battery
Size	Volume: 6.7 in^3 (109 cm^3) Length: 4.45 in (113.1 mm) Width: 2.17 in (55.2 mm) Thickness: 0.87 in (22 mm)
Frequency range	GSM 900 880–915 MHz (TX) 925–960 MHz (RX) GSM 1800 1710–1785 MHz (TX) 1805–1880 MHz (RX) GSM 1900 1850–1910 MHz (TX) 1930–1990 MHz (RX)
Transmitter output power	Up to 2 W
Battery voltage	3.7 V dc
Operating temperature	14°F to + 131°F (-10°C to + 55°C)

Index

Numerics

1-touch dialing 33

A

access points 97
- creating 98
- settings 98

activation keys 109

alarms
- alarm clock 16
- calendar 79

anykey answer 97

applications
- installing 106
- Java 105

audio file formats 19

autolock 101

B

backing up 108

bitrate 28

Bluetooth 86
- connecting two devices, pairing 88
- device address 88
- headset 105
- pairing 88
- security 87
- switching off 89

bookmarks 81

brightness, display 96

C

cache, clearing 84

calendar
- alarm 78
- creating entries 78
- synchronizing PC Suite 79

call log 76

call register. See *log.*

calls
- call forward settings 103
- conference call 32
- dialed numbers 76
- duration 76
- international 32
- received 76
- restrict 104
- settings 96
- transferring 34
- using loudspeaker 17

camera
- adjust contrast 39
- adjust lighting & color 39, 40, 42
- image setup settings 40
- saving video clips 42
- self-timer 40
- sending images 39
- sequence mode 40
- still image camera settings 40
- video recorder settings 43

CD-ROM 90

certificates 102

chat. See *instant messaging.*

clear screen. See *standby mode.*

clearing memory
- log information 111

clock 16

codecs 28

codes 101, 102

computer connections 89

conference call 32

connection
- manager 90
- settings 97

contact cards
inserting pictures 36
sending 36
storing touch tones 35
copying
contacts 37
text 53
copyright protection. See *activation keys.*
cutting text 53

D

data connections
details 90, 91
ending 90
indicators 14
date 16
digital rights management. See *DRM.*
DNS 99
domain name service. See *DNS.*
DRM 109

E

e-mail 54
automatic retrieving 59
offline 60
opening 59
remote mailbox 58
retrieving from mailbox 58
settings 63
viewing attachments 59
enhancement settings 105

F

file formats
.jad 105
.jar 105, 106
.sis 105
music player 19
RealPlayer 45
file manager 17
fixed dialing 37
FM radio 48
forward calls 103

G

gallery
adding files to album 44
creating new album 44
viewing images & videos 44
GPRS. See *packet data connections.* 97

H

handsfree. See *loudspeaker.*
headset settings 105
help application 15

I

indicators 14
instant messaging
blocking 69
connecting to a server 67
groups 70
recording messages 69
sending instant messages 68
Internet access points (IAP). See *access points.*
Internet. See *Web.*
IP address 99

J

Java. See *applications.*

K

Kb/s 28

L

lock code 102
log 77
loopset settings 105
loudspeaker 17
voice tags 33
while listening to the radio 48

M

mailbox 58, 103
automatic retrieving 59

media
gallery 43
music player 44
RealPlayer 45
recording videos 42
taking pictures 39
media files
file formats 45
media files. See *file formats.*
memory
clearing memory 76
viewing memory consumption 12
memory card 108
messaging
e-mail 54
multimedia messages 54
text messages 54
multimedia messages 54
music
album art 27
details 27
file formats 19
transferring 22
music player
controls 19
equalizer 20
file formats
indicators 20
options 25
play modes 20
playlists 24
Music shop 21

P

packet data connections 97, 100
pasting text 53
PC connections 89
PC Suite
calendar data 79
synchronization 89
transferring images to a PC 43
transferring media files 12
viewing phone memory data 18
personalization 30
phonebook. See *contact cards.*
PIN code 101
profiles 29

R

radio 48
reminder. See *calendar, alarm.*
remote mailbox 58
remote SIM mode 87
ringing tones
personal ringing tone 38
receiving in a text message 57

S

security
autolock 101
lock code 102
phone and SIM 101
sending
contact cards, business cards 36
images 39
service commands 51
settings
access codes 101
access points 98
Bluetooth connection 86
calendar 79
call forwarding 103
certificates 102
data connections 97
date and time 100
display 96
headset 105
IM 67, 68, 71
language 95
lock code 102
loopset 105
original settings 95
personalizing the phone 29
PIN code 101
restrict calls 104

screen saver 96
UPIN code 101
UPUK code 102
writing language 95
shortcuts
Web connection 81
SIM card
copying contacts 37
messages 61
names and numbers 37
sis file 105
SMS. See *text messages.*
snooze 16
software 106
songs
file formats 19
sound clips 43
sounds
muting a ringing tone 34
recording sounds 47
speakerphone. See *loudspeaker.*
speed dialing. See *1-touch dialing.*
standby mode 29
synchronization 91

T

text messages 54
service center 62
themes 30
time 16
transferring files 106

U

UPIN code 101
UPUK code 102
USB cable 89
USSD commands 51

V

VBR 28
video clips 43
video player. See *RealPlayer.*
Visual Radio 48, 49
voice
commands 105
dialing 33
mailbox 32, 103
messages 32
tags 33, 105
volume control
during a call 32
loudspeaker 17

W

Web
browser 81
wireless LAN 13
access point settings 99

Notes

Nokia N91
Manual del Usuario

Importante: Para prevenir cualquier malfuncionamiento y evitar daños, lea detalladamente este manual de instrucciones antes de conectar y operar este aparato. Conserve el manual para referencias futuras.

DECLARACIÓN DE CONFORMIDAD
Nosotros, NOKIA CORPORATION, declaramos bajo nuestra única responsabilidad, que el producto RM-43 se adapta a las condiciones dispuestas en la Normativa de Consejo siguiente: 1999/5/EC. Existe una copia de la Declaración de Conformidad disponible en la dirección **http://www.nokia.com/phones/declaration_of_conformity/**.

CE 434 !

Este símbolo significa que dentro de la Unión Europea el producto debe llevarse a un local especial para su eliminación luego de finalizada su vida útil. Este reglamento se aplica tanto a su dispositivo como a cualquier accesorio marcado con este símbolo. No deseche estos productos junto con los demás desperdicios domésticos.

Copyright © 2006 Nokia. Todos los derechos reservados.

Se prohíbe la duplicación, desvío, distribución o almacenamiento parcial o de todo el contenido de este documento en cualquier forma sin previo permiso escrito por Nokia.

Nokia, Nokia Connecting People y Pop-Port son marcas, registradas o no, de Nokia Corporation. Otros nombres de productos y compañías aquí mencionados podrían ser marcas o nombres, registradas o no, de sus respectivos propietarios.

Nokia Tune es una marca de sonido registrada de Nokia Corporation.

symbian

Este producto incluye software con licencia de Symbian Ltd © 1998-200(6). Symbian y Symbian OS son marcas registradas de Symbian Ltd.

Java POWERED

Java™ y todas las marcas basadas en Java son marcas, registradas o no, de Sun Microsystems, Inc.

Microsoft, Windows y Media Player son marcas registradas o marcas comerciales de Microsoft Corporation en EE.UU y/u otros países.

Stac ®, LZS ®, ©1996, Stac, Inc., ©1994-1996 Microsoft Corporation. Incluye una o más patentes de los Estados Unidos: Nº 4701745, 5016009, 5126739, 5146221 y 5414425. Otras patentes pendientes.

Hi/fn ®, LZS ®,©1988-98, Hi/fn. Incluye una o más patentes de los Estados Unidos: Nº 4701745, 5016009, 5126739, 5146221 y 5414425. Otras patentes pendientes.

Parte del software en este producto es © Copyright ANT Ltd. 1998. Todos los derechos reservados.
Nº de Patente de EE.UU. 5818437 y otras patentes pendientes. Software Ingreso de Texto T9 Copyright (C) 1997-2006. Tegic Communications, Inc. Todos los derechos reservados.

This product is licensed under the MPEG-4 Patent Portfolio License (i) for personal and non-commercial use in connection with information which has been encoded in compliance with the MPEG-4 Visual Standard by a consumer engaged in a personal and non-commercial activity and (ii) for use in connection with MPEG-4 video provided by a licensed video provider. No license is granted or shall be implied for any other use. Additional information including that relating to promotional, internal and commercial uses may be obtained from MPEG LA, LLC. See **http:www.mpegla.com**.

Este producto está autorizado bajo la Licencia en Cartera de Patentes Visual MPEG-4 (i) para uso personal y no comercial en relación con la información que ha sido codificada de acuerdo con el Estándar Visual MPEG-4 por un consumidor que se dedica a una actividad personal y no comercial y (ii) para su uso en relación con el video MPEG-4 provisto por un proveedor autorizado de video. Ningún permiso es concedido o será implícito para cualquier otro uso. Más información, incluso a lo relacionado a los usos comerciales, internos y de promoción podrían obtenerse de MPEG LA, LLC. Consulte **http://www.mpegla.com**.

Nokia opera bajo un sistema de desarrollo continuo. Nokia se reserva el derecho de hacer cambios y mejoras en cualquier producto descrito en este manual sin previo aviso.

Bajo ningún concepto Nokia será responsable por cualquier pérdida de datos o ganancias o por cualquier otro daño especial o incidental, por cualquier causa directa o indirecta.

El contenido de este documento está provisto 'tal cual'. Además de las exigidas por las leyes, no se emite ninguna otra garantía de ningún tipo, sea expresa o implícita, incluyendo, pero sin limitarse a, garantías implícitas de mercadeo y de capacidad para un propósito particular, en cuanto a la precisión y fiabilidad o contenido de este documento. Nokia se reserva el derecho de revisar este documento o descontinuar su uso en cualquier momento sin previo aviso.

La disponibilidad de ciertos productos y servicios podrían variar por región. Consulte con su distribuidor local de Nokia.

CONTROLES DE EXPORTACIÓN

Este dispositivo podría contener bienes, tecnología o software sujeto a las leyes y regulaciones de exportación de los Estados Unidos y otros países. Se prohíbe su desvío contraviniendo las leyes.

AVISO FCC / INDUSTRIA CANADIENSE

Su dispositivo puede causar interferencias radioeléctricas o televisivas (por ejemplo, cuando usa su teléfono muy próximo a equipos receptores electrónicos). La FCC o Industria Canadiense puede exigirle que deje de usar su teléfono si tales interferencias no se pueden eliminar. Si necesita ayuda, póngase en contacto con su centro de servicio local. Este dispositivo cumple con la sección 15 de los reglamentos de la FCC. El funcionamiento de este dispositivo está sujeto a la condición de que no cause interferencia perjudicial.

Edición 1 EN, 9248320

Notas

Contenido

Para su seguridad 134

Su Nokia N91 137

Disco duro 137
Ver el consumo de memoria 137
Memoria baja: memoria disponible 137
LAN inalámbrica 138
Indicadores esenciales 140
Soporte Nokia en la Web 141
Ayuda 141
Tutorial 141
Reloj 142
Configuraciones del reloj 142
Reloj mundial 142
Control de volumen y altavoz 143
Administrador de archivos 143
Bloquear el teclado (Bloqueo de teclas) 144

Reproductor de música 145

Reproducir una canción 145
Controles 145
Indicadores 146
Modos de reproducción 146
Reproducir en orden aleatorio 146
Repetir reproducción 146
Modificar los sonidos 146
Ecualizador 146
Configuraciones de audio 147
Tienda de música 147
Ir a tienda de música 147
Configuraciones de tienda de música 147
Transferir música 148
Transferir de PC 148
Transferir música 149
Listas de reproducción 150
Crear una lista de reproducción 150
Agregar canciones a una lista de reproducción 150
Reorganizar las canciones 150
Eliminar canciones de una lista de reproducción 151
Agregar elementos en vistas a una lista de reproducción 151
Detalles de las listas de reproducción 151
Listas de reproducción automáticas 152

Opciones disponibles ... 152
Opciones sólo en la vista En reproducción ... 153
Renombrar elementos ... 153
Ver y editar detalles de canciones ... 153
Detalles de la biblioteca ... 153
Configurar una canción como el tono de timbre ... 154
Cambiar el arte del álbum ... 154
Sobre audio codificado ... 154
Velocidad de bits ... 154
Velocidad de bits variable ... 155

Personalizar su dispositivo ... 156

Modos: configurar tonos ... 156
Perfil desconectado ... 157
Cambiar la apariencia de su dispositivo ... 157
Standby activo ... 158

Realizar llamadas ... 159

Llamadas de voz ... 159
Realizar una llamada de multiconferencia ... 160
Marcación rápida de un número telefónico ... 160
Marcación por voz ... 160
Contestar o rechazar una llamada ... 161
Llamada en espera (servicio de red) ... 162
Opciones durante una llamada de voz ... 162

Contactos (Directorio) ... 163

Guardar nombres y números ... 163
Números y direcciones predeterminados ... 164
Directorio SIM y servicios SIM ... 164
Copiar contactos ... 165
Agregar tonos de timbre para contactos ... 165
Crear grupos de contactos ... 165
Eliminar miembros de un grupo ... 166

Aplicaciones multimedia ... 167

Cámara ... 167
Tomar fotos ... 167
Visor de imágenes ... 169
Sugerencias para tomar fotos ... 169
Grabar videos ... 170
Galería ... 172
Ver imágenes y videoclips ... 172
Carpetas ... 173
Reproductor de música ... 173
RealPlayer™ ... 173
Reproducir video ... 173
Contenido directo sobre el aire (streaming) ... 174
Recibir configuraciones de RealPlayer ... 174
FlashPlayer ... 175
Organizar archivos flash ... 175
Reproducir archivos flash ... 176

Grabadora 176
Radio 176
Escuchar la radio 177
Ver contenido visual 177
Estaciones guardadas 178
Configuraciones 178

Mensajería 179

Escribir texto 180
Ingreso de texto tradicional 180
Ingreso de texto predictivo: texto predictivo 181
Copiar texto en el portapapeles 182
Escribir y enviar mensajes 183
Recibir configuraciones MMS y de e-mail 185
Buzón de entrada: recibir mensajes 186
Mensajes multimedia 186
Datos y configuraciones 186
Mensajes de servicio Web 187
Mis carpetas 187
Buzón de correo 187
Abrir el buzón de correo 188
Recuperar mensajes de e-mail 188
Recuperar mensajes de e-mail automáticamente 189
Borrar mensajes de e-mail 189
Desconectarse del buzón de correo 190
Buzón de salida: mensajes a la espera de ser enviados 190
Ver mensajes en la tarjeta SIM 191
Configuraciones de mensajes 191
Mensajes de texto 191
Mensajes multimedia 192
E-mail 193
Mensajes de servicio Web 195
Difusión celular 195
Otros configuraciones 195

Personal 196

Web 196
Notas 196
Ir a: agregar accesos directos 196
Borrar un acceso directo 197
MI: mensajería instantánea (chat) 197
Recibir configuraciones de MI 197
Conectarse a un servidor MI 197
Modificar sus configuraciones de MI 198
Buscar grupos y usuarios MI 198
Incorporarse a grupos MI 198
Chat 198
Ver e iniciar conversaciones 199
Contactos MI 199
Administrar los grupos MI 200
Configuraciones de servidor MI 201

Push to talk (servicio de red) 201
Definir un punto de acceso PTT 202
Configuraciones PTT 202
Registrarse a PTT 203
Salir de PTT 203
Realizar una llamada individual 203
Contestar una llamada individual 204
Realizar llamadas de grupo 204
Suscribirse al estado de conexión de otros usuarios 204
Enviar una solicitud de devolución de llamada 204
Canales 204
Responder a una invitación de canal 205
Ver el registro PTT 206
Registro 206
Últimas llamadas 206
Duración de llamadas 206
Paquetes de datos 206
Monitorear eventos de comunicación 207

Agenda 208

Crear entradas de agenda 208
Programar una alarma de agenda 209
Vistas de la agenda 209
Vista de tareas 209
Borrar entradas de la agenda 210
Configuraciones de la agenda 210

Servicios 211

Punto de acceso para servicios 211
Vista Favoritos 211
Agregar favoritos manualmente 212
Enviar favoritos 212
Establecer una conexión 212
Seguridad de conexión 212
Explorar 213
Ver las páginas guardadas 214
Descargar y comprar elementos 214
Finalizar una conexión 215
Vaciar la memoria caché 215
Configuraciones para servicios 215

Conectividad 217

Conectividad Bluetooth 217
Configuraciones 218
Enviar datos usando la conectividad Bluetooth ... 219
Vincular dispositivos 220
Recibir datos usando la conectividad Bluetooth ... 220
Desactivar la conectividad Bluetooth 220
Cable de datos 221

Conexiones PC 221
CD-ROM 221
Su dispositivo como módem 221
Administrador de conexiones 222
Conexiones de datos 222
Ver los detalles de las conexiones de datos 222
Red local de área inalámbrica (WLAN) 223
LAN inalámbrica 223
Sincronización 223
Crear un perfil de sincronización nuevo 223
Sincronizar datos 224
Administrador de dispositivos 225
Configuraciones del perfil de servidor 225

Herramientas 226

Calculadora 226
Calcular porcentajes 226
Convertidor 226
Configurar una moneda base y tasas de cambio 227
Configuraciones 227
Configuraciones del dispositivo 227
Configuraciones de llamadas 228
Conexión 229
Fecha y hora 234
Seguridad 234
Desvío de llamadas 237
Restricción de llamadas 237
Red 238
Configuraciones de accesorios 238
Comandos de voz 239
Administrador de aplicaciones 239
Instalar aplicaciones y software 240
Eliminar aplicaciones y software 241
Configuraciones de aplicación 241
Memoria del disco duro 241
Herramientas 242
Manejo de derechos digitales 242

Solución de problemas 244

Preguntas y respuestas 244

Seguridad del producto 247

Información sobre las baterías 247
Carga y descarga 247
Tiempos de funcionamiento 248
Normas de autenticación de baterías Nokia 248
Cuidado y mantenimiento 249
Información adicional de seguridad 250
Información técnica 254

Índice 255

Para su seguridad

Lea estas sencillas normas. El incumplimiento de ellas puede ser peligroso o ilegal. Lea el manual del usuario completo para obtener más información.

ENCIENDA EL TELÉFONO EN FORMA SEGURA
No encienda el dispositivo donde el uso de teléfonos móviles esté prohibido o pueda causar interferencia o peligro.

LA SEGURIDAD VIAL ANTE TODO
Obedezca todas las leyes locales. Mantenga siempre las manos libres para maniobrar el vehículo mientras conduce. Su prioridad cuando conduce debe ser la seguridad vial.

INTERFERENCIAS
Todos los dispositivos móviles pueden ser susceptibles a interferencias que afectan su rendimiento.

APAGUE EL TELÉFONO EN LOS HOSPITALES
Respete todas las restricciones existentes. Apague el dispositivo cuando se encuentre cerca de equipo médico.

APAGUE EL TELÉFONO EN LOS AVIONES
Respete todas las restricciones existentes. Los dispositivos celulares pueden causar interferencias en los aviones.

APAGUE EL TELÉFONO CUANDO CARGUE COMBUSTIBLE
No utilice el dispositivo en las estaciones de servicio. Tampoco cerca de combustibles ni elementos químicos.

APAGUE EL TELÉFONO CERCA DE DONDE SE REALIZAN EXPLOSIONES
Respete todas las restricciones existentes. No utilice el dispositivo donde se estén realizando explosiones.

ÚSELO CORRECTAMENTE
Use el dispositivo sólo en la posición normal como se explica en la documentación del producto. Evite el contacto innecesario con la antena.

SERVICIO TÉCNICO CALIFICADO
Sólo personal de servicio técnico calificado puede instalar o reparar este producto.

ACCESORIOS Y BATERÍAS
Utilice sólo baterías y accesorios aprobados. No conecte productos incompatibles.

NO MOJE EL TELÉFONO
Su dispositivo no es resistente al agua. Manténgalo seco.

COPIAS DE SEGURIDAD
Recuerde hacer copias de seguridad o guardar un registro por escrito de toda la información importante.

CONEXIÓN CON OTROS DISPOSITIVOS.
Antes de conectar el teléfono con otro dispositivo, lea el manual del usuario para obtener instrucciones detalladas de seguridad.
No conecte productos incompatibles.

LLAMADAS DE EMERGENCIA
Asegúrese de que el teléfono esté encendido y funcionando. Pulse tantas veces como sea necesario para borrar la pantalla y regresar al modo standby. Ingrese el número de emergencia, luego pulse . Indique su ubicación. No finalice la llamada hasta que reciba instrucciones de hacerlo.

Acerca de su dispositivo

El dispositivo móvil descrito en este manual está aprobado para su uso en las redes EGSM 900/1800/1900 y UMTS 2100. Comuníquese con su proveedor de servicios para obtener más información sobre redes.

Cuando use las funciones de este dispositivo, obedezca todas las leyes y respete la privacidad y los derechos legítimos de los demás.

Las protecciones de copyright pueden impedir la copia, modificación, transferencia o desvío de imágenes, música (incluso tonos de timbre) y otro contenido.

Cuando tome y utilice imágenes o videoclips, obedezca todas las leyes y respete las costumbres locales y la privacidad y derechos legítimos de los demás.

Advertencia: Para poder usar cualquiera de las funciones en este dispositivo, con la excepción de la alarma, es preciso encender el dispositivo. No lo encienda donde el uso de dispositivos móviles pueda causar interferencia o peligro.

Servicios de red

Para utilizar el teléfono, debe contratar los servicios de un proveedor de servicios móviles. Muchas de las funciones de este dispositivo dependen de las funciones de la red inalámbrica para su funcionamiento. Es posible que estos servicios de red no estén disponibles en todas las redes o bien deberá hacer arreglos específicos con su proveedor de servicios antes de utilizar los servicios de red. Es posible que el proveedor de servicios le entregue instrucciones adicionales sobre cómo utilizarlos y le indique qué cargos son aplicables. Algunas redes pueden tener limitaciones que inciden en el uso de los servicios de red. Por ejemplo, es posible que algunas redes no admitan todos los servicios y caracteres de idiomas.

Su proveedor de servicios puede haber pedido que ciertas funciones de su dispositivo sean inhabilitadas o desactivadas. De ser así, éstas no aparecerán en el menú. Es posible que su dispositivo también haya sido configurado especialmente. Esta configuración puede incluir cambios en nombres de menús, orden de menús e iconos. Comuníquese con su proveedor de servicios para obtener más información.

Este dispositivo admite los protocolos WAP 2.0 (HTTP y SSL) que se ejecutan en los protocolos TCP/IP. Algunas funciones de este dispositivo, como MMS, explorador, e-mail y descarga de contenido utilizando el explorador o sobre MMS, requieren soporte de red para estas tecnologías.

Para obtener información sobre disponibilidad y uso de los servicios de tarjetas SIM, comuníquese con el proveedor de su tarjeta SIM. Este puede ser su proveedor de servicios, operador de red u otro proveedor.

Accesorios, baterías y cargadores

Siempre apague el dispositivo y desconecte el cargador antes de retirar la batería.

Quitar la batería antes de apagar el dispositivo puede dañar el disco duro.

Verifique el modelo del cargador antes de usarlo con este dispositivo. Este dispositivo está diseñado para su uso con un cargador DC-4, AC-3 y AC-4 y un cargador AC-1, ACP-7, ACP-8, ACP-9, ACP-12U, LCH-8, LCH-9, o LCH-12 cuando se usa el adaptador de cargador CA-44.

La batería diseñada para uso con este dispositivo es BL-5C.

Advertencia: Utilice sólo baterías, cargadores y accesorios aprobados por Nokia para este modelo particular. El uso de cualquier otro tipo de batería, cargador o accesorio puede anular toda aprobación o garantía y puede ser peligroso.

Para obtener más información sobre disponibilidad de accesorios aprobados, comuníquese con su proveedor. Cuando desconecte el cordón eléctrico de cualquier accesorio, sujete el enchufe, no el cordón.

Su Nokia N91

Número de modelo: Nokia N91-1.

En adelante, denominado Nokia N91.

Felicitaciones por la compra de su Nokia N91. Su dispositivo combina sonido estéreo, capacidad de almacenamiento de hasta 3000 canciones, descargas de música sobre el aire, acceso a su colección de música y funcionalidad smartphone con Wi-Fi y una cámara 2Mpix.

Disco duro

Su dispositivo está equipado con un disco duro de 4 Gb para proveer alta capacidad de almacenamiento para distintos tipos de archivos. Los sensores en el dispositivo detienen el disco duro si la temperatura de funcionamiento es muy baja o alta, si se cae el dispositivo, o si se retira la cubierta posterior. Las partes del disco duro son frágiles y deberá utilizarlo con cuidado. Todos los impactos causados por choques, vibraciones, golpes, caídas y el uso inadecuado del dispositivo, al igual que las temperaturas extremas, humedad y líquidos, pueden causar que el disco duro no funcione debidamente y la pérdida o corrupción de datos.

No retire la cubierta posterior cuando el dispositivo está encendido.

Ver el consumo de memoria

El dejar las aplicaciones funcionando en segundo plano aumenta la demanda de energía de la batería y reduce la vida útil de la batería. Mantenga pulsada para activar las aplicaciones abiertas y pulse **Opciones** > **Salir** para salir de una aplicación.

Para ver qué tipos de datos tiene en el dispositivo y cuánta memoria consumen los distintos tipos de datos, pulse y seleccione **Herramient.** > **Disco duro**. Desplácese a **Mem. libre** para ver la cantidad de memoria disponible en el dispositivo.

Memoria baja: memoria disponible

El dispositivo le notifica si hay poca memoria disponible. Si la memoria del dispositivo comienza a reducirse cuando tiene el explorador abierto, éste se cierra automáticamente para liberar espacio en la memoria.

Para liberar memoria del dispositivo, transfiera datos al disco duro en **Adm. arch.** Marque los archivos que moverá, seleccione **Opciones** > **Mover a carpeta** > **HARD DRIVE** y seleccione una carpeta.

Sugerencia: Para liberar memoria del dispositivo, use Image Store disponible en Nokia PC Suite para transferir archivos de sonido, imágenes o videoclips a una PC compatible. Consulte el CD-ROM suministrado en el paquete de ventas.

Para eliminar datos y liberar memoria, utilice **Adm. arch.** o desplácese a la aplicación respectiva. Por ejemplo, puede eliminar lo siguiente:

- Mensajes de las carpetas **Buzón entrada**, **Borrador** y **Enviados** en **Mensajes**
- Mensajes de e-mail recuperados de la memoria del teléfono
- Páginas Web guardadas
- Imágenes, videos o archivos de sonido guardados
- Información de contactos
- Notas de agenda
- Aplicaciones descargadas. También consulte "Administrador de aplicaciones", pág. 239
- Cualquier otra información que ya no necesita

LAN inalámbrica

Su dispositivo está aprobado para uso en una red de área local inalámbrica (LAN inalámbrica). Con LAN inalámbrica, puede conectar su dispositivo a la Internet.

Importante: Siempre habilite uno de los métodos de encriptación disponibles para aumentar la seguridad de su conexión de LAN inalámbrica. El uso de encriptación reduce el riesgo de acceso no autorizado a sus datos.

Algunos lugares, como Francia, tienen restricciones en el uso de LAN inalámbrica. Para obtener más información, consulte a sus autoridades locales.

Las funciones que utilizan LAN inalámbrica o que permiten su uso en segundo plano mientras se utilizan otras funciones, aumentan la demanda de alimentación de la batería y reducen la vida útil de la batería.

Su dispositivo admite las siguientes funciones de LAN inalámbrica:

- Estándar IEEE 802.11g
- Funcionamiento a una frecuencia de 2,4 GHz
- Métodos de encriptación equivalentes a privacidad alámbrica (WEP) con claves de hasta 128 bits, acceso protegido Wi-Fi (WPA) y 802,1x. Sólo se pueden usar estas funciones si son admitidas por la red.

Puede crear un punto de acceso a Internet (IAP) en una LAN inalámbrica y usarlo para aplicaciones que necesitan conectarse a la Internet.

Crear un punto de acceso a Internet

1 Pulse y seleccione **Conect.** > **Adm. con.** > **WLAN dispon.**

2 El dispositivo busca LANs inalámbricas dentro del área de alcance. Para crear un punto de acceso a Internet en una red, seleccione **Opciones** > **Definir pto. acceso**.

3 El dispositivo crea un punto de acceso a Internet con configuraciones predeterminadas. Para ver o cambiar las configuraciones del punto de acceso a Internet, consulte, "Puntos de acceso", pág. 230.

Cuando una aplicación le solicita que seleccione un punto de acceso, seleccione el punto de acceso creado o **Buscar WLAN** para buscar LANs inalámbricas dentro del área de alcance.

Una conexión de LAN inalámbrica es establecida cuando crea una conexión de datos usando un punto de acceso a Internet de una LAN inalámbrica. La conexión activa de LAN inalámbrica termina cuando finaliza la conexión de datos. Para terminar la conexión, consulte "Administrador de conexiones", pág. 222.

Puede usar LAN inalámbrica durante una llamada de voz o cuando tiene paquetes de datos activos. Sólo puede estar conectado a un dispositivo de punto de acceso de LAN inalámbrica a la vez, pero varias aplicaciones pueden usar la misma conexión de LAN inalámbrica.

Cuando activa el perfil **Desconectado**, todavía puede usar LAN inalámbrica (si está disponible). Recuerde cumplir con cualquier requisito de seguridad aplicable al establecer y usar una conexión de LAN inalámbrica.

Hay dos modos de operación en LAN inalámbrica: **Infraestructura** y **Ad-hoc**.

El modo de operación de infraestructura le permite dos tipos de comunicación: los dispositivos inalámbricos están conectados entre sí a través de un dispositivo de punto de acceso de LAN inalámbrica o los dispositivos inalámbricos están conectados a una LAN alámbrica mediante un dispositivo de punto de acceso de LAN inalámbrica.

En el modo de operación ad hoc, los dispositivos pueden enviar y recibir datos directamente entre sí. Para crear un punto de acceso a Internet para la red ad hoc, consulte "Puntos de acceso", pág. 230.

Sugerencia: Para verificar la dirección de identificación Media Access Control (MAC) que identifica su dispositivo, ingrese *#62209526# en el modo standby.

Indicadores esenciales

El dispositivo está en uso en una red GSM.

3G (servicio de red): el teléfono está en uso en una red UMTS. Consulte "Red", pág. 238.

tiene uno o más mensajes no leídos en la carpeta **Buzón entrada** en **Mensajes**.

ha recibido nuevo e-mail en el buzón de correo remoto.

hay mensajes a la espera de ser enviados en la carpeta **Buzón salida**. Consulte "Buzón de salida: mensajes a la espera de ser enviados", pág. 190.

tiene llamadas perdidas. Consulte "Llamadas de voz", pág. 159.

aparece si **Tipo de timbre** está configurado en **Silencio** y **Tono aviso mensaje**, **Tono aviso MI** y **Tono de alerta e-mail** están configurados en **Desactivado**. Consulte "Modos: configurar tonos", pág. 156.

el teclado del teléfono está bloqueado. Consulte "Bloquear el teclado (Bloqueo de teclas)", pág. 144.

tiene una alarma del reloj activada. Consulte "Reloj", pág. 142.

2 la segunda línea telefónica está en uso. Consulte "Configuraciones de llamadas", pág. 228.

todas las llamadas al teléfono serán desviadas a otro número. Si tiene dos líneas telefónicas, el indicador para la primera línea es 1 y para la segunda 2.

tiene un auricular conectado al teléfono.

se ha perdido la conexión a un auricular con conectividad Bluetooth.

D una llamada de datos está activa.

una conexión de paquetes de datos GPRS o EDGE está disponible.

una conexión de paquetes de datos GPRS o EDGE está activa.

una conexión de paquetes de datos GPRS o EDGE está en espera. Estos iconos aparecen en lugar del indicador de la intensidad de la señal (aparece en la esquina superior izquierda en el modo standby). Consulte "Conexiones de paquetes de datos en las redes GSM y UMTS", pág. 230.

una conexión de paquetes de datos UMTS está disponible.

una conexión de paquetes de datos UMTS está activa.

una conexión de paquetes de datos UMTS está en espera.

la conectividad Bluetooth está en **Activado**.

se están transmitiendo datos usando una conexión Bluetooth. Consulte "Conectividad Bluetooth", pág. 217.

ha configurado el dispositivo para buscar LANs inalámbricas y una LAN inalámbrica está disponible. Consulte "Red local de área inalámbrica (WLAN)", pág. 223.

una conexión de LAN inalámbrica está activa en una red que tiene encriptación.

una conexión de LAN inalámbrica está activa en una red que no tiene encriptación.

una conexión USB está activa.

tiene un mensaje de e-mail leído en espera de ser cargado al teléfono.

tiene un mensaje de e-mail que no ha sido leído en espera de ser cargado al teléfono.

Soporte Nokia en la Web

Consulte **http://www.nokia.com/support** para EE.UU y **http://latinoamerica2.ext.nokia.com/mitelefono** para Latinoamérica o su página local de Nokia para obtener la versión más reciente de este manual, información adicional, descargas y servicios asociados con su producto Nokia.

Ayuda

Su dispositivo cuenta con ayuda contextual. Puede acceder a la ayuda desde una aplicación o desde el menú principal.

Cuando una aplicación esté abierta, para acceder a la ayuda desde la vista actual, seleccione **Opciones** > **Ayuda**.

Cuando esté leyendo las instrucciones, para cambiar entre la ayuda y la aplicación que está abierta en segundo plano, mantenga pulsada .

Para acceder a la ayuda desde el menú principal, seleccione **Herramient.** > **Ayuda**. Seleccione la aplicación que desea para ver los temas de ayuda. Para buscar temas de ayuda usando palabras claves, seleccione **Opciones** > **Buscar palab. clave**.

Cuando esté leyendo un tema de ayuda, para ver otros temas asociados, desplácese a la izquierda o derecha con el joystick.

Tutorial

El tutorial le da información sobre algunas de las funciones de su dispositivo. Para acceder al tutorial en el menú, pulse y seleccione **Tutorial** y la sección que desea ver.

Reloj

Pulse y seleccione **Reloj**.

Para configurar una nueva alarma, seleccione **Opciones** > **Fijar alarma**. Ingrese la hora de alarma y seleccione **Aceptar**. Cuando la alarma está activa, aparece el símbolo .

Para apagar la alarma, seleccione **Parar**. Cuando suene el tono de alarma, pulse cualquier tecla o seleccione **Pausa** para detener la alarma por cinco minutos, tras los cuales volverá a sonar. Puede hacer esto hasta un máximo de cinco veces.

Si en la hora programada para la alarma el dispositivo está apagado, éste se enciende automáticamente y empieza a emitir el tono de alarma. Si selecciona **Parar**, el dispositivo le pregunta si desea activarlo para las llamadas. Pulse **No** para apagar el dispositivo o **Sí** para realizar y recibir llamadas. No seleccione **Sí** cuando el uso del teléfono móvil pueda causar interferencia o peligro.

Para desactivar una alarma, seleccione **Reloj** > **Opciones** > **Eliminar alarma**.

Configuraciones del reloj

Para cambiar las configuraciones del reloj, seleccione **Reloj** > **Opciones** > **Configuraciones**.

Para cambiar el reloj que aparece en el modo standby, desplácese hacia abajo y seleccione **Tipo de reloj** > **Analógico** o **Digital**.

Para permitir que la red del teléfono móvil actualice la información de hora, fecha y zona horaria en su teléfono (servicio de red), desplácese hacia abajo y seleccione **Hora de operador de red**. Para que la configuración **Hora de operador de red** quede activada, el teléfono se reinicia.

Para cambiar el tono de la alarma, desplácese hacia abajo y seleccione **Tono de alarma reloj**.

Reloj mundial

Abra **Reloj** y desplácese a la derecha para abrir la vista de reloj mundial. En la vista del reloj mundial, puede ver las horas en diferentes ciudades.

Para agregar ciudades a la lista, seleccione **Opciones** > **Agregar ciudad**. Ingrese las primeras letras del nombre de la ciudad. El recuadro de búsqueda aparece automáticamente y los equivalentes a las ciudades aparecen. Seleccione una ciudad. Podrá agregar un máximo de hasta quince ciudades en la lista.

Para configurar su ciudad actual, desplácese a la ciudad y seleccione **Opciones** > **Mi ciudad actual**. La ciudad aparece en la vista principal del reloj y la hora en su teléfono cambia de acuerdo a la ciudad seleccionada. Asegúrese que la hora esté correcta y que corresponda a su zona horaria.

Control de volumen y altavoz

Para subir o bajar el nivel del volumen, cuando tiene una llamada activa o está escuchando un sonido, mantenga pulsada o en el lado izquierdo del dispositivo.

El altavoz incorporado le permite hablar y escuchar el teléfono a corta distancia sin tener que sujetar el teléfono cerca de su oído, por ejemplo, colocarlo cerca en una mesa. Las aplicaciones de sonido usan el altavoz en forma predeterminada.

Advertencia: Cuando use el altavoz, no acerque el dispositivo al oído, ya que el volumen podría estar demasiado alto.

Para localizar el altavoz, consulte 'Teclas y partes' en la guía rápida.

Para usar el altavoz durante una llamada, inicie la llamada y seleccione **Opciones** > **Activar altavoz**.

Para apagar el altavoz cuando tiene una llamada activa o está escuchando algún sonido, seleccione **Opciones** > **Activar teléfono**.

Administrador de archivos

Muchas funciones del teléfono utilizan memoria para guardar información. Estas funciones incluyen contactos, mensajes, imágenes, tonos de timbre, notas de tareas y agenda, documentos y aplicaciones descargadas. La memoria disponible depende de cuánta información exista ya guardada en la memoria del teléfono.

Para explorar los archivos y carpetas en la memoria del dispositivo o el disco duro, pulse y seleccione **Herramient.** > **Adm. arch.**. Se abre la vista de la memoria del dispositivo. Mueva el joystick a la derecha para ver las carpetas.

Para mover o copiar archivos a una carpeta, pulse y pulse el joystick al mismo tiempo para marcar un archivo y seleccione **Opciones** > **Mover a carpeta** o **Copiar a carpeta**.

Puede abrir distintos formatos de archivo directamente desde Administrador de archivos. Sin embargo, para obtener la funcionalidad completa debe abrir el archivo en la aplicación correspondiente, no en Administrador de archivos.

Iconos en **Adm. arch.**:

Carpeta

Carpeta con subcarpeta

Para encontrar un archivo, seleccione **Opciones** > **Buscar**, seleccione la ubicación de búsqueda e ingrese el texto que corresponde al nombre del archivo.

Sugerencia: Puede usar Nokia Phone Browser disponible en Nokia PC Suite para ver el contenido de la memoria del dispositivo y disco duro de su teléfono. Consulte el CD-ROM en el paquete de ventas.

Bloquear el teclado (Bloqueo de teclas)

Para bloquear: mueva el botón de bloqueo, localizado en la parte superior del dispositivo junto a la tecla encender/apagar, a la izquierda. Cuando mueva el botón con la tapa cerrada, todas las teclas quedan bloqueadas. Cuando abre la tapa, el teclado numérico queda activado.

Para desbloquear: mueva el botón a la derecha.

Cuando el bloqueo del teclado está activado, pueden efectuarse llamadas al número de emergencia oficial programado en su dispositivo. Ingrese el número de emergencia y pulse .

Reproductor de música

Reproductor de música admite los archivos con extensiones AAC, AAC+, eAAC+, MP3, WMA, WAV, 3G2, 3GP, 3GPP, 3GPP2, MPEG, MP4, M4A, DCF, ODF, ASF y M3U. **Reproductor de música** no admite necesariamente todas las funciones de un formato de archivo ni todas las variaciones de los formatos de archivo.

Para obtener más información sobre el reproductor, seleccione **Opciones** > **Ayuda** para acceder a la ayuda contextual.

Las protecciones de copyright pueden impedir la copia, modificación, transferencia o desvío de imágenes, música (incluso tonos de timbre) y otro contenido.

Reproducir una canción

1 Pulse y seleccione **Música**.

2 Mueva el joystick para desplazarse hacia una canción preferida, artista u otra categoría y pulse el joystick para abrir las listas de categorías.

3 Pulse la tecla Reproducir/Pausa ▶II para reproducir la canción o lista de canciones seleccionada.

Para abrir la vista **En reproducción** del reproductor desde cualquier otra vista, pulse la tecla música . Para regresar a la vista anterior, pulse la tecla otra vez.

Para seleccionar más música para reproducirla en la vista **En reproducción**, seleccione **Opciones** > **Ir a menú Música**.

Controles

Los indicadores del visor de imágenes muestran lo siguiente:

⏭ pulse para ir la próxima canción. Para avance rápido, manténgala pulsada.

⏮ pulse una vez para regresar al comienzo de la canción. Para ir a la canción anterior, pulse en dos segundos tras comenzar la canción. Para rebobinar la canción, manténgala pulsada.

pulse para bajar el volumen. Está localizada en el lado izquierdo del dispositivo.

pulse para subir el volumen. Está localizada en el lado izquierdo del dispositivo.

Para bloquear todas las teclas del dispositivo, con la excepción de las teclas de volumen y encender/apagar, deslice el botón de bloqueo en la parte superior del dispositivo.

Sugerencia: También puede usar las teclas en el control remoto del dispositivo. El botón de bloqueo en el lado del control remoto sólo bloquea las teclas del control remoto.

Indicadores

Iconos en **Música**:

repetir una

repetir todas

orden aleatorio activado

altavoz silenciado (aparecen en el cuadro emergente de volumen)

Modos de reproducción

Reproducir en orden aleatorio

1 Para activar o desactivar la reproducción en orden aleatorio, seleccione **Opciones** > **Mezclar**.

2 Seleccione **Desactivar** o **Activar**. La vista **En reproducción** aparece con o sin el icono de orden aleatorio en la esquina superior derecha.

Repetir reproducción

1 Para repetir la canción actual, todas las canciones, o desactivar la función de repetir, seleccione **Opciones** > **Repetir**.

2 Seleccione **Desactivar**, **Todo**, o **Una**. La vista **En reprod.** aparece con o sin el icono de repetir todas las canciones o repetir canción en la esquina superior derecha.

Modificar los sonidos

Ecualizador

Para modificar el tono de la reproducción de música, seleccione **Opciones** > **Ecualizador**.

Para usar una de las configuraciones predeterminadas del ecualizador, selecciónela con el joystick y seleccione **Activar**. Cada valor predeterminado tiene diferentes configuraciones en un ecualizador gráfico de ocho bandas.

Modificar los valores predeterminados del ecualizador

1 Para crear un valor predeterminado nuevo, seleccione **Opciones**> **Nuevo favorito**. Para modificar un valor predeterminado existente, seleccione **Opciones** > **Editar**.

2 Para ajustar las configuraciones de frecuencias, mueva el joystick a la izquierda o derecha para seleccionar la frecuencia baja, media o alta y hacia arriba o abajo para subir o bajar el volumen para la banda de frecuencia seleccionada.

3 Para restaurar las configuraciones a sus valores originales, seleccione **Opciones** > **Restaurar a predet.**

Para borrar un valor predeterminado, desplácese a ésta y pulse C. Los valores predeterminados originales no se pueden borrar.

Configuraciones de audio

Para ajustar el balance, cambiar la imagen del estéreo o amplificar el bajo, seleccione **Opciones** > **Configuraciones audio** y alguna de las opciones.

Tienda de música

En la tienda de música puede buscar, explorar y comprar música para descargarla en su dispositivo. La variedad, disponibilidad y apariencia de los servicios de tienda de música pueden variar.

Ir a tienda de música

En la tienda de música puede buscar, explorar y comprar música para descargarla en su dispositivo. La disponibilidad y apariencia de esta opción pueden variar.

1. En **Menú Música**, seleccione **Opciones** > **Ir a Tienda de música**. (Está opción no está disponible si la dirección Web del servicio de música no ha sido definida. Consulte "Configuraciones de tienda de música", pág. 147.)
2. Seleccione alguna de las opciones en la pantalla para buscar, explorar o seleccionar tonos de timbre.

Configuraciones de tienda de música

Es posible que las configuraciones también estén predeterminadas y no las pueda editar.

Deberá tener conexiones a Internet válidas para poder usar este servicio. Para obtener más información, consulte "Puntos de acceso", pág. 230.

La disponibilidad y apariencia de las configuraciones de la tienda de música pueden variar.

También debe tener la dirección Web de la tienda de música en las configuraciones tienda de música.

Para cambiar o definir la dirección Web para la tienda de música, en el menú Música, seleccione **Opciones** > **Ir a Tienda de música**. (Tienda de música también puede estar disponible como un acceso directo por separado en la carpeta Operador.)

En tienda de música, seleccione **Opciones** > **Configuraciones**. Defina lo siguiente:

Dirección: ingrese la dirección Web del servicio tienda de música.

Punto acceso predete...: seleccione si se conectará automáticamente o recibirá una pregunta antes de conectarse. Sólo está disponible con WLAN.

Nombre de usuario: ingrese su nombre de usuario para la tienda de música. Si deja en blanco este campo, es posible que tenga que ingresar su nombre de usuario al registrarse.

Contraseña: ingrese su contraseña para la tienda de música. Si deja en blanco este campo, es posible que tenga que ingresar su contraseña al registrarse.

Transferir música

Puede transferir música de una PC compatible u otros dispositivos compatibles usando una conexión por cable USB o Bluetooth. Para obtener más información sobre la conexión a su dispositivo, consulte "Conectividad", pág. 217.

Los requisitos de la PC:

- Sistema operativo Windows® XP (o posterior)
- una versión compatible de la aplicación de reproductor Windows Media®. Puede obtener más información sobre la compatibilidad del reproductor Windows Media® de la sección N91 del sitio Web de Nokia.
- Nokia PC Suite 6.7 o una versión posterior. El software está disponible para la instalación en el CD-ROM en el paquete de ventas.

Método de conexión:

- Cable de datos USB DKE-2 (miniUSB - USB)
- PC compatible que tenga funcionalidad Bluetooth
- Red WLAN activa

Transferir de PC

Puede usar tres diferentes métodos para transferir archivos:

- Para ver su dispositivo en PC como un disco duro externo al cual puede transferir cualquier archivo de datos, establezca la conexión con el cable, conectividad Bluetooth o WLAN. Si está usando el cable USB DKE-2, seleccione **Transferencia de datos** como el método de conexión. Este método le da control total de los elementos que mueve y a dónde, pero no puede ver toda la información de archivo. Es posible que no pueda transferir material con restricciones de copia. Para actualizar la biblioteca tras actualizar la selección de canciones en su dispositivo, desplácese a **Menú Música** y seleccione **Opciones** > **Actualizar Bibliot. mus.**
- Para sincronizar música con el reproductor Windows Media, conecte el cable USB y selesccione **Reproductor de música** como el método de conexión. Este método es ideal si desea sincronizar todos o algunos archivos de música entre una PC compatible y su dispositivo.
- Para usar Nokia Audio Manager en Nokia PC Suite, conecte el cable USB DKE-2 y seleccione **PC Suite** como el método de conexión.

Conexión por cable USB

Conecte el cable USB DKE-2 a una PC compatible y a su dispositivo. Si está usando Windows XP, su dispositivo es reconocido automáticamente como un disco duro externo.

Todas las aplicaciones en su dispositivo son cerradas y el teléfono queda en el modo desconectado.

Para seleccionar el modo de conexión USB, use el joystick.

Para cambiar el modo de conexión USB predeterminado, pulse , seleccione **Conect.** > **Cable dat.** y alguna de las opciones en **Modo de cable de datos**.

Transferir música

Las siguientes instrucciones han sido escritas según la información sobre la versión compatible del reproductor Windows Media® más reciente cuando se escribió este manual. Sin embargo, los detalles de las instrucciones aplicables específicamente a las ubicaciones y nombres de distintas opciones pueden no ser precisas y estar sujetas a cambios. Para obtener información más actualizada consulte los manuales y ayudas actualizadas correspondientes a Windows Media Player.

Sincronización manual

Tras conectar su dispositivo con una PC compatible, el reproductor Windows Media selecciona la sincronización manual, si hay menos de 4 GB de espacio libre en su dispositivo o si no hay suficiente almacenamiento para toda su biblioteca. Con la sincronización manual, usted selecciona las canciones y listas de reproducción que desea mover, copiar o borrar.

Ingrese un nombre para usar como el nombre de su dispositivo en reproductor Windows Media.

Para transferir la selección manualmente:

1. Tras conectar su dispositivo con Windows Media Player, seleccione su dispositivo en el panel de navegación, si hay más de un dispositivo conectado.
2. Arrastre y suelte las canciones o álbumes en el panel de lista para la sincronización. Para eliminar todas las canciones o álbumes, seleccione un elemento en la lista y pulse **Remove from list**.
3. Verifique que el panel de lista contenga los archivos que desea sincronizar y que tiene suficiente espacio en el dispositivo. Pulse **Start Sync.** para iniciar la sincronización.

Sincronización automática

Para cambiar la opción de transferencia predeterminada en reproductor Windows Media, haga clic en la flecha bajo **Sync**, seleccione su dispositivo y pulse **Set up Sync**. Borre o seleccione el recuadro **Sync this device automatically**.

Si el recuadro **Sync this device automatically** está seleccionado y conecta su dispositivo, su biblioteca de música N91 es actualizada automáticamente tomando como base las listas de pistas seleccionadas en reproductor Windows Media.

Si no hay listas de reproducción seleccionadas, se utiliza toda la biblioteca de música de PC para la sincronización. Tenga en cuenta que la biblioteca de su PC puede contener más archivos de los que pueden caber en el disco duro de su Nokia N91. Consulte la ayuda del reproductor Windows Media para obtener más información.

Las listas de reproducción en su dispositivo no son sincronizadas con las listas de reproducción en el reproductor Windows Media.

Listas de reproducción

Crear una lista de reproducción

Para crear una lista de reproducción de las canciones:

1 Seleccione **Opciones** > **Crear lista reprod.** en la vista **Listas reprod.**

2 Ingrese un nombre para la lista de reproducción y seleccione **Aceptar** o pulse el joystick.

3 Abra y cierre el título de artista para encontrar las canciones de desea incluir en la lista de reproducción. Pulse el joystick para agregar elementos. Para mostrar la lista de canciones bajo título de artista, mueva el joystick a la derecha. Para ocultar la lista de canciones, mueva el joystick a la izquierda.

4 Cuando haya terminado de seleccionar, seleccione **Hecho**.

Agregar canciones a una lista de reproducción

Puede agregar canciones a una lista de reproducción existente o crear una lista de reproducción nueva.

1 En una lista de canciones, seleccione **Opciones** > **Agregar canciones**.

2 Abra y cierre el título de artista para encontrar las canciones que desea incluir en la lista de reproducción. Pulse el joystick para agregar elementos. Para mostrar la lista de canciones bajo título de artista, mueva el joystick a la derecha. Para ocultar la lista de canciones, mueva el joystick a la izquierda.

3 Cuando haya terminado de seleccionar, seleccione **Hecho**.

Sugerencia: Puede agregar canciones a una lista de reproducción que está reproduciendo en el momento. Para buscar una canción ingrese las letras con el teclado.

Reorganizar las canciones

1 Seleccione **Opciones** > **Reordenar lista** en la vista de lista de reproducción.

2 La canción seleccionada está lista para ser transferida. Desplácese hacia arriba o abajo para moverla o dejarla en su lugar.

3 Seleccione **Excluir**.

4 Desplácese para seleccionar la canción que desea mover.

5 Seleccione **Incluir**.

6 Desplácese a una posición nueva.

7 Seleccione **Excluir**.

8 Cuando haya reorganizado la lista, seleccione **Hecho**.

Sugerencia: Para arrastrar y soltar, también puede pulsar el joystick.

Eliminar canciones de una lista de reproducción

Puede eliminar canciones de una lista de reproducción.

1 Seleccione una canción en una lista de reproducción.

2 Seleccione **Opciones** > **Elim. de lista reprod.**

3 Seleccione **Sí** para confirmar. Este paso no borra la canción del dispositivo, sólo la elimina de la lista de reproducción.

Agregar elementos en vistas a una lista de reproducción

Puede agregar canciones, álbumes, artistas, géneros y compositores de distintas vistas del **Menú Música** a listas de reproducción existentes o nuevas.

1 En una vista abierta desde el **Menú Música**, seleccione un elemento.

2 Seleccione **Opciones** > **Agregar lista repr.** > **Lista reprod. guardada** o **Nueva lista reprod.**

3 Si seleccionó **Nueva lista reprod.**, ingrese un nombre para la lista de reproducción y seleccione **Aceptar**. Si seleccionó **Lista reprod. guardada**, seleccione la lista de reproducción y **Aceptar**.

Sugerencia: En algunas vistas puede seleccionar canciones múltiples para agregarlas al seleccionar **Opciones** > **Marcar**.

Sugerencia: Puede agregar a una lista de reproducción una canción que está escuchando en el momento.

Detalles de las listas de reproducción

En la vista **Listas de reproducción**, seleccione **Opciones** > **Detalles lista repr.** Aparece la siguiente información:

Nombre: el nombre de la lista de reproducción

Canción.: la cantidad actual de canciones en la lista de reproducción

Ubicación: la ubicación de la lista de reproducción en el disco duro

Fecha: la fecha más reciente en que la lista de reproducción fue modificada

Hora: la hora en que la lista de reproducción fue modificada

Listas de reproducción automáticas

Las siguientes listas de reproducción aparecen automáticamente en la vista de listas de reproducción:

Pistas más tocadas: incluye las cuarenta canciones más reproducidas, la lista comienza con las canciones reproducidas más recientemente

Pistas recientes: incluye las cuarenta canciones reproducidas recientemente en orden inverso, terminando con la última canción reproducida.

Últimas adiciones: incluye canciones que agregó a la biblioteca durante la última semana.

Opciones disponibles

Los siguientes elementos están disponibles como opciones en distintas vistas del reproductor de música.

Ir a En reproducción: abre una lista que muestra la información sobre la canción que está reproduciendo en el momento

Ir a Tienda de música: abre una conexión inalámbrica a una tienda Web, en la cual puede buscar, explorar y comprar música

Actualizar Bibliot. mus.: actualiza la lista de la biblioteca al escanear el disco duro en busca de canciones nuevas y eliminar enlaces rotos

Detalles de canción/**Detalles lista repr.**/**Detalles Bibl. musical**: muestra la información sobre el elemento seleccionado

Ayuda: abre la aplicación de ayuda

Crear lista reprod.: abre una vista en la cual puede crear una lista de reproducción nueva

Enviar: le permite enviar el elemento seleccionado a otro dispositivo compatible

Eliminar: elimina el elemento seleccionado tras confirmarlo. La canción se borra permanentemente del disco duro.

Renombrar: cambia el nombre del elemento seleccionado

Agregar lista repr.: agrega el elemento seleccionado a un lista de reproducción existente o nueva

Arte del álbum: abre una vista que muestra la carátula actual del álbum seleccionado. Puede cambiar o agregar una carátula usando las imágenes en su teléfono.

Marcar/Desmarcar: le permite seleccionar varios elementos para ejecutar acciones

Como tono de timbre: selecciona la canción seleccionada como el tono de timbre del perfil activo. Si selecciona canciones del disco duro para usarlas como tonos de timbre es posible que no funcionen con la vibración. Para rendimiento óptimo, guarde los tonos de timbre en la memoria del dispositivo.

Reordenar lista: reorganiza las canciones en una lista de reproducción.

Opciones sólo en la vista En reproducción

Ir a menú Música: abre la vista **Menú Música**

Mezclar: activa o desactiva el modo de orden aleatorio. En este modo las canciones se reproducen en orden aleatorio.

Repetir: repite la canción o todas las canciones en la lista que se está reproduciendo en el momento.

Ecualizador: abre una vista en la cual puede ajustar el sonido y el balance de frecuencia.

Configuraciones audio: abre una vista en la cual puede cambiar el balance del sonido, imagen estéreo y respuesta del bajo.

Renombrar elementos

1. Seleccione un **Artistas**, **Álbumes**, **Géneros** o **Compositores** de la lista.
2. Seleccione **Opciones** > **Renombrar**.
3. Ingrese el nuevo nombre y seleccione **Aceptar**. Todas las canciones que pertenecen a un artista, álbum, género o compositor seleccionado son actualizadas.

Ver y editar detalles de canciones

En una vista de canción, seleccione **Opciones** > **Detalles de canción**. La siguiente información aparece y puede editarla.

Nombre de canción: el nombre de la canción

Artista: el nombre del artista

Álbum: el nombre del álbum al cual pertenece la canción

Número de pista: la posición de la canción en el álbum al cual pertenece la canción

Año: el año en que salió el álbum

Género: la categoría de estilo en la cual se clasifica la canción

Compositor: el nombre del compositor de la canción

Comentario: cualquier otra información sobre esta canción

Para ver detalles como **Formato**, **Duración** y **Copyright**, seleccione **Opciones** > **Ver detalles de archivo** en la vista **Detalles de canción**.

Detalles de la biblioteca

En **Menú Música** seleccione **Opciones** > **Detalles Bibl. musical**. La siguiente información aparece en la pantalla:

Canción.: la cantidad actual de canciones en la biblioteca

Duración: la duración de todas las canciones actuales

Utilizado: el tamaño total de los archivos en la biblioteca de música

Disponible: el espacio disponible en el disco duro en gigabytes

Actualizada: la fecha más reciente en la que se modificó la biblioteca

Configurar una canción como el tono de timbre

1 Seleccione una canción.

2 Seleccione **Opciones** > **Como tono timbre**. La canción seleccionada ha sido programada como el tono de timbre para el perfil activo.

Cambiar el arte del álbum

1 Seleccione una canción.

2 Seleccione **Opciones** > **Arte del álbum**.

3 Seleccione **Opciones** > **Cambiar**.

4 Desplácese por las imágenes en su Galería y seleccione **Aceptar**.

5 Para restaurar el arte del álbum original, seleccione **Restaurar original**.

Sobre audio codificado

Las aplicaciones como reproductor Windows Media y Nokia Audio Manager codifican los archivos de audio con codecs como WMA o MP3 para ahorrar espacio de almacenamiento. La codificación se hace al retirar las partes de la señal de audio que no son audibles o que el oído humano casi no puede percibir.

Sin importar el codificador y calidad de codec, la calidad del sonido el archivo no es tan buena como el original.

Los codecs básicos admitidos por este dispositivo son AAC, WMA y MP3. Todos estos tienen distintas variaciones.

Velocidad de bits

Al codificar audio, la calidad del sonido resultante depende de la velocidad de muestra y velocidad de bits usada para la codificación. La velocidad de muestra se expresa en miles de muestras por segundo (kHz) y para música de CD la velocidad de muestra es fija en 44.1 kHz. La velocidad de bits se expresa en kilobits por segundo (kb/s). Mientras más alta la velocidad de bits, mejor será la calidad del sonido.

El nivel de calidad requerido depende de sus exigencias, los audífonos usados y el ruido a su alrededor. Para MP3s una velocidad de bits entre 128 y 192 kb/s genera usualmente buenos resultados como para escuchar música pop con su dispositivo. WMA o AAC por lo general tiene un resultado similar a MP3 con una velocidad de bits un paso más bajo

(96-160 kb/s). Las variaciones de AAC, como AAC+ y AAC+ realzado (eAAC+) proveen calidad de audio CD a velocidades de bits tan bajas como 48 kb/s. La música clásica y música con matices por lo general requieren una velocidad de bits más alta que la música básica pop.

Para discurso u otras fuentes para las cuales la calidad del sonido es menos importante que ahorrar espacio de almacenamiento, puede usar velocidades de bits entre 8 y 64 kb/s. A velocidades de bits bajas WMA o AAC por lo general proveen mejores resultados que MP3.

Velocidad de bits variable

Su dispositivo también admite la velocidad de bits variable (VBR). Velocidad de bits variable significa que la velocidad de bits de la codificación varía con la complejidad del material codificado. Al usar VBR, la codificación es optimizada para preservar una calidad de audio constante en la pista, en lugar de perder calidad en secciones complejas de audio con una codificación constante de velocidad de bits (CBR).

Personalizar su dispositivo

Su proveedor de servicios puede haber pedido que ciertas funciones de su dispositivo sean inhabilitadas o desactivadas. De ser así, éstas no aparecerán en el menú. Su dispositivo podría haber estado configurado especialmente para su proveedor de red. Esta configuración puede incluir cambios en nombres de menús, orden de menús e iconos. Comuníquese con su proveedor de servicios para obtener más información.

Las aplicaciones abiertas y las carpetas que tienen aplicaciones abiertas, son indicadas por un pequeño icono en el menú principal.

Modos: configurar tonos

Para configurar y personalizar los tonos de timbre, tonos de alerta de mensaje y otros tonos para diferentes eventos, entornos o grupos de llamantes, pulse y seleccione **Herramient.** > **Modos**. Si el perfil seleccionado no es **Normal**, el nombre del perfil aparece en la parte superior de la pantalla en el modo standby.

Para cambiar el perfil, pulse en el modo standby. Desplácese hasta el perfil que desea activar y seleccione **Aceptar**.

Para modificar un perfil, pulse y seleccione **Herramient.** > **Modos**. Desplácese al perfil y seleccione **Opciones** > **Personalizar**. Desplácese a la configuración que desea cambiar y pulse el joystick para acceder a las opciones. Los tonos almacenados en el disco duro son indicados con . Puede desplazarse por la lista de tonos y escucharlos antes de seleccionar uno. Para detener el sonido, pulse cualquier tecla.

Al seleccionar un tono, **Descargas tonos** abre un lista de favoritos. Puede seleccionar un favorito y conectarse a una página Web para descargar tonos.

Si desea escuchar el nombre de la persona que lo llama cuando suena el dispositivo, seleccione **Opciones** > **Personalizar** y configure **Decir nomb. llamante** como **Activar**. El nombre de la persona deberá estar guardado en **Contactos**.

Para crear un nuevo perfil, seleccione **Opciones** > **Crear nuevo**.

Si escuchar música y el perfil **Música** está seleccionado, sólo escuchará un bip para las llamadas entrantes y los demás tonos de alerta serán desactivados.

Perfil desconectado

El perfil **Desconectado** le permite usar el dispositivo sin necesidad de conexión a la red inalámbrica. Al activar el perfil **Desconectado**, se desactiva la conexión a la red inalámbrica, como es indicado por ✕ en el área que indica la intensidad de la señal. Se restringen todas las señales de teléfono móvil hacia y desde el dispositivo. Al tratar de enviar mensajes, éstos se guardan en el buzón de salida para ser enviados en otro momento.

Cuando tiene el perfil **Desconectado** activado, todavía puede usar la LAN inalámbrica (si está disponible), por ejemplo, para leer su e-mail o explorar por la Internet. Recuerde cumplir con cualquier requisito de seguridad aplicable al establecer y usar una conexión de LAN inalámbrica. Consulte "LAN inalámbrica", pág. 138.

Advertencia: En el perfil Desconectado, no se pueden realizar (ni recibir) llamadas, excepto las llamadas de emergencia, ni usar otras funciones que requieran la cobertura de red. Para realizar una llamada, primero debe activar la función de teléfono modificando los perfiles. Si el dispositivo está bloqueado, ingrese el código de bloqueo.

Advertencia: Su dispositivo deberá estar encendido para usar el perfil **Desconectado**. No encienda el dispositivo donde el uso de dispositivos móviles esté prohibido o pueda causar interferencia o peligro.

Para salir del perfil **Desconectado**, pulse y seleccione **Modos**, seleccione otro perfil y seleccione **Opciones** > **Activar** > **Sí**. El dispositivo rehabilita las transmisiones inalámbricas (siempre y cuando la intensidad de la señal sea suficiente). Si tiene activada la conectividad Bluetooth antes de acceder al perfil **Desconectado**, esta conexión será desactivada. La conectividad Bluetooth es reactivada automáticamente tras salir del perfil **Desconectado**. Consulte "Configuraciones", pág. 218.

Cambiar la apariencia de su dispositivo

Para cambiar la apariencia de su dispositivo, como la imagen de fondo e iconos, pulse y seleccione **Herramient.** > **Temas**. El tema activo es indicado por ✓. En **Temas** puede agrupar elementos de otros temas o seleccionar imágenes de **Galería** para personalizar aún más los temas. Los temas en el disco duro son indicados por .

Para abrir una conexión de explorador y descargar más temas, seleccione **Descargas temas**.

Para activar un tema, desplácese al tema y seleccione **Opciones** > **Aplicar**. Para obtener una vista previa del tema antes de activarlo, seleccione **Opciones** > **Vista previa**.

Para editar temas, desplácese a un tema y seleccione **Opciones** > **Editar** para cambiar las configuraciones de **Imagen de fondo** y **Ahorro de energía**.

Para restaurar el tema seleccionado a sus valores originales, seleccione **Opciones** > **Restaurar tem. orig.** al editar un tema.

Standby activo

Use su pantalla standby para tener acceso rápido a sus aplicaciones de uso frecuente. En forma predeterminada, el modo standby está activado.

Pulse , seleccione **Herramient.** > **Config.** > **Teléfono** > **Modo en espera** > **Standby activo** y pulse el joystick para activar o desactivar el modo de standby activo.

La pantalla de modo standby activo aparece con aplicaciones predeterminadas en la parte superior de la pantalla y los eventos de agenda, notas de tarea y reproductor listados en la parte inferior.

Desplácese a una aplicación o evento y pulse el joystick.

No se pueden usar los accesos directos del joystick estándar en el modo standby cuando el modo de standby activo está activado.

Para cambiar los accesos directos predeterminados para las aplicaciones, pulse , seleccione **Herramient.** > **Config.** > **Teléfono** > **Modo en espera** > **Aplic. en standby act.** y pulse el joystick.

Algunos accesos directos pueden ser fijos y no podrá cambiarlos.

Realizar llamadas

Llamadas de voz

Sugerencia: Para subir o bajar el volumen durante una llamada, pulse las teclas en el lado izquierdo de su dispositivo. Si ha configurado el volumen en **Silenciar micrófono**, no podrá ajustar el volumen. Para ajustar el volumen, seleccione **Micró.**, luego pulse la tecla de volumen.

1 En el modo standby, ingrese el número telefónico, incluso el código de área. Pulse para eliminar un número.

Para llamadas internacionales, pulse * dos veces para el carácter + (reemplaza el código de acceso internacional) e ingrese el código del país, código de área (omita el cero al comienzo si es necesario) y el número telefónico.

2 Pulse para llamar al número.

3 Pulse para finalizar la llamada (o cancelar el intento de llamada).

Sugerencia: También puede usar la tecla en el control remoto para contestar y finalizar la llamada.

Pulsar la tecla finalizará siempre la llamada, así tenga otra aplicación activa.

Para realizar una llamada desde **Contactos**, pulse y seleccione **Contactos**. Desplácese al nombre deseado o ingrese las primeras letras del nombre en el campo de búsqueda. Los contactos que coinciden aparecen en una lista. Pulse para llamar.

Antes de poder realizar una llamada de esta forma, deberá copiar los contactos en su tarjeta SIM a **Contactos**. Consulte "Copiar contactos", pág. 165.

Para llamar a su buzón de voz (servicio de red), mantenga pulsada 1 en el modo standby. También consulte "Desvío de llamadas", pág. 237.

Sugerencia: Para cambiar el número telefónico de su buzón de voz, pulse y seleccione **Herramient.** > **Buzón voz** > **Opciones** > **Cambiar número**. Ingrese el número (provisto por su proveedor de servicios) y seleccione **Aceptar**.

Para llamar a un número que llamó recientemente en el modo standby, pulse para acceder a una lista de los últimos veinte números que llamó o trató de llamar. Desplácese al número deseado y pulse para llamar al número.

Realizar una llamada de multiconferencia

1 Llame al primer participante.

2 Para llamar a otro participante, seleccione **Opciones** > **Nueva llamada**. La primera llamada es retenida automáticamente.

3 Cuando la nueva llamada es contestada, para incorporar al primer participante en la llamada de multiconferencia, seleccione **Opciones** > **Multiconferencia**.

Para agregar un nuevo participante, repita el paso 2 y seleccione **Opciones** > **Multiconferencia** > **Incluir en multiconf.** Su dispositivo admite llamadas de multiconferencia entre un máximo de seis participantes, incluyéndolo a usted.

Para mantener una conversación privada con uno de los participantes, seleccione **Opciones** > **Multiconferencia** > **Privada**. Seleccione un participante y seleccione **Privada**. La llamada de multiconferencia es retenida en su dispositivo. Los otros participantes pueden aún continuar con la llamada de multiconferencia. Cuando finalice la conversación privada, seleccione **Opciones** > **Multiconferencia** para reincorporarse a la llamada de multiconferencia.

Para desconectar a un participante, seleccione **Opciones** > **Multiconferencia** > **Excluir participante**, desplácese al participante y seleccione **Excluir**.

4 Para finalizar la llamada de multiconferencia activa, pulse .

Marcación rápida de un número telefónico

Para activar la marcación rápida, pulse y seleccione **Herramient.** > **Config.** > **Llamadas** > **Marcación rápida** > **Activar**.

Para asignar un número telefónico a una de las teclas de marcación rápida (de **2** a **9**), pulse y seleccione **Herramient.** > **Marc. ráp.** Desplácese a la tecla a la que desea asignar el número telefónico y seleccione **Opciones** > **Asignar**. **1** está reservada para el buzón de voz.

Para llamar en el modo standby, pulse la tecla de marcación rápida y .

Marcación por voz

Su dispositivo admite los comandos de voz mejorados. Los comandos de voz mejorados no dependen de la voz del hablante, así que el usuario no graba etiquetas de voz por adelantado. En cambio, el dispositivo genera una etiqueta de voz para los ingresos en contactos y la compara con la etiqueta de voz hablada. La función de reconocimiento de voz en el dispositivo se adapta a la voz del usuario para reconocer mejor los comandos de voz.

La etiqueta de voz para un contacto es el nombre o sobrenombre que se guarda en la tarjeta de contacto. Para escuchar la etiqueta de voz sintetizada, abra una tarjeta de contacto y seleccione **Opciones** > **Reproducir etiq. voz**.

Realizar una llamada usando una etiqueta de voz

Nota: Usar las etiquetas de voz podría resultar difícil en un lugar ruidoso o durante una emergencia, así que no debería depender sólo de la marcación por voz en todas las circunstancias.

Cuando usa la marcación por voz, el altavoz está en uso. Sujete el dispositivo a una corta distancia de usted cuando pronuncie la etiqueta de voz.

1 Para iniciar la marcación de voz, mantenga pulsada la tecla de selección derecha —. Si está usando un auricular compatible con el botón auricular, mantenga pulsado este botón para iniciar la marcación por voz.

2 El dispositivo emite un tono breve y aparece **Hable ahora**. Pronuncie claramente el nombre o sobrenombre que está guardado en la tarjeta de contacto.

3 El dispositivo reproduce una etiqueta de voz sintetizada para el contacto reconocido en el idioma seleccionado en el dispositivo y muestra el nombre y número. Tras una interrupción de 1,5 segundos, el dispositivo marca el número.

Si el contacto reconocido no fue correcto, seleccione **Siguiente** para ver una lista de otras coincidencias o **Cancelar** para cancelar la marcación por voz.

Si varios números fueron guardados bajo el mismo nombre, el dispositivo selecciona el número predeterminado, si fue configurado. De lo contrario, el dispositivo selecciona el primer número disponible de los siguientes: **Celular**, **Celular (casa)**, **Celular (oficina)**, **Teléfono**, **Teléfono (casa)** y **Teléfono (oficina)**.

Contestar o rechazar una llamada

Para contestar la llamada, pulse ⌐ o abra la tapa.

Para silenciar el tono de timbre cuando hay una llamada entrante, seleccione **Silencio**.

Sugerencia: También puede usar la tecla ⌐ en el control remoto para contestar y finalizar la llamada.

Si no desea contestar la llamada, pulse ¬ para rechazarla. El llamante escucha un tono de línea ocupada. Si ha activado la función **Desvío llam.** > **Si está ocupado** para desviar llamadas, al rechazar una llamada entrante también desviará la llamada. Consulte "Desvío de llamadas", pág. 237.

Cuando rechaza una llamada entrante, también puede enviar un mensaje de texto al llamante informándole la razón por la cual no puede contestar la llamada. Seleccione **Silencio** > **Env. mens.** Puede editar el texto antes de enviarlo. Para configurar esta opción y escribir un mensaje de texto estándar, consulte "Configuraciones de llamadas", pág. 228.

Llamada en espera (servicio de red)

Puede contestar una llamada mientras tiene otra en curso si ha activado **Llamada en espera** en **Config.** > **Llamadas** > **Llamada en espera**.

Sugerencia: Si desea cambiar los tonos de timbre del dispositivo para varios entornos y ocasiones, por ejemplo, cuando desea que su dispositivo no suene, consulte "Modos: configurar tonos", pág. 156.

Para contestar la llamada en espera, pulse . La primera llamada es retenida.

Para intercambiar entre las dos llamadas, pulse **Cambiar**. Seleccione **Opciones** > **Transferir** para conectar una llamada entrante o una llamada en espera con una llamada activa y para desconectarse de ambas llamadas. Para finalizar la llamada activa, pulse o cierre la tapa. Para finalizar ambas llamadas, seleccione **Opciones** > **Finalizar todas**.

Opciones durante una llamada de voz

Muchas de las opciones que puede usar durante una llamada de voz son servicios de red. Seleccione **Opciones** durante una llamada para acceder a algunas de las siguientes opciones: **Silenciar micrófono** o **Micró.**, **Contestar**, **Rechazar**, **Cambiar**, **Retener** o **Recuperar**, **Activar teléfono**, **Activar altavoz**, o **Activar altavoz** (si un auricular compatible con la conectividad Bluetooth está conectado), **Finaliz. llam. activa** o **Finalizar todas**, **Nueva llamada**, **Multiconferencia** y **Transferir**. Seleccione una de estas opciones:

Colgar y cambiar: para finalizar una llamada activa y cambiarla al contestar la llamada en espera.

Enviar MMS (sólo en redes UMTS): para enviar una imagen o un video en un mensaje multimedia al otro participante de la llamada. Puede editar el mensaje y cambiar el destinatario antes de enviarlo. Pulse para enviar el archivo a un dispositivo compatible (servicio de red).

Enviar DTMF: para enviar secuencias de tonos DTMF, por ejemplo, una contraseña. Ingrese la secuencia de tonos DTMF o búsquela en **Contactos**. Para ingresar un carácter de espera (**w**) o un carácter de pausa (**p**), pulse * continuamente. Pulse **Aceptar** para enviar el tono.

Sugerencia: Puede agregar tonos DTMF a los campos **Número de teléfono** o **DTMF** en una tarjeta de contacto.

Contactos (Directorio)

Pulse y seleccione **Contactos**. En **Contactos** puede agregar un tono de timbre personal o una imagen en miniatura a una tarjeta de contacto. Puede crear grupos de contactos, los cuales le permiten enviar mensajes de texto o e-mail a varios destinatarios a la vez. Puede agregar información de contactos recibida (tarjetas de negocios) a sus contactos. Consulte "Datos y configuraciones", pág. 186. La información de contactos sólo puede enviarse o recibirse de dispositivos compatibles.

Sugerencia: Haga copias de seguridad, con regularidad, de la información en el dispositivo. Puede restaurar la información, como contactos, en su dispositivo en otro momento. Puede usar Nokia PC Suite para hacer una copia de seguridad de sus contactos en una PC compatible. Consulte el CD-ROM suministrado con su dispositivo.

Guardar nombres y números

1 Seleccione **Opciones** > **Nuevo contacto**.

2 Rellene los campos que desee y seleccione **Hecho**.

Para editar tarjetas de contacto en **Contactos**, desplácese a la tarjeta de contacto que desea editar y seleccione **Opciones** > **Editar**.

Para borrar una tarjeta de contacto en **Contactos**, seleccione una tarjeta y pulse . Para borrar varias tarjetas de contacto a la misma vez, mantenga pulsada y pulse el joystick para marcar cada contacto que desea borrar. Pulse para borrar los contactos marcados.

Sugerencia: Para agregar y editar tarjetas de contacto, utilice Nokia Contacts Editor disponible en Nokia PC Suite. Consulte el CD-ROM suministrado con su dispositivo.

Para adjuntar una imagen en miniatura a una tarjeta de contacto, abra la tarjeta de contacto y seleccione **Opciones** > **Editar** > **Opciones** > **Agregar miniatura**. La imagen en miniatura aparece cuando el contacto llama.

Sugerencia: La marcación rápida es una manera rápida de llamar a los números telefónicos utilizados con más frecuencia. Podrá asignar teclas de marcación rápida a ocho números telefónicos. Consulte "Marcación rápida de un número telefónico", pág. 160.

Sugerencia: Para enviar información de contacto, seleccione la tarjeta que desea enviar. Seleccione **Opciones** > **Enviar** > **Vía mensaje corto**, **Vía multimedia**, o **Vía Bluetooth**. Consulte "Mensajería", pág. 179 y "Enviar datos usando la conectividad Bluetooth", pág. 219.

Para agregar un contacto a un grupo, seleccione **Opciones** > **Agregar a grupo:** (sólo aparece si ha creado un grupo). Consulte "Crear grupos de contactos", pág. 165.

Para ver la cantidad de contactos, grupos y la memoria disponible, seleccione **Contactos** > **Opciones** > **Información Contactos**.

Números y direcciones predeterminados

Puede asignar números y direcciones predeterminados a una tarjeta de contacto. De esta forma, si el contacto tiene varios números o direcciones, podrá fácilmente llamarle o enviarle un mensaje al contacto a un número o dirección específica. El número predeterminado es también usado en marcación por voz.

1. En contactos, seleccione un contacto y pulse el joystick.
2. Seleccione **Opciones** > **Predeterminados**.
3. Seleccione un valor predeterminado al que desea agregar un nombre o dirección y seleccione **Asignar**.
4. Seleccione un número o dirección que desee configurar como predeterminado.

La dirección o número predeterminado aparecerá subrayado en la tarjeta de contacto.

Directorio SIM y servicios SIM

Para obtener información sobre disponibilidad y uso de los servicios de tarjetas SIM, comuníquese con el proveedor de su tarjeta SIM. Este puede ser su proveedor de servicios, operador de red u otro proveedor.

Pulse y seleccione **Contactos** > **Opciones** > **Contactos SIM** > **Directorio SIM** para ver los nombres y números guardados en la tarjeta SIM. En el directorio SIM podrá agregar, editar o copiar números a contactos y realizar llamadas.

Para ver la lista de números de marcación fija, seleccione **Opciones** > **Contactos SIM** > **Contac. marcac. fija**. Esta configuración sólo aparece si es admitida por su tarjeta SIM.

Para restringir las llamadas de su dispositivo a números telefónicos específicos, seleccione **Opciones** > **Activar marcac. fija**. Para agregar nuevos números a la lista de marcación fija, seleccione **Opciones** > **Nuevo contacto SIM**. Se necesita el código PIN2 para estas funciones.

Al usar **Marcación fija**, no es posible hacer conexiones de paquetes de datos, excepto cuando envía mensajes de texto por una conexión de paquetes de datos. En este caso, el número del centro de mensajes y el número telefónico del destinatario deberán estar incluidos en la lista de marcación fija.

Cuando la marcación fija está activada, pueden efectuarse llamadas al número oficial de emergencia programado en su dispositivo.

Copiar contactos

Para copiar nombres y números de una tarjeta SIM a su dispositivo, pulse y seleccione **Contactos** > **Opciones** > **Contactos SIM** > **Directorio SIM**. Seleccione los nombres que desea copiar y **Opciones** > **Copiar a Contactos**.

Para copiar contactos en su tarjeta SIM, pulse y seleccione **Contactos**. Seleccione los nombres que desea copiar y **Opciones** > **Copiar** > **Al directorio SIM**. Sólo los campos de tarjeta de contacto admitidos por su tarjeta SIM son copiados.

Para editar el número telefónico que admite llamadas de su dispositivo, pulse y seleccione **Contactos** > **Opciones** > **Contactos SIM** > **Contac. marcac. fija**. Desplácese al contacto y seleccione **Opciones** > **Editar**. Puede necesitar su código PIN2 para editar sus números de marcación fija.

Sugerencia: Puede sincronizar sus contactos con una PC compatible con Nokia PC Suite. Consulte el CD-ROM suministrado con su dispositivo.

Agregar tonos de timbre para contactos

Cuando un contacto o participante de un grupo le llame, el dispositivo reproducirá el tono de timbre seleccionado (si el número telefónico del llamante se envía con la llamada y su dispositivo lo reconoce).

1. Pulse el joystick para abrir una tarjeta de contacto o ir a la lista de grupos y seleccione un grupo de contactos.
2. Seleccione **Opciones** > **Tono de timbre**. Verá una lista de tonos de timbre.
3. Seleccione el tono de timbre que desea usar para el contacto o grupo seleccionado.

Para borrar el tono de timbre, seleccione **Tono predeter.** de la lista de tonos de timbre.

Crear grupos de contactos

1. En **Contactos**, mueva el joystick hacia la derecha para abrir la lista de grupos.
2. Seleccione **Opciones** > **Nuevo grupo**.
3. Escriba un nombre para el grupo o use el nombre predeterminado **Grupo** (número) y seleccione **Aceptar**.
4. Abra el grupo y seleccione **Opciones** > **Agregar miembros**.

5 Desplácese a un contacto y pulse el joystick para marcarlo. Para agregar múltiples miembros a la vez, repita este procedimiento en todos los contactos que desea agregar.

6 Seleccione **Aceptar** para agregar los contactos al grupo.

Para dar otro nombre a un grupo, seleccione **Opciones** > **Renombrar**, ingrese el nuevo nombre y seleccione **Aceptar**.

Eliminar miembros de un grupo

1 En la lista de grupos, abra el grupo que desea modificar.

2 Desplácese al contacto y seleccione **Opciones** > **Eliminar de grupo**.

3 Seleccione **Sí** para borrar el contacto del grupo.

Sugerencia: Para verificar a qué grupo pertenece el contacto, desplácese al contacto y seleccione **Opciones** > **Pertenece a grupos**.

Aplicaciones multimedia

Cámara

El Nokia N91 admite una resolución de captura de imagen de 1600 x 1200 píxeles. La resolución de imagen en estos materiales puede verse diferente.

Para activar la cámara, pulse en la vista Standby y seleccione **Cámara**. La cámara se inicia y puede ver la vista a capturar.

Si la cámara está en el modo **Video**, mueva el joystick a la izquierda para iniciar el modo **Imagen**.

Las imágenes son guardadas automáticamente en la galería en el formato .jpg. Puede enviar imágenes en un mensaje multimedia como un archivo adjunto de e-mail o usando la conectividad Bluetooth.

Tomar fotos

Para tomar una foto, pulse el joystick. No mueva el dispositivo antes de que se guarde la imagen. La imagen se guarda automáticamente en la carpeta **Imágenes** de **Galería**. Consulte "Galería", pág. 172.

Guardar la imagen tomada puede tardar más si cambia las configuraciones de zoom, luz o color.

Al tomar una foto, tenga en cuenta lo siguiente:

- Para mantener la cámara fija, use ambas manos.
- Para ajustar el color antes de tomar una foto, seleccione **Opciones** > **Ajustar** > **Equilibrio de blancos** o **Tono de color**. Consulte "Ajustar los colores", pág. 168.
- La calidad de una foto tomada con el acercamiento (zoomed) tiene menos calidad que la de una foto tomada sin acercamiento (nonzoomed).
- La cámara cambia al modo de ahorro de batería cuando no se pulsan las teclas por algún período de tiempo. Para continuar tomando fotos, pulse el joystick.

Después de tomar la foto:

- Para borrar la imagen, seleccione **Opciones** > **Eliminar**.
- Para enviar la imagen **Vía multimedia**, **Vía e-mail** o **Vía Bluetooth**, seleccione **Opciones** > **Enviar**. Para obtener más información, consulte "Mensajería", pág. 179 y "Conectividad Bluetooth", pág. 217. Esta opción no está disponible durante una llamada activa.
- Para enviar una imagen a la otra persona durante una llamada activa, seleccione **Opciones** > **Enviar MMS**.

Modos de exposición

- Para configurar la cámara para tomar seis fotos en secuencia, seleccione **Opciones** > **Modo de secuencia**. También puede usar el modo de secuencia con el **Temporizador**. Consulte "Usted en la foto: Temporizador", pág. 168. Las fotos son guardadas automáticamente en **Galería**.

  Después de tomar las fotos, éstas aparecen en una cuadrícula en la pantalla principal. Para ver una foto, pulse el joystick para abrirla.

- Para configurar la cámara para tomar fotos en condiciones de poca iluminación, seleccione **Opciones** > **Modo noche**

Usted en la foto: Temporizador

Use el temporizador para retrasar la captura de una imagen con el fin de incluirse en la foto. Para programar el tiempo de espera del temporizador, seleccione **Opciones** > **Temporizador** > **10 segundos**, **20 segundos** o **30 segundos**. Para activar el temporizador, seleccione **Activar**. El indicador del temporizador ( ) destella y el dispositivo emite un bip cuando el tiempo está en progreso. La cámara toma la foto al finalizar el tiempo de espera seleccionado. También puede usar el **Temporizador** en el modo de secuencia. Consulte "Modos de exposición", pág. 168.

Ajustar los colores

Para activar la cámara para que reproduzca los colores con más precisión o para agregar efectos a sus fotos o videos, seleccione **Opciones** > **Ajustar** y seleccione alguna de las siguientes opciones:

Equilibrio de blancos: seleccione de la lista la condición de iluminación actual. Esto hace que la cámara reproduzca colores con más precisión.

Tono de color: seleccione un efecto de color de la lista.

La pantalla cambia para ajustarse a las configuraciones seleccionadas, para mostrarle cómo los cambios afectan las fotos o videos.

Las configuraciones regresan a sus valores predeterminados.

Ajustar las configuraciones de cámara para imagen fija

Para cambiar las configuraciones principales, seleccione **Opciones** > **Configuraciones**. Estas configuraciones permanecen tal cual hasta que las cambie. Seleccione alguna de las siguientes opciones:

Calidad de imagen: **Alta**, **Normal** o **Básica**. Cuanto mejor es la calidad de la imagen, más memoria consume.

Resolución de imagen: **Alta** (resolución 1600x1200), **Normal** (resolución 800x600) o **Básica** (resolución 640x480). Cuanto mejor es la resolución de la imagen, más memoria se consume. La calidad de una foto tomada con el acercamiento (zoomed) tiene menos calidad que la de una foto tomada sin acercamiento (nonzoomed).

Nombre imag. predet.: configure un nombre para las fotos que tome. De manera predeterminada, las fotos reciben el nombre en el formato 'Imagen'. Por ejemplo, si programa Playa como el nombre preconfigurado, cuando toma fotos, la cámara le dará el nombre Playa, Playa(01), Playa(02), etc., hasta que vuelva a cambiar la configuración.

Mostrar imag. captur.: seleccione **Sí** para ver la imagen después de tomarla o **No** para seguir tomando fotos inmediatamente.

Memoria en uso: seleccione donde desea guardar sus imágenes.

Visor de imágenes

El visor de imágenes muestra lo siguiente:

El indicador para la memoria en uso (1) muestra la ubicación en la que se guardan las fotos actualmente: en la memoria del dispositivo o en el disco duro .

El contador de imágenes (2) en la parte superior del visor muestra una cifra estimada de cuántas imágenes, dependiendo de la calidad de imagen seleccionada, pueden guardarse en la memoria disponible en su dispositivo.

el indicador zoom (3) muestra el nivel de zoom. Mueva el joystick hacia arriba y abajo para acercar y alejar la imagen.

el indicador **Modo noche** (4) muestra que la cámara ha sido optimizada para tomar fotos en condiciones de poca iluminación.

el indicador **Modo de secuencia** (5) muestra que la cámara está en el modo de secuencia.

el indicador de temporizador (6) muestra que el temporizador está activado. Consulte "Usted en la foto: Temporizador", pág. 168.

Sugerencias para tomar fotos

Calidad de imagen

Use la calidad de imagen adecuada. La cámara tiene tres modos de calidad de imagen (**Básica**, **Normal** y **Alta**). Use la configuración **Alta** para asegurarse que la cámara produzca la foto de la mejor calidad disponible. Tenga en cuenta que una foto de mejor calidad requiere más espacio de almacenamiento. Para archivos adjuntos de MMS y

e-mail puede ser necesario usar la configuración **Normal** o **Básica**. Puede definir la calidad en **Opciones** > **Configuraciones**.

Fondo

Use un fondo sencillo. Para retratos y otras fotos con personas, evite que el sujeto de la foto se encuentre en frente de un fondo cargado y complejo que quite la atención del sujeto. Mueva la cámara o el sujeto, cuando no cuente con estas condiciones. Mueva la cámara más cerca del objeto para tomar fotos más claras.

Profundidad

Al tomar fotos de paisajes y escenarios, agregue profundidad a sus fotos al poner objetos en primer plano. Si el objeto en primer plano está muy cerca de la cámara, puede aparecer borroso.

Condiciones de iluminación

Cambiar la fuente, cantidad y dirección de la luz puede cambiar las fotos drásticamente. Aquí se presentan algunas condiciones de iluminación comunes:

- **Fuente de iluminación detrás del sujeto.** Evite que el sujeto se encuentre en frente de una fuente de iluminación fuerte. Si la fuente de iluminación está detrás del sujeto o visible en la pantalla, la foto resultante puede ser de bajo contraste, puede resultar muy oscura y puede tener efectos de luz no deseados.
- **Sujeto con iluminación lateral.** Una fuerte iluminación lateral da un efecto drástico pero puede ser muy fuerte, lo cual resulta en demasiado contraste.
- **Fuente de iluminación en frente del sujeto.** La luz del sol muy fuerte puede causar que el sujeto cierre los ojos. El contraste también puede ser muy alto.
- **Optimal lighting** se encuentra en situaciones de mucha iluminación difusa y suave, por ejemplo, en un día claro, un día parcialmente nublado o un día soleado bajo la sombra de árboles.

Grabar videos

Para activar la cámara, pulse y seleccione **Cámara** en la vista Standby. La cámara se inicia y puede ver la vista a capturar.

Si la cámara está en el modo **Imagen**, abra la grabadora de video al mover el joystick a la derecha.

Para ajustar el color antes de grabar video, seleccione **Opciones** > **Ajustar** > **Equilibrio de blancos** o **Tono de color**. Consulte "Ajustar los colores", pág. 168.

1 Pulse el joystick para comenzar a grabar. Aparece el icono de grabación y se emite un tono para avisarle al sujeto que el video se está grabando.

2 Para hacer una pausa en la grabación en cualquier momento, pulse **Pausa**. El icono de pausa (❚❚) destella en la pantalla. La grabación de video se detiene automáticamente si la grabación es programada en pausa y no se pulsa ninguna tecla en un minuto.

3 Seleccione **Continuar** para seguir grabando.

4 Seleccione **Parar** o pulse el joystick para detener la grabación. El videoclip se guarda automáticamente en la carpeta **Videoclips** en **Galería**. Consulte "Galería", pág. 172.

Los indicadores del visor de imágenes muestran lo siguiente:

el indicador de micrófono muestra que el micrófono está en silencio.

el indicador de temporizador muestra que el temporizador está activado. Consulte "Usted en la foto: Temporizador", pág. 168.

el indicador zoom muestra el nivel de zoom. Mueva el joystick hacia arriba y abajo para acercar y alejar la imagen.

Después de grabar un videoclip:

- Para reproducir inmediatamente el videoclip que acaba de grabar, seleccione **Opciones** > **Reproducir**.
- Para borrar el video, seleccione **Opciones** > **Eliminar**.
- Para regresar al visor para grabar un video nuevo, pulse el joystick.
- Para enviar el video, seleccione **Opciones** > **Enviar** > **Vía multimedia**, **Vía e-mail** o **Vía Bluetooth**. Para obtener más información, consulte "Mensajería", pág. 179 y "Conectividad Bluetooth", pág. 217. Esta opción no está disponible durante una llamada activa.
- Para enviar un videoclip a la otra persona durante una llamada activa, seleccione **Opciones** > **Enviar MMS**.

Configuraciones de video

Hay dos tipos de configuraciones para la grabadora de video: configuraciones temporales y las configuraciones predeterminadas. Para ajustar las configuraciones de contraste, brillo y color, consulte "Ajustar los colores", pág. 168. Las configuraciones temporales regresan a las predeterminadas después de cerrar la cámara, pero las configuraciones predeterminadas quedan igual hasta que las cambie de nuevo. Para cambiar las configuraciones predeterminadas, seleccione **Opciones** > **Configuraciones** y alguna de las siguientes opciones:

Duración: para videoclip óptimo para MMS, seleccione **Corto**.

Resolución de video—**Alta** (CIF, 352x288), **Normal** (QCIF 176x144) o **Básica** (SubQCIF 128x96). Cuanto mejor es la resolución, el video más memoria consume.

Nombre de video predeterminado: configure un nombre para el video que grabó. Seleccione **Fecha** o **Texto**. Si selecciona **Texto**, los videos obtienen el nombre en el formato Video en forma predeterminada. Por ejemplo, si programa Playa como el nombre preconfigurado, cuando toma videos la cámara le dará el nombre Playa, Playa(01), Playa(02), etc., hasta que vuelva a cambiar la configuración.

Galería

Para guardar y organizar sus imágenes, videoclips, clips de sonido y enlaces de streaming, pulse y seleccione **Medios** > **Galería**. Para abrir la galería desde la cámara, seleccione **Opciones** > **Ir a Galería**.

Seleccione **Imágenes**, **Videoclips**, **Pistas**, **Clips de sonido**, **Enlac. streaming**, **Presentaciones** o **Todos los arch.** y pulse el joystick para abrirlo.

Puede explorar y abrir carpetas y marcar, copiar y mover elementos a las carpetas. También, puede crear carpetas y marcar, copiar y agregar elementos a las carpetas. Consulte "Carpetas", pág. 173.

Sugerencia: Puede transferir imágenes de su dispositivo a una PC compatible con Nokia Phone Browser disponible en Nokia PC Suite. Consulte el CD-ROM suministrado con su dispositivo.

Pulse el joystick para abrir un archivo. Los videoclips, archivos .ram y enlaces de streaming se abren y se reproducen en la aplicación RealPlayer y los clips de sonido se reproducen en la aplicación Reproductor de música. Consulte "RealPlayer™", pág. 173. y "Reproductor de música", pág. 145. Consulte "Ver imágenes y videoclips", pág. 172.

Para renombrar un archivo, seleccione un archivo y **Opciones** > **Renombrar**.

Para descargar archivos en **Galería** en una de las carpetas principales utilizando el explorador, seleccione **Descarg. gráf.** o **Descarg. vid.**, **Descarg. pistas** o **Descargas son.**. Se abre el explorador y podrá seleccionar un favorito del cual el sitio puede descargar.

Ver imágenes y videoclips

Las fotos tomadas se guardan en la carpeta **Imágenes** y los videoclips se guardan en la carpeta **Videoclips** en **Galería**. La calidad de un videoclip es indicada por uno de los siguientes iconos: (Alta), (Normal) y (Compartir). También pueden enviarle imágenes y videoclips en un mensaje multimedia, como un archivo adjunto de e-mail o mediante la conectividad Bluetooth. Para poder ver una imagen recibida o videoclip en la galería o en el reproductor multimedia, deberá guardarlo.

Abra la carpeta **Imágenes** o **Videoclips** en **Galería**. Para explorar los archivos, desplácese hacia arriba o abajo y pulse el joystick para abrir un archivo para verlo.

Para eliminar una imagen o un videoclip, pulse C.

Carpetas

Con las carpetas puede administrar convenientemente sus imágenes y videoclips.

Para crear una carpeta nueva, seleccione **Opciones** > **Organizar** > **Nueva carpeta**. Ingrese un nombre para la carpeta y seleccione **Aceptar**.

Para agregar una imagen o videoclip en un carpeta en la galería, desplácese a la imagen o videoclip y seleccione **Opciones** > **Organizar** > **Copiar a disco duro** / **Copiar a mem. teléf.** o **Mover a disco duro** / **Mover a mem. teléf.** dependiendo de la ubicación actual del archivo. Se abre una lista de carpetas. Seleccione la carpeta a la cual desea agregar la foto o videoclip y pulse el joystick.

Para eliminar un archivo de una carpeta, pulse C. El archivo no se borra de la carpeta **Imágenes** ni **Videoclips** en **Galería**.

Reproductor de música

Consulte "Reproductor de música" en pág. 145 para obtener más información.

RealPlayer™

Pulse y seleccione **Medios** > **RealPlyr**. Con **RealPlyr**, puede reproducir videoclips o archivos multimedia de contenido directo sobre al aire. Puede activar un enlace de streaming cuando explora por páginas Web o guardarlo en el disco duro.

RealPlyr admite archivos con extensiones como .awb o .m4a, pero no admite todas las variaciones de los formatos de archivo. Por ejemplo, algunos archivos .mp4 pueden incluir contenido que no cumple con los estándares 3GPP y, por lo tanto, no se pueden abrir.

Sugerencia: Puede transferir archivos de música desde su dispositivo a su disco duro con Nokia PC Suite. Consulte el CD-ROM suministrado con su dispositivo.

Reproducir video

1 Para reproducir un archivo multimedia guardado en la memoria del teléfono o en el disco duro, seleccione **Opciones** > **Abrir** y seleccione:

Clips más recientes: para reproducir uno de los últimos seis archivos reproducidos en **RealPlyr**.

Clip guardado: para reproducir un archivo guardado en **Galería**. Consulte "Galería", pág. 172.

2 Desplácese a un archivo y pulse el joystick para reproducir el archivo.

Sugerencia: Para ver un videoclip en el modo de pantalla completa, pulse 2 . Pulse otra vez para regresar al modo de pantalla normal.

Contenido directo sobre el aire (streaming)

Varios proveedores de servicios requieren que use un punto de acceso a Internet (IAP) como su punto de acceso predeterminado. Otros proveedores de servicios le permiten usar un punto de acceso WAP.

Los puntos de acceso pueden ser configurados al encender su dispositivo por primera vez.

Comuníquese con su proveedor de servicios para obtener más información.

En **RealPlyr**, sólo puede abrir una dirección rtsp:// URL. Sin embargo **RealPlyr** aceptará un enlace http a un archivo .ram.

Para reproducir contenido directo sobre el aire, seleccione un enlace de streaming guardado en **Galería**, en una página Web o recibido en un mensaje de texto o multimedia. Antes de iniciar la reproducción de contenido directo, su dispositivo se conecta al sitio y empieza a cargar el contenido. El contenido no se guarda en su dispositivo.

Recibir configuraciones de RealPlayer

Puede recibir configuraciones **RealPlyr** en un mensaje de texto especial del operador de red o proveedor de servicios. Consulte "Datos y configuraciones", pág. 186. Para obtener más información, comuníquese con su operador de red o proveedor de servicios.

Cambiar las configuraciones de RealPlayer

Seleccione **Opciones** > **Configuraciones** y alguna de las siguientes opciones:

Video: para ajustar el contraste del video y para que **RealPlyr** repita automáticamente los videoclips tras terminar de reproducirlos.

Audio: para seleccionar si desea repetir la reproducción de listas de pistas y reproducir los clips de sonido en una lista de reproducción en orden aleatorio.

Conexión: para usar un servidor proxy, cambie el punto de acceso predeterminado y configure el rango de puerto usado al conectarse. Comuníquese con su proveedor de servicios para obtener las configuraciones correctas.

Proxy:

- **Usar proxy**: seleccione **Sí**.
- **Dirección de proxy**: ingrese la dirección IP del servidor proxy.
- **Número puerto proxy**: ingrese el número de puerto del servidor proxy.

Glosario: los servidores proxy son servidores intermediarios entre los servidores multimedia y sus usuarios. Ciertos proveedores de servicios los usan para proveer seguridad adicional o acelerar el acceso a las páginas de explorador que contienen clips de sonido o videoclips.

Red:

- **Punto acceso predet.**: desplácese al punto de acceso que desea usar para conectarse a la Internet y pulse el joystick.
- **Tiempo de conexión**: configure el tiempo para que **RealPlyr** se desconecte de la red al hacer una pausa en la reproducción de un clip multimedia a través de un enlace de red, seleccione **Definido usuario** y pulse el joystick. Ingrese la hora y seleccione **Aceptar**.
- **Puerto UDP menor**: ingrese el número menor de puerto del rango de puerto del servidor. El valor mínimo es 6970.
- **Puerto UDP mayor**: ingrese el número mayor de puerto del rango de puerto del servidor. El valor máximo es 32000.

  Seleccione **Opciones** > **Config. avanzadas** para editar los valores de ancho de banda para diferentes redes.

FlashPlayer

Con Flash player, puede ver, reproducir e interactuar con archivos flash creados para dispositivos móviles.

Organizar archivos flash

Pulse y seleccione **Medios** >**Rep Flash**. Desplácese a la derecha.

Para abrir una carpeta o reproducir un archivo flash, desplácese al archivo y pulse el joystick.

Para enviar un archivo flash a un dispositivo compatible, desplácese al archivo y pulse la tecla Llamar.

Para copiar un archivo flash a otra carpeta, seleccione **Organizar** > **Copiar a carpeta**.

Para mover un archivo flash a otra carpeta, seleccione **Organizar** > **Mover a carpeta**

Para crear una carpeta para organizar sus archivos flash, seleccione **Organizar** > **Nueva carpeta**.

Las opciones disponibles pueden variar.

Para borrar un archivo flash, desplácese al archivo y pulse .

Reproducir archivos flash

Pulse y seleccione **Medios** >**Rep Flash**. Desplácese al archivo flash y pulse el joystick.

Seleccione **Opciones** y alguna de las siguientes opciones:

Nota: las opciones pueden no estar disponibles en todos los archivos flash.

- **Pausa** para hacer una pausa en la reproducción.
- **Parar** para detener la reproducción.
- **Volumen** para ajustar el volumen de reproducción. Para subir o bajar el volumen desplácese a la izquierda o derecha.
- **Calidad** para seleccionar la calidad de reproducción.

  Si la reproducción parece lenta o irregular, cambie la configuración **Calidad** a **Normal** o **Baja**.
- **Pantalla completa** para reproducir el archivo usando toda la pantalla. Para regresar a la pantalla regular, seleccione **Pantalla normal**.

  Aunque no es visible en la pantalla completa, las funciones de las teclas pueden estar disponibles cuando pulse alguna tecla en el área debajo de la pantalla.
- **Ajustar a pantalla** para reproducir el archivo en su tamaño original tras acercarlo (zoom).
- **Activar modo panorám.** para poder moverse por la pantalla con el joystick cuando ha acercado la imagen.

Las opciones disponibles pueden variar.

Grabadora

Pulse y seleccione **Medios** > **Grabadora** para grabar conversaciones telefónicas y recordatorios de voz. Si está grabando una conversación telefónica, ambos participantes escucharán un tono cada cinco segundos durante la grabación.

No se puede usar la grabadora cuando tiene activa una llamada de datos o conexión GPRS.

Radio

Nota: Está función no esta disponible en todas los países, consulte la disponibilidad de esta función con su operador local.

Pulse y seleccione **Radio**. Puede usar la aplicación como una radio FM tradicional con sintonización automática y estaciones guardadas o con información visual en pantalla asociada al programa de radio, si sintoniza estaciones que ofrecen el servicio de Visual Radio. El servicio Visual Radio usa los paquetes de datos (servicio de red). Puede escuchar la radio FM mientras usa otras aplicaciones.

Si no tiene acceso al servicio Visual Radio, es posible que los operadores y estaciones de radio en su área no admitan Visual Radio. Es posible que el servicio Visual Radio no esté disponible en todas las áreas y países.

Puede realizar y recibir llamadas como de costumbre mientras escucha la radio. Cuando hay una llamada activa se enmudece el volumen de la radio.

La radio selecciona la banda de frecuencia usada tomando como base la información de país recibida de la red. Si esta información no está disponible, es posible que se le solicite seleccionar la región en la que se encuentra o puede seleccionar la región de las configuraciones de Visual Radio.

Escuchar la radio

La calidad de la transmisión de radio depende de la cobertura de la estación de radio en su área.

La radio FM depende de una antena además de la antena del dispositivo móvil. Para que la radio FM funcione adecuadamente, es preciso conectar un auricular o accesorio compatible al dispositivo.

Pulse y seleccione **Radio**. Para iniciar una búsqueda de una estación, seleccione o y pulse el joystick. Para cambiar la frecuencia manualmente, seleccione **Opciones** > **Ajuste manual**.

Si ya ha guardado estaciones de radio, seleccione o en la pantalla y pulse el joystick para ir a la estación guardada siguiente o anterior o pulse la tecla numérica correspondiente para seleccionar la ubicación de memoria de la estación.

Para ajustar el volumen, mantenga pulsada o en el lado izquierdo del dispositivo. Para escuchar la radio usando el altavoz manos libres, seleccione **Opciones** > **Activar altavoz**.

Advertencia: Escuche música a un nivel moderado. La exposición continua a un nivel de volumen alto puede perjudicar su capacidad de audición. No sostenga el dispositivo cerca del oído si está utilizando el altavoz, ya que el volumen puede resultar muy alto.

Para ver las estaciones disponibles según su área, seleccione **Opciones** > **Directorio estaciones** (servicio de red).

Para guardar la estación actualmente sintonizada en su lista de estaciones, seleccione **Opciones** > **Guardar estación**. Para abrir la lista de emisoras guardadas, seleccione **Opciones** > **Estaciones**.

Para regresar al modo standby y dejar que continúe la reproducción de radio FM en segundo plano, seleccione **Opciones** > **Reproducir en 2° plano**.

Ver contenido visual

Para verificar la disponibilidad y costos y para suscribirse al servicio, comuníquese con su operador de red o proveedor de servicios.

Para ver el contenido visual disponible para una estación sintonizada, seleccione u **Opciones** > **Iniciar servicio visual**. Si el servicio de ID visual no ha sido guardado para la estación, ingréselo o seleccione **Recuperar** para buscarlo en el directorio de estaciones (servicio de red).

Cuando la conexión al servicio visual ha sido establecida, la pantalla muestra el contenido visual actual.

Para ajustar las configuraciones de pantalla para el contenido visual, seleccione **Opciones** > **Config. de pantalla**.

Estaciones guardadas

Puede guardar hasta veinte estaciones en la radio. Para abrir su lista de estaciones, seleccione **Opciones** > **Estaciones**.

Para escuchar una estación guardada, seleccione **Opciones** > **Estación** > **Escuchar**. Para ver el contenido visual para una estación con el servicio Visual Radio, seleccione **Opciones** > **Estación** > **Iniciar servicio visual**.

Para cambiar los detalles de la estación, seleccione **Opciones** > **Estación** > **Editar**.

Configuraciones

Seleccione **Opciones** > **Configuraciones** y alguna de las siguientes opciones:

Tono de inicio: seleccione si se emite un tono al abrir la aplicación.

Servicio de inicio auto.: seleccione **Sí** para que el servicio Visual Radio comience automáticamente al seleccionar una estación guardada que ofrece el servicio visual.

Punto de acceso: seleccione el punto de acceso usado para la conexión de datos. No necesita un punto de acceso para usar la aplicación como radio FM tradicional.

Región actual: seleccione la región donde se encuentra. Esta configuración se muestra sólo si no había cobertura de la red cuando se inició la aplicación.

Mensajería

Pulse y seleccione **Mensajes**. En **Mensajes**, puede crear, enviar, recibir, ver, editar y organizar mensajes de texto, mensajes multimedia, mensajes de e-mail y mensajes de texto especiales que contienen datos. También puede recibir mensajes y datos usando la conectividad Bluetooth, recibir mensajes de servicio Web, mensajes de difusión celular y enviar comandos de servicio.

Nota: Sólo los dispositivos con funciones compatibles pueden recibir y mostrar mensajes multimedia. La apariencia de un mensaje multimedia puede variar según el dispositivo receptor.

Cuando abre **Mensajes**, puede ver la función **Crear mensaje** y las siguientes carpetas:

Buzón entrada: aquí se guardan los mensajes recibidos, con la excepción de los mensajes de e-mail y de difusión celular. Los mensajes de e-mail son guardados en **Buzón de correo**.

Mis carpetas: para organizar sus mensajes en carpetas.

Buzón de correo: conéctese a su buzón de correo remoto para recuperar sus nuevos mensajes de e-mail o ver sin conexión los mensajes de e-mail que ya había recuperado. Consulte "E-mail", pág. 193.

Borrador: aquí se guardan los borradores de mensajes que no han sido enviados.

Enviados: aquí se guardan los últimos veinte mensajes que han sido enviados, con la excepción de los mensajes enviados usando la conectividad Bluetooth. Para cambiar la cantidad de mensajes para guardar, consulte "Otros configuraciones", pág. 195.

Buzón salida: aquí se guardan temporalmente los mensajes a la espera de su envío.

Ejemplo: Los mensajes se guardan en el buzón de salida cuando su dispositivo está fuera del área de cobertura de la red. También puede programar los mensajes de e-mail para ser enviados la próxima vez que se conecte a su buzón de correo remoto.

Informes: puede solicitar a la red que le envíe un informe de entrega de los mensajes de texto y mensajes multimedia que ha enviado (servicio de red). Puede que no sea posible recibir un informe de entrega de un mensaje multimedia enviado a una dirección de e-mail.

Sugerencia: Cuando tiene abierta alguna de las carpetas predeterminadas, para intercambiar entre las carpetas, mueva el joystick a la derecha o izquierda.

Para ingresar y enviar solicitudes de servicio (también denominadas comandos USSD), como comandos de activación para servicios de red, a su proveedor de servicios, seleccione **Opciones** > **Comando servicio** en la vista principal de **Mensajes**.

Difusión celular (servicio de red) le permite recibir de su proveedor de servicios mensajes sobre varios temas, por ejemplo sobre el clima o tráfico. Para conocer los temas y números de temas relevantes disponibles, comuníquese con su proveedor de servicios. En la vista principal de **Mensajes**, seleccione **Opciones** > **Difusión celular**. En la vista principal puede ver el estado, número o nombre de un tema y si ha sido marcado (⚑) para el seguimiento.

Los mensajes de difusión celular no se pueden recibir en redes UMTS. Una conexión de paquetes de datos puede impedir la recepción de difusión celular.

Escribir texto

ABC, **abc** y **Abc** indican el modo de caracteres seleccionado. **123** indica el modo numérico.

Ingreso de texto tradicional

El indicador aparece en la esquina superior derecha de la pantalla cuando está escribiendo texto usando el ingreso de texto tradicional.

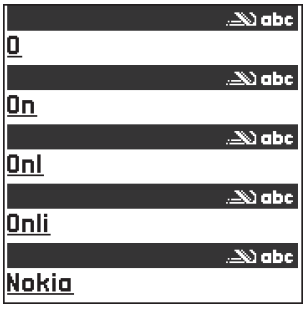
- Pulse una tecla numérica (de **1** a **9**) varias veces hasta que aparezca el carácter deseado. Observe que hay más caracteres disponibles para una tecla numérica que los impresos en la tecla.
- Para insertar un número, mantenga pulsada la tecla numérica.
- Para intercambiar entre los modos de ingreso de texto y numérico, mantenga pulsada **#**.
- Si la letra siguiente está en la misma tecla que la letra actual, espere hasta que el cursor aparezca (o mueva rápidamente el joystick hacia la derecha para finalizar el tiempo de espera), e ingrese la letra.
- Para borrar un carácter, pulse ⌫. Mantenga pulsada ⌫ para borrar más de un carácter.
- Los signos de puntuación más comunes están disponibles en **1**. Pulse **1** varias veces para obtener el signo de puntuación deseado.

  Pulse ***** para abrir una lista de caracteres especiales. Use el joystick para desplazarse por la lista y pulse **Selecc.** para seleccionar un carácter.
- Para insertar un espacio, pulse **0**. Para mover el cursor al renglón siguiente, pulse **0** tres veces.
- Para intercambiar entre los diferentes modos de caracteres, pulse **#**.

Ingreso de texto predictivo: texto predictivo

Puede ingresar cualquier letra con una sola pulsación de tecla. El ingreso de texto predictivo está basado en un diccionario incorporado al cual se le puede agregar palabras. aparece en la esquina superior derecha de la pantalla cuando escribe texto usando el ingreso de texto predictivo.

1 Para activar el ingreso de texto predictivo, pulse y seleccione **Activar texto predictivo**. Esto activa el ingreso de texto predictivo para todos los editores en el dispositivo.



2 Para escribir la palabra deseada, pulse las teclas de **2** a **9**. Pulse cada tecla sólo una vez por letra. Por ejemplo, para escribir "Nokia" cuando el diccionario español está seleccionado, pulse **6** para N, **6** para o, **5** para k, **4** para i y **2** para a.

La palabra sugerida cambia tras cada pulsación de tecla.

3 Cuando termina de escribir la palabra correctamente, desplácese a la derecha para confirmarla o pulse **0** para agregar un espacio.

Si la palabra no es la correcta, pulse * varias veces para ver cada coincidencia o pulse y seleccione **Texto predictivo** > **Coincidencias**.

Si el signo **?** aparece después de la palabra, significa que la palabra no está en el diccionario. Para agregar una palabra al diccionario, seleccione **Escribir**, ingrese la palabra usando el ingreso de texto tradicional y seleccione **Aceptar**. La palabra es agregada al diccionario. Cuando el diccionario está lleno, una palabra nueva reemplazará a la más antigua.

4 Escriba la siguiente palabra.

Sugerencia: Para activar o desactivar el ingreso de texto predictivo, pulse # dos veces rápidamente.

Sugerencias sobre el uso del ingreso de texto

Para insertar un número en el modo de ingreso de texto, mantenga pulsada la tecla numérica deseada.

Para intercambiar entre los diferentes modos de caracteres, pulse #.

Para borrar un carácter, pulse C. Mantenga pulsada C para borrar más de un carácter.

Mantenga pulsada * para abrir una lista de los caracteres especiales.

Sugerencia: Para insertar varios caracteres especiales a la vez, pulse **5** cuando seleccione cada carácter.

Los signos de puntuación más comunes están disponibles en **1**. Pulse **1** y después * varias veces para buscar el signo de puntuación deseado.

Sugerencia: El ingreso de texto predictivo trata de adivinar el signo de puntuación de uso frecuente (**.,?!'**) necesario. El orden y disponibilidad de los signos de puntuación dependen del idioma del diccionario.

Pulse * varias veces para ver una a una las palabras similares que el diccionario encontró.

Pulse , seleccione **Texto predictivo** y desplácese a la derecha para seleccionar alguna de las siguientes opciones:

Coincidencias: para ver una lista de palabras que corresponden a sus pulsaciones de tecla.

Insertar palabra: para agregar una palabra al diccionario usando el ingreso de texto tradicional. Cuando el diccionario está lleno, una palabra nueva reemplaza a la más antigua.

Editar palabra: para editar la palabra usando el ingreso de texto tradicional. Está disponible si la palabra está activa (subrayada).

Sugerencia: Cuando pulsa , las siguientes opciones aparecen (dependiendo del modo de edición): **Texto predictivo** (ingreso de texto predictivo), **Modo alfabético** (ingreso de texto tradicional), **Modo numérico**, **Cortar** (si se ha seleccionado texto), **Copiar** (si se ha seleccionado texto), **Pegar** (cuando ya se ha cortado o copiado texto), **Insertar número**, **Insertar símbolo** y **Idioma de escritura:** (cambia el idioma de ingreso para todos los editores del teléfono).

Escribir palabras compuestas

Escriba la primera parte de la palabra compuesta, para confirmarla, mueva el joystick a la derecha. Escriba la segunda parte de la palabra compuesta. Para completar la palabra compuesta, pulse **0** para agregar un espacio.

Desactivar el ingreso de texto predictivo

Pulse y seleccione **Texto predictivo** > **Desactivar** para desactivar el ingreso de texto predictivo para todos los editores del teléfono.

Copiar texto en el portapapeles

Para seleccionar letras y palabras, mantenga pulsada . Al mismo tiempo, mueva el joystick a la izquierda o derecha. A medida que la selección se mueve, el texto aparece subrayado.

5 Para copiar el texto al portapapeles, mientras mantiene pulsada , seleccione **Copiar**.

6 Para insertar texto en un documento, mantenga pulsada y seleccione **Pegar** o pulse una vez y seleccione **Pegar**.

Para seleccionar renglones de texto, mantenga pulsada . Al mismo tiempo, mueva el joystick hacia arriba o abajo.

Para borrar el texto seleccionado del documento, pulse .

Escribir y enviar mensajes

La apariencia de un mensaje multimedia puede variar, dependiendo del dispositivo receptor.

Las protecciones de copyright pueden impedir la copia, modificación, transferencia o desvío de imágenes, música (incluso tonos de timbre) y otro contenido.

Sugerencia: Puede comenzar a crear un mensaje desde cualquier aplicación que tenga la opción **Enviar**. Seleccione un archivo (imagen o texto) para agregarlo al mensaje y seleccione **Opciones** > **Enviar**.

Antes de poder crear un mensaje multimedia o escribir un e-mail, deberá tener activadas las configuraciones de conexión correctas. Consulte "Recibir configuraciones MMS y de e-mail", pág. 185 y "E-mail", pág. 193.

La red inalámbrica puede limitar el tamaño de los mensajes MMS. Si la imagen insertada excede este límite, el dispositivo puede reducir el tamaño para poder enviarla por MMS.

1 Seleccione **Crear mensaje** y alguna de las siguientes opciones:

Mensaje corto: para enviar un mensaje de texto

Mensaje multimedia: para enviar un mensaje multimedia (MMS)

E-mail: para enviar un e-mail. Si no ha configurado su cuenta de e-mail, se le indicará que lo haga. Para crear las configuraciones de e-mail con la guía de buzón de correo, seleccione **Iniciar**.

2 En el campo **Para**, pulse el joystick para seleccionar los destinatarios o grupos de contactos o ingresar el número telefónico o dirección de e-mail del destinatario. Pulse * para agregar un punto y coma (**;**) para separar a los destinatarios. También puede copiar y pegar el número o dirección del portapapeles.

Sugerencia: Desplácese a un contacto y pulse el joystick para marcarlo. Puede marcar varios destinatarios a la vez.

3 En el campo **Asunto**, ingrese el asunto del mensaje multimedia o e-mail. Para cambiar los campos visibles, seleccione **Opciones** > **Campos de dirección**.

4 Mueva el joystick hacia abajo para desplazarse al campo de mensaje y escribir el mensaje. Para usar una plantilla, seleccione **Opciones** > **Insertar** o **Insertar objeto** > **Plantilla**.

5 Para agregar un objeto multimedia al mensaje multimedia, seleccione **Opciones** > **Insertar objeto** > **Imagen**, **Clip de sonido** o **Videoclip**. Cuando se agrega sonido, el icono aparece.

No puede enviar videoclips guardados en el formato .mp4 en un mensaje multimedia. Para cambiar el formato en el cual se guardan los videos grabados, consulte "Configuraciones de video", pág. 171.

6 Para tomar una nueva foto o grabar sonido o video para un mensaje multimedia, seleccione **Insertar nuevo** > **Imagen**, **Clip de sonido** o **Videoclip**. Para insertar una nueva diapositiva al mensaje, seleccione **Diapositiva**.

Para ver una vista previa del mensaje multimedia, seleccione **Opciones** > **Vista previa**.

7 Para agregar un archivo adjunto a un e-mail, seleccione **Opciones** > **Insertar** > **Imagen**, **Clip de sonido**, **Videoclip** o **Nota**. Los archivos adjuntos de e-mail son indicados por .

Sugerencia: Para enviar archivos adjuntos además de sonidos y notas, abra la aplicación indicada y seleccione **Enviar** > **Vía e-mail**, si está disponible.

8 Para enviar el mensaje, seleccione **Opciones** > **Enviar** o pulse .

Su dispositivo admite el envío de mensajes de texto que exceden el límite de caracteres para un solo mensaje. Los mensajes más extensos son enviados como una serie de dos o más mensajes. Es posible que su proveedor de servicios le cobre los cargos correspondientes. Los caracteres que llevan tilde u otros signos y los caracteres de otros idiomas, como el chino, usan más espacio, de tal manera que se limita la cantidad de caracteres que se puede enviar en un solo mensaje. En la barra de navegación, puede ver el indicador de la longitud del mensaje contando hacia atrás. Por ejemplo, 10 (2) significa que aún puede agregar 10 caracteres para enviar el texto como dos mensajes de texto.

Los mensajes de e-mail van automáticamente al **Buzón salida** antes de ser enviados. Si el envío no se logra, el e-mail queda en el **Buzón salida** con el estado **Fallido**.

Sugerencia: En **Mensajes** también puede crear presentaciones y enviarlas en un mensaje multimedia. En la vista del editor de mensajes multimedia, seleccione **Opciones** > **Crear presentación** (sólo aparece si **Modo creación MMS** está configurado en **Dirigido** o **Libre**). Consulte "Mensajes multimedia", pág. 192.

Sugerencia: Cuando envía un mensaje multimedia a una dirección de e-mail o a un dispositivo que admite la recepción de imágenes grandes, use el tamaño grande de imagen. Si no está seguro si el dispositivo receptor o si la red no admite el envío de imágenes grandes, se recomienda que use un tamaño de imagen más pequeño o un clip de sonido que no exceda quince segundos. Para cambiar la configuración, seleccione **Opciones** > **Configuraciones** > **Mensaje multimedia** > **Tamaño de imagen** en la vista principal de **Mensajes**.

Nota: Los indicadores del mensaje enviado indican que el mensaje ha sido enviado por su dispositivo al número del centro de mensajes en él programado. Esto no indica que el mensaje haya sido recibido por el destinatario. Para obtener más información sobre los servicios de mensajería, comuníquese con su proveedor de servicios.

Recibir configuraciones MMS y de e-mail

Puede recibir las configuraciones en un mensaje de texto enviadas por su operador de red o proveedor de servicios. Consulte "Datos y configuraciones", pág. 186.

Para obtener más información sobre disponibilidad y suscripción a los servicios de datos, comuníquese con su operador de red o proveedor de servicios. Siga las instrucciones provistas por su proveedor de servicios.

Ingresar manualmente las configuraciones MMS:

1 Pulse y seleccione **Herramient.** > **Config.** > **Conexión** > **Puntos de acceso** y defina las configuraciones para un punto de acceso de mensajería multimedia. Consulte "Conexión", pág. 229.

2 Pulse y seleccione **Mensajes** > **Opciones** > **Configuraciones** > **Mensaje multimedia** > **Punto acceso en uso** y el punto de acceso que creó como la conexión preferida. También consulte "Mensajes multimedia", pág. 192.

Antes de poder enviar, recibir, recuperar, responder y reenviar e-mail, deberá hacer lo siguiente:

- Configurar correctamente un punto de acceso a Internet (IAP). Consulte "Conexión", pág. 229.

- Determinar correctamente sus configuraciones de e-mail. Si selecciona **Buzón de correo** en la vista principal de **Mensajes** y no ha configurado su cuenta de e-mail, se le indicará que lo haga. Para crear las configuraciones de e-mail con la guía de buzón de correo, seleccione **Iniciar**. Consulte "E-mail", pág. 193.

  Deberá tener una cuenta de e-mail por separado. Siga las instrucciones provistas por su buzón de correo remoto y de su proveedor de servicios de Internet (ISP).

Buzón de entrada: recibir mensajes

Buzón entrada, esta carpeta tiene los siguientes indicadores:

mensaje de texto no leído.

mensaje multimedia no leído.

datos recibidos a través de la conectividad Bluetooth.

Cuando recibe un mensaje, y el texto **1 mensaje nuevo** aparecerán en el modo standby. Para abrir el mensaje, seleccione **Mostrar**. Para abrir un mensaje en **Buzón entrada**, desplácese al mensaje y pulse el joystick.

Para responder a un mensaje recibido, seleccione **Opciones** > **Responder**.

Mensajes multimedia

Importante: Tenga precaución al abrir los mensajes. Los objetos de los mensajes multimedia pueden contener software malicioso o dañar el dispositivo o PC.

Puede recibir una notificación indicando que un mensaje multimedia se encuentra a la espera en el centro de mensajes multimedia. Para iniciar una conexión de paquetes de datos para recuperar el mensaje en su dispositivo, seleccione **Opciones** > **Recuperar**.

Cuando abre un mensaje multimedia ( ), podría ver una imagen y un mensaje. aparece si hay sonido incluido, o si hay video incluido. Para reproducir el sonido o video, seleccione el indicador.

Para ver los tipos de objetos multimedia incluidos en el mensaje multimedia, abra el mensaje y seleccione **Opciones** > **Objetos**.

Si el mensaje incluye una presentación multimedia, aparece . Para reproducir la presentación, seleccione el indicador.

Datos y configuraciones

Nota: Está función no esta disponible en todas los países, consulte la disponibilidad de esta función con su operador local.

Su dispositivo puede recibir varios tipos de mensajes con datos ( ):

Mensaje de configuración: puede recibir configuraciones de su operador de red, proveedor de servicios o departamento de administración de información de la compañía en un mensaje de configuración. Para guardar las configuraciones, seleccione **Opciones** > **Guardar todos**.

Tarjeta de visita: para guardar información en **Contactos**, seleccione **Opciones** > **Guardar tarj. visita**. Los certificados o archivos de sonido adjuntos a la tarjeta de negocios no serán guardados.

Tono timbre: para guardar el tono de timbre, seleccione **Opciones** > **Guardar**.

Logo operad.: para que el logotipo aparezca en el modo standby en lugar de la identificación del operador de red, seleccione **Opciones** > **Guardar**.

Entr. agenda: para guardar la invitación, seleccione **Opciones** > **Guardar en Agenda**.

Mensaje Web: para guardar el favorito en la lista de favoritos en Web, seleccione **Opciones** > **Agregar a favoritos**. Si el mensaje contiene configuraciones de puntos de acceso y favoritos, para guardar los datos, seleccione **Opciones** > **Guardar todos**.

Notif. e-mail: la nota le indica cuántos mensajes de e-mail hay en su buzón de correo remoto. Una notificación extendida puede contener información más detallada.

Mensajes de servicio Web

Los mensajes de servicio Web ( ) son notificaciones (por ejemplo, titulares de periódico) y pueden contener un mensaje de texto o un enlace. Para obtener más información sobre disponibilidad y suscripción, comuníquese con su proveedor de servicios.

Mis carpetas

En **Mis carpetas**, puede organizar sus mensajes en carpetas, crear carpetas nuevas y renombrar y eliminar las carpetas.

Sugerencia: Puede usar textos en la carpeta de plantillas para no volver a escribir mensajes que envía con frecuencia.

Buzón de correo

Si selecciona **Buzón de correo** y no ha configurado su cuenta de e-mail, se le indicará que lo haga. Para crear las configuraciones de e-mail con la guía de buzón de correo, seleccione **Iniciar**. También consulte "E-mail", pág. 193.

Cuando crea un buzón de correo nuevo, el nombre que usted da al buzón reemplaza el nombre **Buzón de correo** en la vista principal de **Mensajes**. Puede crear hasta seis buzones de correo.

Abrir el buzón de correo

Cuando abre el buzón de correo, el dispositivo le pregunta si desea conectarse al buzón de correo (**¿Conectar con buzón de correo?**).

Para conectarse con el buzón de correo y recuperar encabezamientos de e-mail o mensajes nuevos, seleccione **Sí**. Cuando ve mensajes en línea, está conectado continuamente a un buzón de correo remoto usando una conexión de datos. También consulte "Conexión", pág. 229.

Para ver los mensajes de e-mail recuperados previamente mientras está desconectado, seleccione **No**. Cuando ve los mensajes de e-mail al estar desconectado, su dispositivo no está conectado al buzón de correo remoto.

Recuperar mensajes de e-mail

Si está desconectado, seleccione **Opciones** > **Conectar** para iniciar la conexión a un buzón de correo remoto.

Importante: Tenga precaución al abrir los mensajes. Los mensajes de e-mail pueden contener software malicioso o dañar el dispositivo o PC.

1 Cuando tenga una conexión abierta con un buzón de correo remoto, seleccione **Opciones** > **Recuperar e-mail** y alguna de las siguientes opciones:

Nuevos: para recuperar todos los mensajes de e-mail nuevos

Seleccionados: para recuperar sólo los mensajes de e-mail que han sido marcados

Todos: para recuperar todos los mensajes del buzón de correo

Para detener la recuperación de mensajes, seleccione **Cancelar**.

2 Tras recuperar los mensajes de e-mail, puede seguir viéndolos en línea o seleccionar **Opciones** > **Desconectar** para cerrar la conexión y ver los mensajes de e-mail mientras está desconectado.

Indicadores de estado de e-mail:

El nuevo e-mail (modo desconectado o en línea) no ha sido recuperado por su dispositivo.

El nuevo e-mail ha sido recuperado por su dispositivo.

El mensaje de e-mail ha sido leído y no ha sido recuperado por su dispositivo.

El mensaje e-mail ha sido leído.

El encabezamiento de e-mail ha sido leído y el contenido del mensaje ha sido borrado del dispositivo.

3 Para abrir un mensaje de e-mail, pulse el joystick. Si el mensaje de e-mail no ha sido recuperado y está desconectado, se le pregunta si desea recuperar este mensaje del buzón de correo.

Para ver archivos adjuntos de e-mail, abra un mensaje que tenga el indicador de archivo adjunto y seleccione **Opciones** > **Archivos adjuntos**. Si el archivo adjunto tiene un indicador oscuro, éste no ha sido cargado al dispositivo, seleccione **Opciones** > **Recuperar**. En la vista de **Archivos adjuntos**, puede recuperar, abrir, guardar o eliminar el archivo adjunto. También puede enviar archivos adjuntos usando la conectividad Bluetooth.

Sugerencia: Si su buzón de correo usa el protocolo IMAP4, puede definir cuántos mensajes recuperará y si desea recuperar archivos adjuntos. Con el protocolo POP3, las opciones son **Sólo encabezados**, **Limite tamaño**, o **Mens. y arch. adj.**

Recuperar mensajes de e-mail automáticamente

Para recuperar mensajes de e-mail automáticamente, seleccione **Opciones** > **Config. de e-mail** > **Recuperación auto.** > **Recuperac. encabez.** > **Siempre activada** o **Sólo red propia** y defina cuándo y con cuánta frecuencia se recuperarán los mensajes.

La recuperación automática de mensajes de e-mail puede aumentar sus costos de llamadas debido al tráfico de datos.

Borrar mensajes de e-mail

Para eliminar del dispositivo el contenido de un mensaje de e-mail mientras lo tiene retenido en el buzón de correo remoto, seleccione **Opciones** > **Eliminar**. En **Elim. mens. de:**, seleccione **Sólo del teléfono**.

El teléfono crea una copia de los encabezamientos de e-mail en el buzón de correo remoto. Aunque borre el contenido del mensaje, el encabezamiento del e-mail queda en su dispositivo. Si también desea borrar el encabezamiento, primero deberá descartar el mensaje de e-mail de su buzón de correo remoto y después establecer una conexión con su dispositivo al buzón de correo remoto para actualizar el estado.

Para borrar un e-mail del dispositivo y del buzón de correo remoto, seleccione **Opciones** > **Eliminar**. En **Elim. mens. de:**, seleccione **Teléfono y servidor**.

Si está desconectado, primero se borra el e-mail de su dispositivo. En la próxima conexión al buzón de correo remoto, el mensaje será borrado del buzón. Si está usando el protocolo POP3, sólo podrá retirar los mensajes marcados para ser borrados cuando cierre la conexión al buzón de correo remoto.

Para cancelar la eliminación de un e-mail del dispositivo y del servidor, desplácese a un e-mail que ha sido marcado para ser borrado durante la próxima conexión ( ) y seleccione **Opciones** > **Deshacer eliminar**.

Desconectarse del buzón de correo

Cuando esté en línea, seleccione **Opciones** > **Desconectar** para finalizar la conexión de paquetes de datos al buzón de correo remoto.

Sugerencia: Puede dejar abierta la conexión del buzón de correo remoto y los mensajes de e-mail nuevos (**Sólo encabezados** como predeterminado) serán recuperados automáticamente en su dispositivo (sólo si la función IMAP IDLE es admitida por su servidor). Para dejar la aplicación de mensajería abierta en segundo plano, pulse dos veces. Dejar la conexión abierta puede aumentar sus costos de llamadas por el tráfico de datos.

Ver mensajes de e-mail mientras está desconectado

La próxima vez que abra **Buzón de correo** y desee ver y leer los mensajes de e-mail mientras está desconectado, responda **No** la solicitud **¿Conectar con buzón de correo?**. Puede leer los encabezamientos de e-mail que recuperó anteriormente, los mensajes de e-mail recuperados o ambos. También puede escribir mensajes nuevos, contestar o reenviar e-mail para enviarlos la próxima vez que se conecte al buzón de correo.

Buzón de salida: mensajes a la espera de ser enviados

Buzón salida es un lugar de almacenamiento provisional para mensajes a la espera de ser enviados.

Los estados de los mensajes en **Buzón salida** son los siguientes:

Enviando: se está estableciendo una conexión y el mensaje se está enviando.

En espera/**En cola**: el mensaje será enviado cuando mensajes anteriores del mismo tipo hayan sido enviados.

Reenviar a las (tiempo): el dispositivo tratará de enviar el mensaje otra vez tras el período de espera. Seleccione **Enviar** para reiniciar el envío inmediatamente.

Aplazado: puede programar los documentos para estar en espera mientras están en **Buzón salida**. Desplácese a un mensaje que está siendo enviado y seleccione **Opciones** > **Aplazar envío**.

Fallido: se llegó a la cantidad máxima de intentos de envío. No se logró el envío. Si estaba tratando de enviar un mensaje de texto, abra el mensaje y verifique que las opciones de envío sean las correctas.

Ejemplo: Los mensajes se guardan en el buzón de salida, por ejemplo, cuando su dispositivo está fuera del área de cobertura de la red. También puede programar los mensajes de e-mail para ser enviados la próxima vez que se conecte a su buzón de correo remoto.

Ver mensajes en la tarjeta SIM

Antes de ver los mensajes SIM, deberá copiarlos en una carpeta en su dispositivo.

1 En la vista principal de **Mensajes**, seleccione **Opciones** > **Mensajes SIM**.

2 Seleccione **Opciones** > **Marcar/Anul. marcar** > **Marcar** o **Marcar todo** para marcar mensajes.

3 Seleccione **Opciones** > **Copiar**. Se abre una lista de carpetas.

4 Para comenzar a copiar, seleccione la carpeta y **Aceptar**. Para ver los mensajes, abra la carpeta.

Configuraciones de mensajes

Rellene todos los campos marcados con **Debe definirse** o con un asterisco rojo. Siga las instrucciones provistas por su proveedor de servicios. También puede obtener las configuraciones de su proveedor de servicios en un mensaje de configuración.

Algunos o todos los centros de mensajes, puntos de acceso u otras configuraciones pueden estar preconfigurados en su dispositivo por su proveedor de servicios u operador de red y es posible que no pueda cambiarlos, crearlos, editarlos ni retirarlos.

Mensajes de texto

Pulse y seleccione **Mensajes** > **Opciones** > **Configuraciones** > **Mensaje corto** y alguna de las siguientes opciones:

Centros de mensajes: vea una lista de todos los centros de mensajes de texto que han sido definidos.

Centro mens. en uso: seleccione el centro de mensajes que usará para entregar mensajes de texto.

E-mail de SMS: para enviar y recibir mensajes de texto de direcciones de e-mail, ingrese el gateway de e-mail y el centro de mensajes de e-mail.

Codificac. caracteres: para usar la conversión de caracteres para otro sistema de codificación cuando esté disponible, seleccione **Soporte reducido**.

Informe de recepción: seleccione si la red enviará informes de entrega de sus mensajes (servicio de red).

Validez de mensajes: seleccione por cuánto tiempo el centro de mensajes reenviará su mensaje si el primer intento falla (servicio de red). Si no se puede encontrar al destinatario dentro del período de validez, el mensaje es borrado del centro de mensajes.

Mens. enviado como: cambie esta opción solamente si está seguro de que su centro de mensajes puede convertir mensajes de texto en estos otros formatos. Comuníquese con su operador de red.

Conexión preferida: seleccione la conexión usada para enviar mensajes de texto: red GSM o paquetes de datos, si es admitido por la red. Consulte "Conexión", pág. 229.

Resp. mismo centro: seleccione si desea enviar el mensaje de respuesta usando el mismo número de centro de mensajes de texto (servicio de red).

Agregar y editar centros de mensajes de texto

Tenga en cuenta que si cambia su tarjeta SIM, la lista de centros de mensajes de texto disponibles es actualizada según los centros en la tarjeta SIM nueva. Los centros de mensajes también pueden estar preconfigurados en su dispositivo por su proveedor de servicios y es posible que no pueda crearlos, editarlos ni borrarlos.

1 Seleccione **Centros de mensajes** > **Opciones** > **Nuevo centro mens.** o **Editar**.

2 Pulse el joystick, escriba un nombre para el centro de mensajes y seleccione **Aceptar**.

3 Desplácese hacia abajo, pulse el joystick y escriba el número del centro de mensajes de texto. Usted recibirá este número de su proveedor de servicios.

4 Seleccione **Aceptar**.

5 Para usar las configuraciones nuevas, vuelva a la vista de configuraciones. Desplácese a **Centro mens. en uso** y seleccione el centro de mensajes nuevo.

Mensajes multimedia

Pulse y seleccione **Mensajes** > **Opciones** > **Configuraciones** > **Mensaje multimedia** y alguna de las siguientes opciones:

Tamaño de imagen: defina el tamaño de la imagen en un mensaje multimedia: **Original** (aparece sólo cuando **Modo creación MMS** está configurado en **Dirigido** o **Libre**), **Pequeño** o **Grande**.

Modo creación MMS: si selecciona **Dirigido**, el dispositivo le informa si usted trata de enviar un mensaje que posiblemente no sea admitido por el destinatario. Seleccione **Restringido**, el dispositivo le impedirá enviar mensajes que no sean admitidos.

Punto acceso en uso: seleccione el punto de acceso usado como la conexión preferida para el centro de mensajes multimedia.

Recuperación multim.: seleccione cómo desea recibir los mensajes multimedia. Para recibir mensajes multimedia automáticamente en su red doméstica, seleccione **Auto. en red domést.** Fuera de su red doméstica, recibirá una notificación indicando que hay un mensaje multimedia que puede recuperar del centro de mensajes multimedia.

Cuando está fuera de su red doméstica, el envío y recepción de mensajes multimedia pueden costar más.

Si selecciona **Recuperación multim.** > **Siempre automático**, su teléfono activa automáticamente una conexión de paquetes de datos para recuperar el mensaje fuera y dentro de su red doméstica.

Mensajes anónimos: seleccione si desea rechazar los mensajes de un remitente anónimo.

Recibir publicidad: defina si desea recibir publicidad en mensajes multimedia.

Informe de entrega: seleccione si desea que el estado del mensaje enviado aparezca en el registro (servicio de red). Puede que no sea posible recibir un informe de entrega de un mensaje multimedia enviado a una dirección de e-mail.

Denegar envío infs.: seleccione si desea impedir que su dispositivo envíe informes de entrega de los mensajes multimedia recibidos.

Validez de mensajes: seleccione por cuánto tiempo el centro de mensajes reenviará su mensaje si el primer intento no se logra (servicio de red). Si no se puede encontrar al destinatario dentro del período de validez, el mensaje es borrado del centro de mensajes.

Sugerencia: También puede obtener las configuraciones multimedia y de e-mail de su proveedor de servicios a través de un mensaje de configuración. Comuníquese con su proveedor de servicios para obtener más información. Consulte "Datos y configuraciones", pág. 186.

E-mail

Pulse y seleccione **Mensajes** > **Opciones** > **Configuraciones** > **E-mail** y alguna de las siguientes opciones:

Buzones de correo: seleccione un buzón de correo para cambiarle las siguientes configuraciones: **Config. conexión**, **Config. de usuario**, **Config. recuperación** y **Recuperación auto.**

Buzón correo en uso: seleccione el buzón de correo que desea usar para enviar e-mail.

Para crear un buzón de correo nuevo, seleccione **Opciones** > **Nuevo buzón correo** en la vista principal del buzón de correo.

Configuraciones de conexión

Para editar las configuraciones para el e-mail que recibe, seleccione **E-mail entrante** y alguna de las siguientes opciones:

Nombre de usuario: ingrese su nombre de usuario que le suministró su proveedor de servicios.

Contraseña: ingrese su contraseña. Si deja este campo sin rellenar, se le indicará que indique su contraseña cuando trate de conectarse a su buzón de correo remoto.

Servidor correo entr.: ingrese la dirección IP o nombre de host del servidor de correo que recibe su e-mail.

Punto acceso en uso: seleccione un punto de acceso a Internet (IAP). Consulte "Conexión", pág. 229.

Nombre de buzón: ingrese un nombre para el buzón de correo.

Tipo buzón de correo: define el protocolo de e-mail recomendado por su proveedor de servicios de buzón de correo remoto. Las opciones son **POP3** e **IMAP4**. Esta configuración puede ser seleccionada sólo una vez y no se puede cambiar si ha guardado o salido de las configuraciones de buzón de correo. Si usa el protocolo POP3, los mensajes de e-mail no son actualizados automáticamente cuando está en línea. Para ver los mensajes de e-mail más recientes, deberá desconectarse y establecer una nueva conexión a su buzón de correo.

Seguridad (puertos): seleccione la opción de seguridad usada para asegurar la conexión a su buzón de correo remoto.

Puerto: defina un puerto para la conexión.

Acceso seguro APOP (sólo para POP3): use con el protocolo POP3 para encriptar el envío de contraseñas al servidor de e-mail remoto mientras se conecta al buzón de correo.

Para editar las configuraciones para el e-mail que envía, seleccione **E-mail saliente** y alguna de las siguientes opciones:

Mi dirección e-mail: ingrese la dirección de e-mail que le suministró su proveedor de servicios. Las respuestas a sus mensajes son enviadas a esta dirección.

Servidor correo sal.: ingrese la dirección IP o nombre de host del servidor de correo que envía su e-mail. Es posible que sólo pueda usar el servidor saliente de su operador de red. Comuníquese con su proveedor de servicios para obtener más información.

Punto acceso en uso: seleccione un punto de acceso a Internet (IAP). Consulte "Conexión", pág. 229.

Las configuraciones para **Nombre de usuario**, **Contraseña**, **Seguridad (puertos)** y **Puerto** son similares a las de **E-mail entrante**.

Configuraciones de usuario

Mi nombre: ingrese su nombre aquí. Su nombre reemplaza su dirección de e-mail en el dispositivo del destinatario cuando el destinatario admite esta función.

Enviar mensaje: defina la forma de enviar el e-mail desde su dispositivo. Seleccione **Inmediatamente** para que el dispositivo se conecte al buzón de correo cuando seleccione **Enviar mensaje**. Si selecciona **En sig. conexión**, el e-mail es enviado cuando la conexión al buzón de correo remoto está disponible.

Env. copia a sí mismo: seleccione si desea guardar una copia del e-mail en su buzón de correo remoto y en la dirección definida bajo **Mi dirección e-mail**.

Incluir firma: seleccione si desea adjuntar una firma a sus mensajes de e-mail.

Nvas. alertas e-mail: seleccione si desea recibir las indicaciones de e-mail nuevos, un tono y una nota, cuando reciba nuevo correo en el buzón de correo.

Configuraciones de recuperación

E-mail para recup.: defina las partes que recuperará del mensaje de e-mail: **Sólo encabezados**, **Limite tamaño**, **Mensajes** (sólo para IMAP4) o **Mens. y arch. adj.**

Cantidad para recup.: defina cuántos mensajes de e-mail nuevos son recuperados al buzón de correo.

Ruta carpeta IMAP4 (sólo para IMAP4): defina una ruta para las carpetas que se suscribirán.

Suscripc. de carpeta (sólo para IMAP4): suscríbase a otras carpetas en el buzón de correo remoto y recupere contenido de estas carpetas.

Recuperación automática

Recuperac. encabez.: seleccione si desea que el dispositivo recupere nuevo e-mail automáticamente. Podrá definir cuándo y con cuánta frecuencia, los mensajes serán recuperados.

La recuperación automática de mensajes de e-mail puede aumentar sus costos de llamadas debido al tráfico de datos.

Mensajes de servicio Web

Pulse y seleccione **Mensajes** > **Opciones** > **Configuraciones** > **Mensaje de servicio**. Seleccione si desea recibir mensajes de servicio. Si desea configurar el dispositivo para que active automáticamente el explorador e inicie una conexión de red para recuperar contenido cuando recibe un mensaje de servicio, seleccione **Descargar mensajes** > **Automáticamente**.

Difusión celular

Verifique con su proveedor de servicios los temas disponibles y números de temas asociados. Pulse y seleccione **Mensajes** > **Opciones** > **Configuraciones** > **Difusión celular** y alguna de las siguientes opciones:

Recepción: seleccione si desea recibir mensajes de difusión celular.

Idioma: seleccione los idiomas en los cuales desea recibir mensajes: **Todos**, **Seleccionados** u **Otros**.

Detección de temas: seleccione si el dispositivo busca automáticamente los nuevos números de temas y guarda los números nuevos sin un nombre en la lista de temas.

Otros configuraciones

Pulse y seleccione **Mensajes** > **Opciones** > **Configuraciones** > **Otras** y alguna de las siguientes opciones:

Guardar mens. env.: seleccione si desea guardar una copia de los mensajes de texto, mensajes multimedia o e-mail que envía en la carpeta **Enviados**.

Nº mens. guardados: defina cuántos mensajes enviados se guardarán en la carpeta **Enviados**. El límite predeterminado es de veinte mensajes. Cuando alcance el límite, el mensaje más antiguo será borrado.

Memoria en uso: seleccione la memoria en la cual desea guardar sus mensajes: **Memoria teléfono** o **Disco duro**.

Personal

Web

Pulse y seleccione **Personal** > **Web** para abrir un explorador Web secundario óptimo para explorar sitios Web que no están formateados originalmente para la exploración móvil.

Web conserva el diseño original de página.

Para desplazarse fácilmente dentro de una página, desplácese en cualquier sentido en una página. Una imagen más pequeña de la página aparece con un marco rojo que indica el área actualmente mostrada.

Para ver la imagen en miniatura de la página actual también puede seleccionar **Opciones** > **Opciones de zoom** > **Descripción de página**. Mueva el marco rojo con el joystick y pulse el joystick para regresar a la vista normal.

Para ver las páginas visitadas recientemente como miniaturas, seleccione **Opciones** > **Opc. navegación**.

Para bloquear las ventanas emergentes, seleccione **Opciones** > **Configuraciones** > **Bloq. mens. emerg.**

Para buscar por un texto en tiempo real, seleccione **Opciones** > **Buscar** y comience a ingresar el texto de búsqueda. El texto que coincide aparece mientras ingresa el texto.

Web feeds: puede suscribirse a noticias y blog feeds de distintos sitios Web. La función Web feeds del explorador también alertará al usuario cuando los enlaces feed están disponibles para un sitio Web.

Para obtener más información sobre otras funciones y opciones en Web, consulte "Servicios", pág. 211.

Notas

Pulse y seleccione **Personal** > **Notas** para escribir notas. Puede enviar notas a otros dispositivos compatibles y guardar archivos de texto sencillo (formato .txt) que recibe en **Notas**.

Ir a: agregar accesos directos

Para guardar accesos directos (enlaces a sus imágenes favoritas, notas, favoritos y demás), pulse y seleccione **Personal** > **Ir a**. Los accesos directos se agregan sólo de las aplicaciones individuales. No todas las aplicaciones cuentan con esta función.

1 Seleccione un elemento desde una aplicación a la cual desea agregar un acceso directo.

2 Seleccione **Agregar a 'Ir a'**. Un acceso directo en **Ir a** es actualizado automáticamente si mueve el elemento al que éste indica (por ejemplo, de una carpeta a otra). Para cambiar el identificador en la esquina inferior izquierda del icono de accesos directos, seleccione **Opciones** > **Icono acc. directo**.

Borrar un acceso directo

Seleccione el acceso directo y pulse . Los accesos directos predeterminados **Notas**, **Agenda** y **Buzón entrada** no se pueden borrar.

Cuando elimina una aplicación o documento con un acceso directo en **Ir a**, el icono de acceso directo del elemento eliminado es oscurecido en la vista **Ir a**. El acceso directo puede borrarse la próxima vez que inicie **Ir a**.

MI: mensajería instantánea (chat)

Nota: Está función no esta disponible en todas los países, consulte la disponibilidad de esta función con su operador local.

Esta función depende del proveedor.

Pulse y seleccione **Personal** > **MI**.

Mensajería instantánea (servicio de red) le permite conversar con otras personas usando mensajes instantáneos e incorporarse a foros de discusión (grupos MI) sobre asuntos específicos. Varios proveedores de servicios mantienen servidores MI a los cuales podrá acceder cuando se haya registrado al servicio MI. Los proveedores de servicios varían en los servicios que admiten.

Seleccione **Conversac.** para comenzar o continuar una conversación con un usuario MI; **Contactos MI** para crear, editar o ver el estado en línea de sus contactos MI, **Grupos de MI** para comenzar o continuar una conversación de grupo con múltiples usuarios MI o **Conv. grabadas** para ver una sesión previa de mensajería instantánea que guardó.

Para obtener más información sobre la disponibilidad de servicios, precios, instrucciones y tarifas, comuníquese con su operador de red o proveedor de servicios.

Recibir configuraciones de MI

Guarde las configuraciones necesarias para acceso al servicio que desea usar. Puede recibir las configuraciones en un mensaje de texto especial del operador de red o proveedor de servicios que ofrece el servicio MI. Consulte "Datos y configuraciones", pág. 186. También puede ingresar manualmente las configuraciones. Consulte "Configuraciones de servidor MI", pág. 201.

Conectarse a un servidor MI

1 Para conectarse a un servidor MI en uso, abra **MI** y seleccione **Opciones** > **Iniciar sesión**. Para cambiar el servidor MI en uso y guardar nuevos servidores MI, consulte "Configuraciones de servidor MI", pág. 201.

2 Ingrese su ID y contraseña y pulse el joystick para iniciar la sesión. Solicite a su proveedor de servicios la ID y contraseña para el servidor MI.

3 Para finalizar la sesión, seleccione **Opciones** > **Finalizar sesión**.

Modificar sus configuraciones de MI

Seleccione **Opciones** > **Configuraciones** > **Configuraciones MI** y alguna de las siguientes opciones:

Usar alias (sólo aparece si los grupos MI son admitidos por el servidor): para ingresar un sobrenombre, seleccione **Sí**.

Permitir mensajes de: para permitir todos los mensajes, seleccione **Todos**.

Permitir invitaciones de: para permitir invitaciones sólo de sus contactos MI, seleccione **Sólo contactos MI**. Las invitaciones MI son enviadas por los contactos MI que desean que usted se incorpore a sus grupos.

Vel. despl. mensaje: seleccione la velocidad en la que se muestran los mensajes nuevos.

Ordenar contactos MI: seleccione cómo sus contactos MI serán organizados: **Alfabéticamente** o **Por est. en línea**.

Recarga disp.: para seleccionar cómo actualizar información para saber si sus contactos MI están en línea o desconectados, seleccione **Automático** o **Manual**.

Contactos desconectados: para seleccionar si sus contactos MI con estado desconectado aparecen en la lista de contactos MI.

Buscar grupos y usuarios MI

Para buscar grupos, en la vista **Grupos de MI**, seleccione **Opciones** > **Buscar**. Puede buscar por **Nombre de grupo**, **Tema** y **Miembros** (ID de usuario).

Para buscar a los usuarios en la vista **Contactos MI**, seleccione **Opciones** > **Nuevo contacto MI** > **Buscar desde servidor**. Se puede buscar por **Nombre de usuario**, **Id. de usuario**, **Número de teléfono** y **Dirección de e-mail**.

Incorporarse a grupos MI

La vista **Grupos de MI** muestra una lista de los grupos MI que ha guardado o a los que se ha incorporado recientemente.

Para incorporarse a un grupo MI guardado, pulse el joystick.

Para unirse a un grupo MI que no figura en la lista, pero para el cual sabe la ID de grupo, seleccione **Opciones** > **Unirse a nvo. grupo**.

Para salir del grupo MI, seleccione **Opciones** > **Salir de grupo MI**.

Chat

Cuando se haya unido a un grupo MI, podrá ver los mensajes intercambiados y enviar sus propios mensajes.

Para enviar un mensaje, escriba el mensaje en el campo de editor de mensaje y pulse el joystick.

Para enviar un mensaje privado a un participante, seleccione **Opciones** > **Enviar mens. priv.**.

Para responder a un mensaje privado, seleccione el mensaje y **Opciones** > **Responder**.

Para invitar a contactos MI en línea a incorporarse al grupo MI, seleccione **Opciones** > **Enviar invitación**.

Para evitar la recepción de mensajes de ciertos participantes, seleccione **Opciones** > **Opciones bloqueo** y la opción deseada.

Grabar conversaciones de chat

Para grabar los mensajes intercambiados durante una conversación o mientras está en un grupo MI, seleccione **Opciones** > **Grabar chat**. Para detener la grabación, seleccione **Opciones** > **Parar grabación**. Para ver las conversaciones de chat grabadas, en la vista principal, seleccione **Conv. grabadas**.

Ver e iniciar conversaciones

La vista **Conversac.** muestra una lista de los participantes en la conversación con los cuales está teniendo una conversación. Las conversaciones en curso son automáticamente cerradas cuando sale de **MI**.

Para ver una conversación, desplácese al nombre de un participante y pulse el joystick.

Para continuar la conversación, escriba su mensaje y pulse el joystick.

Para regresar a la lista de conversaciones sin finalizar la conversación, seleccione **Atrás**. Para cerrar la conversación, seleccione **Opciones** > **Finalizar convers.**.

Para iniciar una conversación, seleccione **Opciones** > **Nueva conversación**.

Para guardar el nombre de un participante de conversación en sus contactos MI, seleccione **Opciones** > **Agreg. a contac. MI**.

Para enviar respuestas automáticas a los mensajes entrantes, seleccione **Opciones** > **Activar resp. auto.**. Todavía puede recibir mensajes.

Contactos MI

En la vista **Contactos MI**, puede recuperar las listas de contactos MI del servidor o agregar un contacto MI nuevo a una lista de contactos. Cuando se conecta al servidor, la lista de contactos MI usada por última vez es recuperada automáticamente del servidor.

Para crear un contacto nuevo, seleccione **Opciones** > **Nuevo contacto MI** > **Escriba manualmente**. Rellene los campos **Sobrenombre** y **Id. de usuario** y seleccione **Hecho**.

Para mover un contacto de una lista en el servidor a la lista recuperada, seleccione **Opciones** > **Nuevo contacto MI** > **Mover de otra lista**.

Para cambiar la lista de contactos MI, seleccione **Opciones** > **Cambiar lista cont.**.

Seleccione alguna de las siguientes opciones:

Abrir conversación: para iniciar una conversación nueva o continuar una conversación con el contacto.

Activar seguimiento: para que el dispositivo le notifique cada vez que el contacto MI está en línea o desconectado.

Pertenece a grupos: para ver a cuáles grupos el contacto MI se ha unido.

Cargar dispon. usuarios: para actualizar la información acerca de si el contacto está en línea o desconectado. El estado en línea se muestra con un indicador junto al nombre del contacto. Esta opción no está disponible si configuró **Recarga disp.** como **Automático** en **Configuraciones MI**.

Administrar los grupos MI

Seleccione la vista **Grupos de MI** para ver una lista de los grupos MI que ha guardado o a los que está suscrito.

Desplácese al grupo, seleccione **Opciones** > **Grupo** y alguna de las siguientes opciones:

Guardar: para guardar un grupo no guardado en el cual está participando.

Ver participantes: para ver quién está unido al grupo.

Detalles: para ver la ID de grupo, tema, participantes, administradores del grupo (sólo aparece si tiene derechos de edición), la lista de participantes bloqueados (sólo aparece si tiene derechos de edición) y si se permiten conversaciones privadas en el grupo.

Configuraciones: para ver y editar las configuraciones de grupo MI. Consulte "Crear un grupo MI nuevo", pág. 200.

Crear un grupo MI nuevo

Seleccione **Grupos de MI** > **Opciones** > **Crear nuevo grupo**. Ingrese las configuraciones para el grupo.

Puede editar las configuraciones para un grupo MI si tiene derechos de administrador para el grupo. El usuario quien crea un grupo obtiene automáticamente derechos de administrador.

Nombre del grupo, **Tema del grupo** y **Saludo inicial**: para agregar detalles que los participantes ven cuando se unen al grupo.

Tamaño de grupo: para definir la cantidad máxima de participantes que pueden unirse al grupo.

Permitir búsqueda: para definir si otras personas pueden encontrar el grupo MI al buscarlo.

Derechos de edición: para definir los participantes del grupo MI a quienes desea dar derecho para invitar contactos a unirse al grupo MI y editar las configuraciones de grupo.

Miembros del grupo: consulte "Agregar o eliminar miembros de un grupo", pág. 201.

Lista de prohibidos: ingrese los participantes a los que no se les permite unirse al grupo MI.

Permitir mens. priv.: para permitir la mensajería sólo entre participantes seleccionados.

Id. de grupo: la ID de grupo se crea automáticamente y no puede cambiarse.

Agregar o eliminar miembros de un grupo

Para agregar miembros a un grupo, seleccione **Grupos de MI**, desplácese a un grupo MI y seleccione **Opciones** > **Grupo** > **Configuraciones** > **Miembros del grupo** > **Sólo seleccionado** o **Todos**.

Para eliminar un miembro del grupo MI, desplácese al miembro y seleccione **Opciones** > **Eliminar**. Para eliminar todos los miembros, seleccione **Opciones** > **Eliminar todos**.

Configuraciones de servidor MI

Seleccione **Opciones** > **Configuraciones** > **Configurac. servidor**. Puede recibir las configuraciones en un mensaje de texto especial del operador de red o proveedor de servicios que ofrece el servicio chat. Solicite a su proveedor de servicios la ID de usuario y contraseña cuando se suscriba al servicio. Si no sabe su ID de usuario o contraseña, comuníquese con su proveedor de servicios.

Servers: ver una lista de todos los servidores MI definidos.

Default server: cambie el servidor MI al cual desea conectarse.

Tipo inic. sesión MI: para iniciar una sesión automáticamente al acceder **MI**, seleccione **Al iniciar aplicac.**.

Para agregar un nuevo servidor a su lista de servidores MI, seleccione **Servers** > **Opciones** > **New server**. Ingrese las siguientes configuraciones:

Server name: el nombre del servidor MI

Access point in use: el punto de acceso que desea usar para el servidor

Web address: la dirección URL del servidor MI

User ID: su ID de usuario

Password: su contraseña de acceso

Push to talk (servicio de red)

Nota: Está función no esta disponible en todas los países, consulte la disponibilidad de esta función con su operador local.

Esta función depende del proveedor.

Comunicación instantánea (PTT) es un servicio de voz sobre IP en tiempo real implementado sobre una red GSM/GPRS. Push to talk provee comunicación de voz directa conectada al pulsar una tecla. Use Push to talk para conversar con una persona o un grupo de personas.

Antes de que pueda usar Push to talk, deberá definir el punto de acceso PTT y las configuraciones PTT. Puede recibir las configuraciones en un mensaje de texto especial del proveedor de servicios que ofrece el servicio de Push to talk.

Pulse y seleccione **Personal** > **PTT**.

En Push to talk, una persona habla mientras la otra escucha por el altavoz integrado. Los altavoces se alternan para responder entre sí. Dado a que sólo un miembro del grupo puede hablar a la vez, la duración máxima de palabras es limitada. La duración máxima es configurada generalmente a treinta segundos. Para obtener más información sobre la duración del turno para hablar en su red, comuníquese con su operador de red o proveedor de servicios.

Advertencia: Asegúrese de sujetar el dispositivo en frente de usted y no cerca de su oído. El altavoz se usa para escuchar. La exposición continua a un nivel de volumen alto puede perjudicar su capacidad de audición.

Las llamadas telefónicas siempre toman prioridad sobre las actividades de Push to talk.

Definir un punto de acceso PTT

Varios proveedores de servicios requieren que use un punto de acceso a Internet (IAP) como su punto de acceso predeterminado. Otros proveedores de servicios le permiten usar un punto de acceso WAP. Comuníquese con su proveedor de servicios para obtener más información.

Si no tiene una conexión WAP, es posible que tenga que comunicarse con su proveedor de servicios para asistencia con la primera conexión o visite www.nokia.com/phonesettings para EE.UU. o http://latinoamerica2.ext.nokia.com/configuraciones para Latinoamérica.

Configuraciones PTT

Pulse y seleccione **Personal** > **PTT** > **Opciones** > **Configuraciones PTT**. Ingrese la siguiente información:

Configuraciones de usuario:

Llamadas PTT entrantes: seleccione **Notificar** si desea ver una notificación de las llamadas entrantes. Seleccione **Aceptar auto.** si desea contestar automáticamente las llamadas PTT. Seleccione **No permitidas** si desea rechazar automáticamente las llamadas PTT.

Tono aviso llamada PTT: seleccione **Definido por perfil** si desea que la configuración de aviso para llamadas entrantes PTT siga su configuración de perfil. Por ejemplo, si su perfil es silencio, PTT está configurado a no molestar (DND) y, con la excepción de una solicitud de devolución de llamada, no estará disponible a otras personas usando Push to talk.

Tono solic. devol. llam.: seleccione un tono para las solicitudes de devolución de llamada.

Inicio de aplicación—Seleccione si desea registrarse al servicio PTT cuando enciende su dispositivo.

Sobrenombre predet.: ingrese su sobrenombre predeterminado (veinte caracteres máximo) que se muestra a otros usuarios. Es posible que su proveedor de servicios haya deshabilitado esta opción en su dispositivo. Para obtener más información, comuníquese con su operador de red o proveedor de servicios.

Mostrar mi direc. PTT: seleccione **En llam. individuales**, **En llamadas de grupo**, **En todas las llamadas**, o **Nunca**. Es posible que su proveedor de servicios haya deshabilitado esta opción en su dispositivo. Para obtener más información, comuníquese con su operador de red o proveedor de servicios.

Mostrar mi est. inicio: seleccione **Sí** si desea que aparezca su estado o **No** si desea ocultar su estado.

Conexión:

Dominio: ingrese el nombre de dominio que obtuvo de su proveedor de servicios.

Nombre punto acceso: ingrese el nombre de punto de acceso PTT. El nombre de punto de acceso se necesita para establecer una conexión a la red GSM/GPRS.

Dirección de servidor: ingrese la dirección IP o nombre de dominio del servidor PTT que obtuvo de su proveedor de servicios.

Nombre de usuario: ingrese su nombre de usuario que obtuvo de su proveedor de servicios.

Contraseña de PTT: ingrese una contraseña, si es requerida, para establecer la conexión de datos. La contraseña es provista, por lo general, por el proveedor de servicios y en la mayoría de los casos distingue entre mayúsculas y minúsculas.

Registrarse a PTT

Pulse y seleccione **Personal** > **PTT**. PTT se registra automáticamente al servicio cuando lo inicia.

Cuando se registra con éxito, PTT se conecta automáticamente a los grupos que estaban activos cuando cerró la aplicación por última vez. Si se perdió la conexión, el dispositivo trata automáticamente de registrase hasta que decida salir de push to talk.

Salir de PTT

Seleccione **Opciones** > **Salir**. **¿Desactivar PTT después de salir de la aplicación?**. Seleccione **Sí** para mantener la aplicación activa en segundo plano o **No** para salir y cerrar el servicio.

Si varias aplicaciones están abiertas y desea cambiar de una aplicación por otra, mantenga pulsada .

Realizar una llamada individual

Seleccione **Opciones** > **Contactos de PTT**.

Seleccione el contacto con quien desea hablar de la lista y seleccione **Opciones** > **Conv. individualm.**.

Mantenga pulsada mientras esté hablando. Cuando termine de hablar, suelte .

Deberá terminar la llamada individual antes de crear una llamada nueva. Seleccione **Desconectar** o pulse .

Sugerencia: Recuerde sujetar el dispositivo en frente de usted durante una llamada PTT de modo que pueda ver la pantalla. Hable hacia el micrófono y no cubra el altavoz con las manos.

Contestar una llamada individual

Pulse para iniciar una llamada individual o para rechazar la llamada.

Sugerencia: También puede realizar llamadas individuales y de grupo desde Contactos. Seleccione **Opciones** > **Conv. individualm.** o **Hacer llam. grupo PTT**.

Realizar llamadas de grupo

Para llamar a un grupo, seleccione **Opciones** > **Contactos de PTT**, seleccione **Opciones** > **Hacer llam. grupo PTT**, marque los contactos a los cuales desea llamar y pulse .

Suscribirse al estado de conexión de otros usuarios

Para suscribirse o terminar la suscripción al estado de registro de usuarios PTT, seleccione **Opciones** > **Contactos de PTT**, seleccione el contacto, seleccione **Opciones** > **Mostrar estado inicio** u **Ocultar estado inicio**.

Enviar una solicitud de devolución de llamada

En **Contactos de PTT**, desplácese al nombre deseado y seleccione **Opciones** > **Enviar solic. dev. llam.**.

Responder a una solicitud de devolución de llamada

Cuando alguien le envía una solicitud de devolución de llamada, **1 solicitud de devolución de llamada** aparece en el modo standby. Pulse **Mostrar** para abrir **Buzón dev. llam.**. Seleccione un contacto y seleccione **Opciones** > **Conv. individualm.** para iniciar una llamada individual.

Canales

Puede unirse a canales públicos preconfigurados. Los canales públicos están abiertos a cualquier persona que sepa el URL para el canal.

Cuando está conectado a un canal y habla, todos los miembros registrados en el canal lo escuchan hablar. Hasta cinco canales pueden estar activos a la misma vez. Cuando tenga más de un canal activo, use **Cambiar** para cambiar el canal en el cual desea hablar.

Seleccione **Opciones** > **Canales de PTT**.

Para conectarse a un canal público preconfigurado por primera vez, seleccione **Opciones** > **Nuevo canal** > **Agregar existente**. Cuando se registra en un canal preconfigurado, deberá ingresar un URL para el canal.

Crear sus propios canales

Para crear un canal preconfigurado nuevo, seleccione **Opciones** > **Nuevo canal** > **Crear nuevo**.

Puede crear sus propios canales públicos, seleccionar el nombre para el canal e invitar a participantes. Estos participantes pueden invitar a más personas al canal público.

También puede configurar canales privados. Sólo los usuarios invitados por el anfitrión pueden unirse y usar los canales privados.

Por cada canal, defina lo siguiente: **Nombre de canal**, **Privacidad de canal**, **Sobrenombre en el canal** e **Imagen en miniatura de canal** (opcional).

Cuando haya creado un canal, se le pregunta si desea enviar invitaciones a canales. Las invitaciones a canales son mensajes de texto.

Hablar en un canal

Para hablar en un canal tras registrarse al servicio PTT, pulse . Escucha un tono que indica que tiene acceso al canal.

Continúe manteniendo pulsada mientras habla. Cuando termine de hablar, suelte .

Si trata de responder en un canal al pulsar mientras otro participante habla, **Espere** aparece. Suelte , espere a que la otra persona termine de hablar y pulse otra vez cuando la persona termine de hablar. También puede mantener pulsada y espere a que **Hablar** aparezca.

Cuando está hablando en un canal, la primera persona pulsa cuando la persona actual deja de hablar y entonces es su turno de hablar.

Cuando termine la llamada PTT, seleccione **Desconectar** o pulse .

Para ver los participantes activos actualmente en un canal con una llamada activa en el canal, seleccione **Opciones** > **Miembros activos**.

También puede invitar participantes nuevos al canal cuando está conectado a éste, seleccione **Opciones** > **Enviar invitación** para abrir la vista de invitación. Sólo puede invitar participantes nuevos cuando es el anfitrión de un canal privado o cuando el canal es público. Las invitaciones a canales son mensajes de texto.

Responder a una invitación de canal

Para guardar una invitación de canal, seleccione **Opciones** > **Guardar canal PTT**. El canal es agregado a sus contactos PTT, vista de canales.

Tras guardar la invitación de canal, se le pregunta si desea conectarse al canal. Seleccione **Sí** para abrir la vista de sesiones PTT. Su dispositivo se registra al servicio, si aún no está registrado.

Si rechaza o borra la invitación, la invitación es guardada en el buzón de entrada de mensajería. Para unirse al canal en otro momento, abra el mensaje de invitación y acepte.

Ver el registro PTT

Pulse y seleccione **Personal** > **PTT** > **Opciones** > **Reg llam PTT**. El registro contiene las siguientes carpetas: **PTT perdidas**, **Llam. recibidas** y **PTT creadas**.

Puede iniciar una llamada individual desde **Reg llam PTT** al pulsar .

Registro

Últimas llamadas

Para registrar los números telefónicos de llamadas perdidas, recibidas y números marcados, pulse y seleccione **Personal** > **Registro** > **Últimas llam.** El dispositivo registra llamadas perdidas y recibidas solamente si la red admite estas funciones, el dispositivo está encendido y si está dentro de la cobertura de la red.

Sugerencia: Cuando ve una nota en el modo standby sobre llamadas perdidas, seleccione **Mostrar** para acceder a la lista de llamadas perdidas. Para devolver la llamada, desplácese a un nombre o número y pulse .

Para eliminar las listas de llamadas recientes, seleccione **Opciones** > **Borrar últimas llam.** en la vista principal de últimas llamadas. Para borrar uno de los registros de llamadas, abra el registro que desea borrar y seleccione **Opciones** > **Borrar lista**. Para borrar un evento individual, abra un registro, desplácese hasta el evento y pulse .

Duración de llamadas

Para registrar la duración aproximada de sus llamadas entrantes y salientes, pulse y seleccione **Registro** > **Duración llam.**.

Para configurar la pantalla para mostrar la duración con una llamada activa, seleccione **Personal** > **Registro** > **Opciones** > **Config.** > **Mostrar durac. llam.**. Seleccione **Sí** o **No**.

Nota: El tiempo real facturado por su proveedor de servicios para las llamadas podría variar según los servicios de red, redondeo de la factura y demás.

Para borrar la duración de llamadas, seleccione **Opciones** > **Poner cont. a cero**. Para esto necesita el código de bloqueo, consulte "Seguridad", "Teléfono y SIM", pág. 234.

Paquetes de datos

Para verificar la cantidad de datos enviados y recibidos durante conexiones de paquetes de datos, pulse y seleccione **Personal** > **Registro** > **Paquete datos**. Por ejemplo, se le podría cobrar por las conexiones de paquetes de datos según la cantidad de datos enviados y recibidos.

Monitorear eventos de comunicación

Iconos en **Registro**: evento de comunicación

entrante,

saliente,

perdido

Para monitorear todas las llamadas de voz, mensajes de texto o conexiones de datos registradas por el dispositivo, pulse , seleccione **Personal** > **Registro** y pulse el joystick hacia la derecha para abrir el registro general. Para cada evento de comunicación, podrá ver el nombre del remitente o destinatario, número telefónico, nombre del proveedor de servicios o punto de acceso. Puede filtrar el registro general para ver sólo un tipo de evento y crear tarjetas de contactos nuevas basadas en la información del registro.

Sugerencia: Si desea ver la duración de una llamada de voz en la pantalla principal durante una llamada activa, seleccione **Opciones** > **Config.** > **Mostrar durac. llam.** > **Sí**.

Sugerencia: Para ver una lista de mensajes enviados, pulse y seleccione **Mensajes** > **Enviados**.

Los subeventos, como mensajes de texto enviados en más de una parte y conexiones de paquetes de datos, son registrados como un evento de comunicación. Las conexiones al buzón de correo, centro de mensajería multimedia o páginas Web se muestran como conexiones de paquetes de datos.

Para filtrar el registro, seleccione **Opciones** > **Filtrar** y un filtro.

Para borrar permanentemente el contenido del registro, registro de llamadas recientes e informes de entrega de mensajes, seleccione **Opciones** > **Borrar registro**. Seleccione **Sí** para confirmar.

Para programar **Duración del registro**, seleccione **Opciones** > **Config.** > **Duración del registro**. Los eventos del registro quedan en la memoria del dispositivo por una cantidad fija de días y después son borrados automáticamente para liberar espacio en la memoria. Si selecciona **Ningún registro**, todo el contenido del registro, registro de llamadas recientes e informes de entrega de mensajes son borrados permanentemente.

Para ver los detalles de un evento de comunicación, en la vista de registro general, desplácese a un evento y pulse el joystick.

Sugerencia: En la vista de detalles, puede, por ejemplo, copiar un número telefónico al portapapeles y pegarlo en un mensaje de texto. Seleccione **Opciones** > **Copiar número**.

Packet data counter and connection timer (Contador de datos de paquete y temporizador de conexión): para ver cuanta información, en kilobytes, ha sido transferida y la duración de una conexión de paquetes de datos, desplácese a un evento entrante o saliente indicado por **Paq.** y seleccione **Opciones** > **Ver detalles**.

Agenda

Sugerencia: Haga copias de seguridad periódicamente de la información en el teléfono con Nokia PC Suite. Puede restaurar la información, como los ingresos de la agenda, en el dispositivo en otro momento.

Crear entradas de agenda

Acceso directo: en la vista diaria, semanal o mensual de la agenda, pulse cualquier tecla (de **1** a **0**). Se abre un ingreso de reunión y los caracteres que ingresa son agregados al campo **Asunto**. En la vista de tareas, se abre una entrada de tarea.

1 Pulse y seleccione **Personal** > **Agenda** > **Opciones** > **Nueva entrada** y alguna de las siguientes opciones:

Reunión: para recordarle de una cita con fecha y hora específica.

Memorándum: para escribir una entrada general para un día.

Aniversario: para recordarle de cumpleaños o fechas especiales. Las entradas de aniversario son repetidos anualmente.

Tarea: para recordarle de una tarea que necesita hacer para una fecha

2 Rellene los campos.

Alarma (citas y aniversarios): seleccione **Activar** y desplácese hacia abajo para rellenar los recuadros **Hora de alarma** y **Día de alarma**. en la vista de día indica una alarma.

Repetir: desplácese a la derecha para repetir la entrada (aparece en la vista de día).

Repetir hasta: puede programar una fecha límite para repetir la entrada.

Sincronización:

Privada: después de sincronizar la entrada de la agenda sólo usted puede verlo y no se muestra a otros con acceso en línea a la vista de agenda.

Pública: la entrada de agenda se muestra a otras personas que pueden acceder y ver su agenda en línea.

Ninguna: la entrada de agenda no es copiado en su PC al sincronizar.

3 Para guardar la entrada, seleccione **Hecho**.

Cuando la alarma de la agenda se activa para una nota, seleccione **Silencio** para apagar el tono de alarma de la agenda. El texto del recordatorio queda en la pantalla.

Para finalizar la alarma de la agenda, seleccione **Parar**. Para programar la alarma para que haga una pausa, seleccione **Pausa**.

Para enviar una nota de agenda a un teléfono compatible, seleccione **Opciones** > **Enviar** > **Vía mensaje corto**, **Vía multimedia**, **Vía e-mail** o **Vía Bluetooth**.

Sugerencia: Puede mover información de la agenda y lista de tareas de varios dispositivos Nokia a su dispositivo o sincronizar su agenda y lista de tareas con una PC compatible usando Nokia PC Suite. Consulte el CD-ROM suministrado con el dispositivo.

Programar una alarma de agenda

Puede programar una alarma para recordarle de citas y aniversarios.

1. Abra una entrada para el cual desea programar una alarma y seleccione **Alarma** > **Activar**.
2. Programe **Hora de alarma** y **Día de alarma**.
3. Desplácese hacia abajo hasta **Repetir** y desplácese a la derecha para seleccionar cuándo se repetirá la alarma.
4. Seleccione **Hecho**.

Para eliminar una alarma de agenda, abra la entrada para el cual desea eliminar la alarma y seleccione **Alarma** > **Desactivar**.

Vistas de la agenda

Sugerencia: Seleccione **Opciones** > **Configuraciones** para cambiar el primer día de la semana o la vista que aparece cuando abre la agenda.

Para abrir la vista mensual o la vista de tareas, seleccione **Opciones** > **Vista del mes** / **Vista de tareas**.

En la vista mensual, las fechas con entradas de agenda son marcadas con un triángulo pequeño en la esquina inferior derecha. En la vista semanal, los memorándums y aniversarios quedan ubicados antes de las 8 a.m. Pulse * para cambiar entre la vista mensual, semanal y de día.

Iconos en las vistas de día y semana:

Memorándum

Aniversario

No hay icono para **Reunión**.

Para ir a una fecha, seleccione **Opciones** > **Ir a fecha**, escriba la fecha y seleccione **Aceptar**.

Pulse # para ir al día de hoy.

Vista de tareas

Pulse y seleccione **Opciones** > **Vista de tareas** para escribir notas y mantener una lista de tareas.

Para agregar una nota, pulse cualquier tecla para comenzar a escribir la tarea en el campo **Asunto**.

Para programar la fecha de vencimiento para la tarea, desplácese al campo **Fecha de vencimiento** e ingrese una fecha.

Para configurar la prioridad para la nota **Tarea**, desplácese al campo **Prioridad** y desplácese a la derecha para seleccionar la prioridad. Los iconos de prioridad son ! (**Alta**) y ↓ (**Baja**). No hay iconos para **Normal**.

Para marcar una tarea completada, desplácese a ésta en la lista **Tarea** y seleccione **Opciones** > **Marcar como hecha**.

Para restaurar una tarea, desplácese a ésta en la lista **Tarea** y seleccione **Opciones** > **Marcar no hecha**.

Borrar entradas de la agenda

Borrar entradas vencidas en **Agenda** libera espacio en la memoria de su teléfono.

Para borrar más de un evento a la vez, vaya a la vista mensual y seleccione **Opciones** > **Eliminar entrada** y alguna de las siguientes opciones:

Antes de fecha: borra todos las entradas de la agenda que ocurren antes de la fecha que especifique.

Todas las entradas: borra todos las entradas de la agenda.

Configuraciones de la agenda

Desplácese a las siguientes configuraciones y pulse el joystick para editar:

Tono alarma agenda: para cambiar el tono de alarma, desplácese al tono nuevo y pulse el joystick.

Vista predeterminada: para cambiar la vista que aparece cuando abre la agenda, desplácese a la vista que desea y pulse el joystick.

La semana comienza: para cambiar el primer día de la semana, desplácese al día que desea y pulse el joystick.

Título vista semana: seleccione **N° de la semana** o **Fechas semana**. Si selecciona cualquier otro día, que no sea lunes, en **La semana comienza**, la configuración anterior no estará disponible.

Para guardar sus configuraciones, pulse **Atrás**.

Servicios

Varios proveedores de servicios mantienen páginas específicamente diseñadas para dispositivos móviles. Estas páginas usan el lenguaje de marcado para telefonía inalámbrica (WML), lenguaje extensible de marcado de hipertexto (XHTML) o lenguaje de marcado de hipertexto (HTML).

Para obtener más información sobre la disponibilidad de servicios, precios y tarifas, comuníquese con su operador de red o proveedor de servicios. Los proveedores de servicios también le darán las instrucciones sobre el uso de sus servicios.

Pulse y seleccione **Servicios**.

Acceso directo: Para iniciar una conexión, mantenga pulsada **0** en el modo standby.

Punto de acceso para servicios

Para acceder a páginas, necesita las configuraciones de servicio para la página que desea usar. Puede recibir las configuraciones en un mensaje de texto especial del operador de red o proveedor de servicios que ofrece la página. Consulte "Datos y configuraciones", pág. 186. Para obtener más información, comuníquese con su operador de red o proveedor de servicios.

Sugerencia: Las configuraciones pueden estar disponibles en el sitio Web de un operador de red o proveedor de servicios.

Ingresar manualmente las configuraciones

1. Pulse y seleccione **Herramient.** > **Config.** > **Conexión** > **Puntos de acceso** y defina las configuraciones para un punto de acceso. Siga las instrucciones facilitadas por su proveedor de servicios. Consulte "Conexión", pág. 229.
2. Pulse y seleccione **Servicios** > **Opciones** > **Admin. anotaciones** > **Agregar favorito**. Escriba un nombre para el favorito y la dirección de la página definida para el punto de acceso actual.
3. Para configurar el punto de acceso creado como el punto de acceso predeterminado, seleccione **Opciones** > **Configuraciones** > **Punto de acceso**.

Vista Favoritos

La vista de favoritos se abre cuando accede a **Servicios**.

Glosario: Un favorito consta de una dirección Internet (obligatorio), título del favorito, punto de acceso y si la página Web requiere un nombre de usuario y contraseña.

Su dispositivo puede tener cargados algunos favoritos de algunos sitios que no estén afiliados con Nokia. Nokia no garantiza ni respalda estos sitios. Si accede a ellos, deberá tomar las mismas precauciones en cuanto a seguridad y contenido, como lo haría con cualquier otro sitio de Internet.

Iconos en la vista favoritos:

La página de inicio definida por el punto de acceso predeterminado. Si usa otro punto de acceso predeterminado para la exploración, la página de inicio cambia a este efecto.

La carpeta de favoritos automáticos contiene favoritos ( ) recopilados automáticamente mientras uan las páginas. Los favoritos en esta carpeta son organizados automáticamente según el dominio.

Cualquier favorito con el título o dirección Internet del favorito.

Agregar favoritos manualmente

1 En la vista de favoritos, seleccione **Opciones** > **Admin. anotaciones** > **Agregar favorito**.

2 Comience a rellenar los campos de ingreso. Sólo deberá definir la dirección URL. El punto de acceso predeterminado se asigna al favorito si no hay otro seleccionado. Pulse * para ingresar caracteres especiales como **/**, **.**, **:** y @. Pulse para borrar caracteres.

3 Seleccione **Opciones** > **Guardar** para guardar el favorito.

Enviar favoritos

Desplácese a un favorito y seleccione **Opciones** > **Enviar** > **Vía mensaje corto**. Pulse para enviar. Puede enviar más de un favorito a la misma vez.

Establecer una conexión

Tras guardar todas las configuraciones de conexión requeridas, puede acceder a las páginas.

1 Seleccione un favorito o ingrese la dirección en el recuadro ( ). Cuando ingresa la dirección, los favoritos coincidentes aparecen en el recuadro en la parte superior. Desplácese hacia arriba para seleccionar un favorito coincidente.

2 Pulse el joystick para descargar la página.

Seguridad de conexión

Si el indicador de seguridad aparece durante la conexión, la transmisión de datos entre el dispositivo y el gateway o servidor de Internet queda encriptado.

El icono de seguridad no indica que la transmisión de datos establecida entre el gateway y el servidor de contenido (o el lugar donde se almacena el recurso solicitado) sea segura. El proveedor de servicios asegura la transmisión de datos entre el gateway y el servidor de contenido.

Seleccione **Opciones** > **Detalles** > **Seguridad** para ver detalles sobre la conexión, estado de encriptación e información sobre la autenticación del servidor y usuario.

Se pueden requerir certificados de seguridad para ciertos servicios, como para los servicios bancarios. Para conexiones de este tipo necesita certificados de seguridad. Para obtener más información, comuníquese con su proveedor de servicios. También consulte "Administración de certificados", pág. 235.

Explorar

Importante: Use sólo servicios fiables que brinden seguridad y protección adecuada contra software perjudicial.

Para descargar una página, seleccione un favorito o ingrese la dirección en el campo ( ).

En una página Web, los enlaces nuevos aparecen subrayados en azul y los enlaces visitados previamente, en violeta. Las imágenes que sirven de enlaces tienen alrededor un marco azul.

Para abrir un enlace, para marcar campos y seleccionar elementos, pulse el joystick.

Acceso directo: Use # para ir al final de una página y * para ir al comienzo.

Para retroceder a la página anterior durante la exploración, seleccione **Atrás**. Si **Atrás** no está disponible, seleccione **Opciones** > **Opc. navegación** > **Historial** para ver una lista cronológica de las páginas que ha visitado durante una sesión de exploración. La lista de historial se borra cada vez que cierra una sesión.

Para obtener el contenido más reciente del servidor, seleccione **Opciones** > **Opc. navegación** > **Cargar de nuevo**.

Para guardar un favorito, seleccione **Opciones** > **Guardar como favorito**.

Sugerencia: Para acceder a la vista de favoritos mientras navega, mantenga pulsado el joystick. Para regresar a la vista del explorador, seleccione **Opciones** > **Volver a página**.

Para guardar una página mientras uan, seleccione **Opciones** > **Opc. avanzadas** > **Guardar página**. Puede guardar páginas en la memoria del dispositivo o en el disco duro y explorar estas páginas cuando está desconectado. Para acceder a las páginas en otro momento, desplácese a la derecha en la vista de favoritos para abrir la vista **Páginas guardadas**.

Para ingresar una nueva dirección URL, seleccione **Opciones** > **Ir a dirección Web**.

Para abrir una sublista de comandos o acciones específicas para la página actualmente abierta, seleccione **Opciones** > **Opciones servicio**, si está disponible.

Puede descargar archivos que no se pueden mostrar en la página, como tonos de timbre, imágenes, logotipos del operador, temas y videoclips. Para descargar un elemento, desplácese al enlace y pulse el joystick.

Al empezar una descarga, aparecerá una lista de descargas en proceso, interrumpidas y completas de la sesión de navegación actual. También puede ver la lista seleccionando **Opciones** > **Herramientas** > **Descargas**. En la lista, desplácese a un elemento y seleccione **Opciones** para hacer una pausa, reanudar o cancelar descargas actuales o abrir, guardar o borrar descargas realizadas.

Los elementos descargados son manejados por las aplicaciones correspondientes. Por ejemplo, una imagen descargada se guarda en **Galería**.

Importante: Sólo instale y utilice aplicaciones y otros software de fuentes que brinden seguridad y protección adecuada contra software perjudicial.

Sugerencia: Su explorador recopila los favoritos automáticamente mientras uan las páginas. Los favoritos son guardados en la carpeta automática de favoritos ( ) y son organizados automáticamente según su dominio. También consulte 'Configuraciones para servicios', pág. 215.

Ver las páginas guardadas

Si uan regularmente páginas con información que no cambia con frecuencia, puede guardarlas y explorarlas cuando está desconectado. En la vista de páginas guardadas también puede crear carpetas para guardar las páginas que guarde.

Para acceder a la vista de páginas guardadas, desplácese a la derecha en la vista de favoritos. En la vista de páginas guardadas, pulse el joystick para abrir una página guardada ( ).

Para guardar una página mientras uan, seleccione **Opciones** > **Opc. avanzadas** > **Guardar página**.

Para iniciar la conexión al servicio del explorador y para descargar la versión más reciente de la página, seleccione **Opciones** > **Cargar de nuevo**. El dispositivo se mantiene conectado tras recargar la página.

Descargar y comprar elementos

Puede descargar elementos como tonos de timbre, imágenes, logotipos del operador, temas y videoclips. Estos elementos pueden ser gratuitos o comprados. Los elementos descargados son manejados por sus aplicaciones respectivas, por ejemplo, una foto se guarda en **Galería**.

Importante: Sólo instale y utilice aplicaciones y otros software de fuentes que brinden seguridad y protección adecuada contra software perjudicial.

1 Para descargar el elemento, desplácese al enlace y pulse el joystick.

2 Seleccione la opción correspondiente para comprar el elemento.

3 Lea detenidamente toda la información provista.

Para continuar con la descarga, seleccione **Aceptar**. Para cancelar la descarga, seleccione **Cancelar**.

Para obtener más información sobre la descarga de música, consulte "Ir a tienda de música", pág. 147.

Las protecciones de copyright pueden impedir la copia, modificación, transferencia o desvío de imágenes, música (incluso tonos de timbre) y otro contenido.

Finalizar una conexión

Seleccione **Opciones** > **Opc. avanzadas** > **Desconectar** para finalizar la conexión y ver la página del explorador en el modo desconectado o para finalizar la conexión y cerrar el explorador, seleccione **Opciones** > **Salir**.

Vaciar la memoria caché

La información o servicios a los cuales ha accedido son guardados en la memoria caché del dispositivo.

La memoria caché es una parte de la memoria que se emplea para guardar datos en forma temporal. Si ha intentado acceder o ha accedido a información confidencial que exige contraseñas, vacíe la memoria caché después de cada uso. La información o los servicios a los que ha accedido se almacenan en la memoria caché. Para vaciar la memoria caché, seleccione **Opciones** > **Opc. avanzadas** > **Borrar caché**.

El dispositivo puede tener cargados algunos favoritos de algunos sitios que no estén asociados con Nokia. Nokia no garantiza ni respalda estos sitios. Si accede a ellos, deberá tomar las mismas precauciones en cuanto a seguridad y contenido, como haría con cualquier otro sitio de Internet.

Configuraciones para servicios

Seleccione **Opciones** > **Configuraciones** y alguna de las siguientes configuraciones:

Punto de acceso: para cambiar el punto de acceso predeterminado, pulse el joystick para abrir una lista de los puntos de acceso disponibles. Consulte "Conexión", pág. 229. Algunos o todos los puntos de acceso pueden estar preconfigurados en su dispositivo por su proveedor de servicios u operador de red y es posible que no pueda cambiarlos, crearlos, editarlos ni retirarlos.

Mostrar imág./obj.: seleccione si desea cargar imágenes mientras uan. Si selecciona **No**, para cargar imágenes en otro momento durante la exploración, seleccione **Opciones** > **Mostrar imágenes**.

Tamaño de letra: seleccione el tamaño del texto.

Codificación predet.: si los caracteres del texto no aparecen correctamente, puede seleccionar otra codificación según el idioma.

Anotaciones auto.: para desactivar la recopilación automática de favoritos, seleccione **Desactivar**. Si desea continuar recopilando favoritos automáticos, pero desea ocultar la carpeta de la vista de favoritos, seleccione **Ocultar carpeta**.

Tamaño de pantalla: seleccione lo que desea que aparezca cuando está explorando. Seleccione **Sólo teclas selec.** o **Pantalla completa**.

Página de inicio: defina la página de inicio.

Página de búsqueda: para definir una página Web que es descargada cuando selecciona **Opc. navegación** > **Abrir pág. búsqueda** en la vista de favoritos o durante la exploración.

Volumen: si desea que el explorador reproduzca sonidos grabados en las páginas Web, seleccione un nivel de volumen.

Entrega: si desea ver el formato más exacto posible de la página cuando está en el modo **Pantalla pequeña**, seleccione **Por calidad**. Si no desea que se descarguen páginas externas de estilo cascada, seleccione **Por velocidad**.

Cookies: active o desactive la recepción y envío de cookies.

Glosario: Los proveedores de servicios usan cookies para identificar a los usuarios y sus preferencias de contenido usado con frecuencia.

Script Java/ECMA: active o desactive el uso de scripts.

Adv. de seguridad: oculte o muestre las notificaciones de seguridad.

Conf. envío de DTMF: seleccione si desea confirmar antes de que el dispositivo envíe tonos DTMF durante una llamada de voz. También consulte "Opciones durante una llamada de voz", pág. 162.

Conectividad

Conectividad Bluetooth

Se puede conectar inalámbricamente a otros dispositivos compatibles usando la tecnología inalámbrica Bluetooth. Los dispositivos compatibles pueden incluir teléfonos móviles, computadoras y accesorios como auriculares y equipos para auto. Puede usar la conectividad Bluetooth para enviar imágenes, videoclips, música y clips de sonido y notas, conectarse inalámbricamente con su PC compatible (por ejemplo, para transferir archivos).

Dado a que los dispositivos con la tecnología Bluetooth se comunican usando ondas de radio, su dispositivo y el otro dispositivo no necesitan estar en línea directa. Los dos dispositivos sólo necesitan estar dentro de una distancia máxima de 10 metros (33 pies), aunque la conexión si puede recibir interferencias de obstrucciones, como paredes o de otros dispositivos electrónicos.

Este dispositivo cumple con la Especificación 1.2 Bluetooth que admite los siguientes perfiles: Perfil Básico de Impresión, Perfil de Acceso Genérico, Perfil de Puerto Serial, Perfil de Acceso Telefónico a Redes, Perfil de Auricular, Perfil Manos Libres, Perfil Genérico de Objetos de Intercambio, Perfil de Objecto Push, Perfil de Transferencia de Archivos, Perfil de Imagen Básica y Perfil de Acceso SIM. Para garantizar la compatibilidad con otros dispositivos que admiten la tecnología Bluetooth, utilice sólo accesorios aprobados por Nokia para este modelo. Compruebe con sus respectivos fabricantes la compatibilidad con este dispositivo.

Glosario: Un perfil corresponde a un servicio o una función y define cómo se conectan los diferentes dispositivos. Por ejemplo, el Perfil Manos Libres se usa entre el dispositivo manos libres y el teléfono. Para que los dispositivos sean compatibles, deben admitir los mismos perfiles.

Puede que existan ciertas restricciones al utilizar la tecnología Bluetooth en algunas localidades. Consulte a las autoridades locales o al proveedor de servicios.

Las funciones que utilizan la tecnología Bluetooth o permitir la activación de estas funciones en segundo plano mientras se utilizan otras funciones, aumentan la demanda de alimentación de la batería y reducen la vida útil de la batería.

Cuando el dispositivo está bloqueado, no se puede usar la conectividad Bluetooth. Consulte "Seguridad", pág. 234 para obtener más información sobre el bloqueo del dispositivo.

Configuraciones

Pulse y seleccione **Conect.** > **Bluetooth**. Cuando abre la aplicación por primera vez, se le solicita que defina un nombre para su dispositivo. Después de activar la conectividad Bluetooth y cambiar **Visibilidad mi tel.** a **Mostrado a todos**, su dispositivo y este nombre pueden ser vistos por otros usuarios que usan la tecnología Bluetooth.

Seleccione alguna de estas opciones:

Bluetooth: seleccione **Activado** o **Desactivado**. Para conectarse inalámbricamente a otro dispositivo compatible, configure primero la conectividad Bluetooth en **Activado** y luego establezca la conexión.

Visibilidad mi tel.: para que su dispositivo sea visto por otros dispositivos que usan la tecnología Bluetooth, seleccione **Mostrado a todos**. Para ocultarlo de otros dispositivos, seleccione **Oculto**.

Mi nombre teléfono: edite el nombre de su dispositivo.

Modo remoto de SIM: para habilitar otro dispositivo, como un accesorio de equipo para auto, para usar la tarjeta SIM en su dispositivo para conectarse a la red, seleccione **Activado**.

Modo de SIM remota

Para usar el modo de SIM remota, active la conectividad Bluetooth y habilite el uso del modo de SIM remota en su dispositivo. Antes de poder activar el modo, los dos dispositivos deberán estar vinculados y la vinculación deberá ser iniciada desde el otro dispositivo. Al vincular, use un código de dieciséis dígitos y configure el otro dispositivo como autorizado. Consulte "Vincular dispositivos", pág. 220. El modo de SIM remota se activa desde el otro dispositivo.

Cuando el modo de SIM remota está activado en su Nokia N91, aparece **SIM remota** en el modo standby. La conexión a la red inalámbrica es desactivada, como se indica por ✕ en el área del indicador de la intensidad de la señal y no podrá usar los servicios de tarjeta SIM o funciones que requieren la cobertura de la red. Sin embargo, una conexión de LAN inalámbrica permanece activa durante el modo de SIM remota.

Advertencia: En el perfil de SIM remota, no se pueden realizar ni recibir llamadas con su dispositivo, excepto las llamadas de emergencia, ni usar otras funciones que requieran la cobertura de red. Para realizar llamadas, primero deberá salir del modo de SIM remota. Si el dispositivo está bloqueado, ingrese el código de bloqueo.

Para salir del modo de SIM remota, pulse la tecla encender/apagar y seleccione **Salir modo rem. SIM**.

Sugerencias de seguridad

Cuando no esté usando la conectividad Bluetooth, seleccione **Bluetooth** > **Desactivado** o **Visibilidad mi tel.** > **Oculto**.

El uso del teléfono en el modo oculto es una manera más segura de evitar software de código malicioso.

No haga vinculación con un dispositivo desconocido. No acepte conectividad Bluetooth de fuentes que no son de confianza.

Enviar datos usando la conectividad Bluetooth

Puede tener varias conexiones Bluetooth activas a la vez. Por ejemplo, si está conectado a un auricular, también puede, al mismo tiempo, transferir archivos a otro dispositivo compatible.

Indicadores de conectividad Bluetooth

- Cuando aparece en el modo en standby, la conectividad Bluetooth está activa.
- Cuando está destellando, su dispositivo está tratando de conectarse con otro dispositivo.
- Cuando aparece continuamente, se están transmitiendo datos usando la conectividad Bluetooth.

Sugerencia: Para enviar texto usando la conectividad Bluetooth, abra **Notas**, escriba el texto y seleccione **Opciones** > **Enviar** > **Vía Bluetooth**.

1 Abra la aplicación o carpeta en la cual tiene el elemento que desea enviar. Por ejemplo, para enviar una imagen a otro dispositivo compatible, abra **Galería**.

2 Seleccione el elemento y **Opciones** > **Enviar** > **Vía Bluetooth**. El icono de dispositivo, el nombre del dispositivo, el tipo de dispositivo o un nombre corto para los dispositivos con tecnología Bluetooth dentro del área de alcance comienzan a aparecer en la pantalla.

Iconos de dispositivos: computadora, teléfono, dispositivo de audio o de video y otro dispositivo.

Para interrumpir la búsqueda, seleccione **Parar**.

3 Seleccione el dispositivo con el cual desea establecer la conexión.

4 Si el otro dispositivo requiere la vinculación antes de poder transmitir datos, sonará un tono y se le pedirá que ingrese un código. Consulte "Vincular dispositivos", pág. 220.

5 Cuando se haya establecido la conexión, **Enviando datos** aparece en la pantalla.

La carpeta **Enviados** en **Mensajes** no guarda los mensajes enviados usando la conectividad Bluetooth.

Sugerencia: Durante la búsqueda de dispositivos, algunos dispositivos pueden mostrar sólo las direcciones únicas (direcciones de dispositivos). Para encontrar la dirección única de su dispositivo, ingrese el código ***#2820#** en el modo standby.

Vincular dispositivos

Para abrir la vista de dispositivos vinculados ( ), en la vista principal de la aplicación **Bluetooth**, desplácese a la derecha.

Antes de vincular, cree su propio código (de uno a dieciséis dígitos) y póngase de acuerdo con el dueño del otro dispositivo para que use el mismo código. Los dispositivos sin interfaz de usuario tienen un código de fábrica predeterminado. El código se usa una sola vez.

Para vincular con un dispositivo, seleccione **Opciones** > **Vincular nuevo disp.** Los dispositivos con tecnología inalámbrica Bluetooth dentro del área de alcance empiezan a aparecer en la pantalla. Seleccione el dispositivo e ingrese el código. También se deberá ingresar el mismo código en el otro dispositivo. Después de la vinculación, el dispositivo queda guardado en la vista de dispositivos vinculados.

Los dispositivos vinculados son indicados por en la búsqueda del dispositivo.

Para configurar un dispositivo como autorizado o no autorizado, desplácese a un dispositivo y seleccione de las siguientes opciones:

Fijar como autoriz.: las conexiones entre su dispositivo y este dispositivo pueden ser realizadas sin que usted lo sepa. No se necesita aceptación ni autorización por separado. Use este estado para sus propios dispositivos, como su auricular o PC compatible o dispositivos pertenecientes a otra persona de confianza. indica los dispositivos autorizados en la vista de dispositivos vinculados.

Fijar no autorizado: los pedidos de conexión de este dispositivo deben ser aceptados cada vez por separado.

Para cancelar una vinculación, desplácese al dispositivo y seleccione **Opciones** > **Eliminar**. Si desea cancelar todos los vínculos, seleccione **Opciones** > **Eliminar todos**.

Sugerencia: Si está conectado actualmente con un dispositivo y borra el vínculo con ese dispositivo, la vinculación es cancelada de inmediato y la conexión se desactiva.

Recibir datos usando la conectividad Bluetooth

Cuando reciba datos mediante la conectividad Bluetooth, sonará un tono y se le pregunta si desea aceptar el mensaje. Si lo acepta, aparece en la pantalla y el elemento queda en la carpeta **Buzón entrada** en **Mensajes**. Los mensajes recibidos a través de la conectividad Bluetooth son indicados por . Consulte "Buzón de entrada: recibir mensajes", pág. 186.

Desactivar la conectividad Bluetooth

Para desactivar la conectividad Bluetooth, seleccione **Bluetooth** > **Desactivado**.

Cable de datos

Puede conectarse con una PC compatible con un cable de datos USB. Con la conectividad del cable de datos, puede usar Nokia PC Suite y transferir música u otros datos, como archivos de imagen, entre su dispositivo y la PC. Consulte "Transferir música", pág. 148 para obtener más información sobre transferencia de música.

Pulse y seleccione **Conect.** > **Cable dat.** > **Modo de cable de datos**. Seleccione **Media Player** para transferir música, **PC Suite** para usar Nokia PC Suite o **USB** para transferir otros datos. Para que el dispositivo pregunte el propósito de la conexión cada vez que conecta el cable, seleccione **Preguntar al conectarse**.

Conexiones PC

Puede usar su dispositivo con una variedad de aplicaciones de conectividad a PC y de aplicaciones para la comunicación de datos. Con Nokia PC Suite puede, por ejemplo, sincronizar los contactos, agenda y lista de tareas y transferir imágenes entre su dispositivo y una PC compatible. Dada la alta capacidad del disco duro de 4 GB puede guardar cientos de imágenes y canciones en el dispositivo.

Haga siempre la conexión desde la PC para sincronizar con su dispositivo.

Para obtener información adicional sobre cómo instalar Nokia PC Suite (compatible con Windows 2000 y Windows XP), consulte el manual del usuario para Nokia PC Suite y la función de ayuda de Nokia PC Suite en la sección "Instalación" en el CD-ROM suministrado en el paquete de ventas.

CD-ROM

El CD-ROM debe activarse tras insertarlo en la unidad CD-ROM de un PC compatible. Si no se activa, haga lo siguiente: Abra Windows Explorer, haga clic con el botón derecho del mouse en la unidad CD-ROM en la cual insertó el CD-ROM y seleccione Autoplay.

Su dispositivo como módem

Puede usar su dispositivo como un módem para enviar y recibir e-mail o para conectarse a la Internet con una PC compatible que usa una conexión Bluetooth o cable de datos. Puede encontrar instrucciones específicas de instalación en el Manual del Usuario de Nokia PC Suite en Modem options en el CD-ROM.

Sugerencia: Cuando use Nokia PC Suite por primera vez, para conectar su dispositivo a una PC compatible y usar Nokia PC Suite, use el asistente Get Connected disponible en Nokia PC Suite. Consulte el CD-ROM suministrado en el paquete de ventas.

Administrador de conexiones

Nota: Está función no esta disponible en todas los países, consulte la disponibilidad de esta función con su operador local.

Puede tener conexiones de datos múltiples a la misma vez cuando usa su dispositivo en las redes GSM y UMTS. Pulse y seleccione **Conect.** > **Adm. con.** Para ver el estado de las conexiones de datos o finalizar las conexiones en las redes GSM y UMTS y LAN inalámbrica, seleccione **Conex. activas**. Para buscar LAN inalámbricas disponibles dentro del área de alcance, seleccione **WLAN dispon.**.

Conexiones de datos

En la vista de conexiones activas, puede ver las conexiones de datos abiertas: llamadas de datos ( ), conexiones de datos de paquetes (o) y conexiones de LAN inalámbrica ( ).

Nota: El tiempo real facturado por su proveedor de servicios para las llamadas podría variar según los servicios de red, redondeo de la factura, impuestos y demás.

Para finalizar la conexión, seleccione **Opciones** > **Desconectar**. Para cerrar todas las conexiones, seleccione **Opciones** > **Desconectar todas**.

Para ver los detalles de una conexión, seleccione **Opciones** > **Detalles**. Los detalles que aparecen dependen del tipo de conexión.

Ver los detalles de las conexiones de datos

Para ver los detalles de una conexión, desplácese a una conexión y seleccione **Opciones** > **Detalles**.

Nombre: el nombre de un punto de acceso a Internet (IAP) en uso o el nombre de la conexión de módem si la conexión es por marcación.

Portador: el tipo de conexión de datos: **Llam. datos**, **Alta vel. (sólo GSM)** o **Paq. datos**.

Estado: el estado actual de la conexión: **Conectando**, **Conex. (inactiva)**, **Conex. (activa)**, **En espera**, **Desconect.** o **Desconect.**

Recibid.: la cantidad de datos, en bytes, recibidos en su dispositivo.

Enviados: la cantidad de datos, en bytes, enviados con su dispositivo.

Duración: la duración de la conexión activa.

Velocid.: la velocidad actual del envío y recepción de datos en kilobytes por segundo.

Marcac.: el nombre por marcación usado.

Pto. acc.: nombre del punto de acceso usado.

Compart. (no aparece si la conexión no es compartida): la cantidad de aplicaciones usando la misma conexión.

Red local de área inalámbrica (WLAN)

LAN inalámbrica

La vista de LAN inalámbrica disponible muestra una lista de LANs inalámbricas dentro del área de alcance, el modo de su red (**Infraestructura** o **Ad-hoc**) e indicador de intensidad de la señal. aparece para las redes con encriptación y si su dispositivo tiene una conexión activa en la red.

Para ver los detalles de una red, seleccione **Opciones** > **Detalles**.

Para crear un punto de acceso a Internet en una red, seleccione **Opciones** > **Definir pto. acceso**. Para obtener más información sobre WLAN, consulte "LAN inalámbrica" en pág.138.

Sincronización

Pulse y seleccione **Conect.** > **Sinc.**. **Sinc.** le permite sincronizar sus notas, agenda y contactos con varias aplicaciones de agenda y directorio en una PC compatible o en Internet.

La aplicación de sincronización usa la tecnología SyncML para la sincronización. Para obtener más información sobre la compatibilidad de SyncML, comuníquese con el proveedor de la aplicación de agenda o directorio que desea sincronizar con los datos de su teléfono.

Puede recibir configuraciones de sincronización en un mensaje de texto especial. Consulte "Datos y configuraciones", pág. 186.

Crear un perfil de sincronización nuevo

1 Si no hay perfiles definidos, su dispositivo le pregunta si desea crear un perfil nuevo. Seleccione **Sí**.

Para crear un perfil nuevo, además de los que ya existan, seleccione **Opciones** > **Nuevo perfil sinc.** Seleccione si desea usar los valores de configuración predeterminados o copiar los valores de un perfil existente para usarlos para crear el perfil nuevo.

2 Defina lo siguiente:

Nombre perfil sinc.: escriba una descripción para el perfil.

Aplicaciones: seleccione las aplicaciones que se sincronizarán con este perfil, **Contactos** y **Agenda**, por ejemplo. Desplácese a cada aplicación y pulse el joystick para definir las configuraciones de sincronización:

- **Incluir en sinc.**: seleccione **Sí** si desea sincronizar la aplicación.
- En **Base de datos remota**, ingrese una ruta correcta a la agenda remota, directorio o base de datos de notas en el servidor.

- Seleccione **Tipo de sincronización**: **Normal** (sincronización en dos sentidos), **Sólo a servidor** o **Sólo a teléfono**.

Config. conexión: defina lo siguiente. Comuníquese con su proveedor de servicios o administrador de sistemas para obtener los valores correctos.

- **Versión de servidor**: seleccione la versión SyncML que el servidor de sincronización usa. Deberá seleccionar **1.2** para sincronizar e-mail.
- **ID de servidor**: ingrese la ID de servidor para el servidor de e-mail (sólo aparece si **Versión de servidor** está configurado en **1.2**).
- **Portador de datos**: seleccione el tipo de conexión: **Internet** o **Bluetooth**.
- **Punto de acceso** (sólo aparece si **Portador de datos** está programado en **Internet**): seleccione un punto de acceso para usarlo con la conexión de datos.
- **Dirección de host**: la dirección IP del servidor de host.
- **Puerto**: el número de puerto del servidor (sólo aparece si **Portador de datos** está configurado en **Internet**).
- **Nombre de usuario**: su ID de usuario para el servidor de sincronización.
- **Contraseña**: su contraseña para el servidor de sincronización.
- **Permitir solicit. sinc.**: seleccione **Sí** si desea que el servidor inicie la sincronización.
- **Aceptar solicit. sinc.**: seleccione **No** si desea que su dispositivo le pregunte antes de que el servidor inicie la sincronización.
- **Autenticación de red** (sólo aparece si **Portador de datos** está configurado en **Internet**): seleccione **Sí** para ingresar el nombre de usuario y contraseña de red. Desplácese hacia abajo para ver los campos de nombre de usuario y contraseña.

3 Seleccione **Atrás** para guardar las configuraciones y regresar a la vista principal.

Sincronizar datos

En la vista principal de **Sinc.**, puede ver los diferentes perfiles de sincronización y el tipo de datos a sincronizar.

1 Seleccione un perfil de sincronización y **Opciones** > **Sincronizar**. El estado de la sincronización aparece en la parte inferior de la pantalla.

Para cancelar la sincronización antes de que finalice, seleccione **Cancelar**.

2 Se le avisará cuando termine la sincronización. Tras completar la sincronización, seleccione **Opciones** > **Ver registro** para abrir un archivo de registro con el estado de sincronización (**Completo** o **Incompleto**) y cuántas entradas de agenda o directorio o notas han sido agregados, actualizados, borrados o ignorados (no sincronizados) en su dispositivo o en el servidor.

Administrador de dispositivos

Pulse y seleccione **Conect.** > **Adm. disp.** Puede recibir perfiles de servidor y varios ajustes de configuración de su operador de red, proveedor de servicios o departamento de administración de información de la compañía. Estos ajustes de configuración pueden incluir configuraciones de punto de acceso para conexiones de datos y otras configuraciones usadas por las diferentes aplicaciones en su dispositivo.

Para conectarse a un servidor y recibir ajustes de configuración para su dispositivo, seleccione **Iniciar config.**

Para permitir o rechazar ajustes de sincronización de su proveedor de servicios, seleccione **Activar config.** o **Desactivar config.**.

Configuraciones del perfil de servidor

Comuníquese con su proveedor de servicios para obtener las configuraciones correctas.

Nombre de servidor: ingrese un nombre para el servidor de configuración.

ID de servidor: ingrese el ID único para identificar el servidor de configuración.

Contraseña servidor: ingrese una contraseña para identificar su dispositivo al servidor.

Modo de sesión: seleccione **Bluetooth** o **Internet**.

Punto de acceso: seleccione un punto de acceso para usar cuando se conecte al servidor.

Dirección de servidor: ingrese la dirección URL del servidor.

Puerto: ingrese el número de puerto del servidor.

Nombre de usuario y **Contraseña**: ingrese su nombre de usuario y contraseña.

Permitir config.: para recibir ajustes de configuración del servidor, seleccione **Sí**.

Aceptar auto. solicit.: para que el dispositivo solicite una confirmación antes de aceptar una configuración del servidor, seleccione **No**.

Autenticación de red: seleccione esta opción si desea usar autenticación de red.

Herramientas

Calculadora

Para sumar, restar, multiplicar, dividir y calcular la raíz cuadrada y porcentajes, pulse y seleccione **Herramient.** > **Calculad.**

Nota: Esta calculadora tiene precisión limitada y está diseñada para cálculos sencillos.

Para guardar un número en la memoria (indicado por **M**), seleccione **Opciones** > **Memoria** > **Guardar**. Para recuperar un número en la memoria, seleccione **Opciones** > **Memoria** > **Recuperar**. Para borrar un número de la memoria, seleccione **Opciones** > **Memoria** > **Borrar**.

Calcular porcentajes

1. Ingrese un número para el cual desea calcular el porcentaje.
2. Seleccione | × |, | ÷ |, | − |, o | + |.
3. Ingrese el porcentaje.
4. Seleccione | % |.

Convertidor

Para convertir medidas **Longitud** de una unidad (**Yardas**) a otra (**Metros**), pulse y seleccione **Herramient.** > **Convertid.**

Tenga en cuenta que **Convertid.** tiene precisión limitada y puede resultar en errores de redondeo de cifras.

1. Desplácese al campo **Tipo** y pulse el joystick para abrir una lista de medidas. Desplácese hasta la medida que desea usar y seleccione **Aceptar**.
2. Desplácese al primer campo **Unidad** y pulse el joystick. Seleccione la unidad de la cual desea hacer la conversión y **Aceptar**. Desplácese al próximo campo **Unidad**, seleccione la unidad a la cual desea hacer la conversión.
3. Desplácese al primer campo **Cantidad** e ingrese el valor que desea convertir. El otro campo **Cantidad** cambia automáticamente para mostrar el valor convertido.

   Pulse **#** para agregar un punto decimal y ***** para los símbolos **+**, **–** (para temperatura) y **E** (exponente).

   Sugerencia: Para cambiar el orden de conversión, ingrese el valor en el segundo campo **Cantidad**. El resultado aparece en el primer campo **Cantidad**.

Configurar una moneda base y tasas de cambio

Antes de poder hacer conversión de divisas, deberá seleccionar una moneda base y agregar las tasas de cambio. La tasa de cambio de la moneda base siempre es 1. La moneda base determina las tasas de cambio para otras monedas.

1 Seleccione **Convertid.** > **Opciones** > **Cambios monedas**. Se abre una lista de monedas y puede ver la moneda base actual en la parte superior.

Sugerencia: Para renombrar una moneda, vaya a la vista de tasas de cambio, desplácese a la moneda y seleccione **Opciones** > **Renombrar moneda**.

2 Para cambiar la moneda base, desplácese a la moneda y seleccione **Opciones** > **Fijar moneda base**.

3 Agregue las tasas de cambio. Desplácese a la moneda e ingrese una tasa de cambio nueva, es decir, la cantidad de unidades de la moneda que equivale a una unidad de la moneda base que ha seleccionado.

Tras insertar todas las tasas de cambio necesarias, puede hacer conversiones de divisas.

Nota: cuando cambie la divisa base, deberá ingresar los nuevos cambios puesto que todos los tipos de cambios ingresados anteriormente quedan programados a cero.

Configuraciones

Para cambiar las configuraciones, pulse y seleccione **Herramient.** > **Config.** Desplácese a un grupo de configuraciones y pulse el joystick para abrirlo. Desplácese a una configuración que desea cambiar y pulse el joystick.

Configuraciones del dispositivo

General

Idioma del teléfono: cambiar el idioma de los textos en pantalla de su dispositivo también afecta el formato usado para fechas y hora y los separadores, por ejemplo, en los cálculos. **Automático** selecciona el idioma según la información en su tarjeta SIM. Tras cambiar el idioma de los textos en pantalla, el dispositivo se reinicia.

Cambiar las configuraciones para **Idioma del teléfono** o **Idioma de escritura** afecta cada aplicación en su dispositivo y el cambio permanece activo hasta que modifique otra vez estas configuraciones.

Idioma de escritura: cambiar el idioma afecta los caracteres y caracteres especiales disponibles cuando escribe texto y cuando usa el ingreso de texto predictivo.

Texto predictivo: puede configurar el ingreso de texto predictivo como **Activado** o **Desactivado** para todos los editores en el dispositivo. El diccionario para el ingreso de texto predictivo no está disponible para todos los idiomas.

Saludo inicial o logo: el saludo inicial o logotipo aparece brevemente cada vez que enciende el dispositivo. Seleccione **Predeterminado** para usar la imagen predeterminada, **Texto** para escribir un saludo inicial (hasta cincuenta letras) o **Imagen** para seleccionar una foto o imagen de **Galería**.

Config. originales: puede restaurar ciertas configuraciones a sus valores originales. Para hacer esto, necesitará el código de bloqueo. Consulte "Seguridad", "Teléfono y SIM", pág. 235. Después de reprogramar las configuraciones, el dispositivo puede tardar un poco más en activarse. Los documentos y archivos no se ven afectados.

Modo en espera

Standby activo: use los accesos directos a aplicaciones en el modo standby en la pantalla. Consulte "Standby activo", pág. 158.

Tecla selecc. izq.: para asignar un acceso directo a la tecla de selección izquierda (—) en el modo standby, seleccione una aplicación de la lista.

Tecla selecc. derecha: para asignar un acceso directo a la tecla de selección derecha (—) en el modo standby, seleccione una aplicación de la lista.

Aplic. en standby act.: para seleccionar los accesos directos a aplicaciones que desea que aparezcan en el modo standby activo, seleccione las aplicaciones de la lista. Esta configuración sólo está disponible si **Standby activo** está activo.

También puede asignar accesos directos a distintos movimientos del joystick. Seleccione de la lista la aplicación para cada movimiento. Los accesos directos del joystick sólo están disponibles si **Standby activo** está desactivado.

Logo del operador: esta configuración sólo aparece si ha recibido y guardado un logotipo del operador. Puede seleccionar si desea que aparezca o no el logotipo del operador.

Pantalla

Brillo: puede cambiar el brillo de la pantalla a más claro o más oscuro.

Tiempo ahorro energía: el protector de pantalla se activa cuando se vence el tiempo de espera.

Tiempo espera de luz: seleccione un tiempo de espera tras el cual la luz de fondo de la pantalla se apaga.

Configuraciones de llamadas

Envío del n° propio (servicio de red): para que su número telefónico aparezca (**Activado**) o no (**Desactivado**) en el teléfono de la persona a quien está llamando o puede ser programado por su operador de red o proveedor de servicios al suscribirse (**Fijado por la red**).

Llamada en espera (servicio de red): si ha activado llamada en espera, la red le notifica de una llamada entrante nueva mientras tiene una llamada en curso. Seleccione **Activar**

para solicitar que la red active la llamada en espera, **Cancelar** para solicitar que la red desactive la llamada en espera, o **Verificar estado** para verificar si la función está activada o desactivada.

Rechaz. llam. c/ SMS: seleccione **Sí** para enviar un mensaje de texto a un llamante informándole por qué no pudo atender la llamada. Consulte "Contestar o rechazar una llamada", pág. 161.

Texto de mensaje: escriba un texto para enviarlo en un mensaje de texto cuando rechaza una llamada.

Marcación automática: seleccione **Activar** y su dispositivo tratará de conectarse un máximo de diez veces después de realizar la primera llamada sin éxito. Para detener la función marcación automática, pulse .

Resumen tras llam.: active esta configuración si desea que el dispositivo muestre brevemente la duración aproximada de la última llamada.

Marcación rápida: seleccione **Activar** y los números asignados a las teclas de marcación rápida (de **2** a **9**) pueden ser marcados al mantener pulsada la tecla. También consulte "Marcación rápida de un número telefónico", pág. 160.

Cualq. tecla contesta: seleccione **Activar** y puede contestar las llamadas entrantes mediante la pulsación de cualquier tecla en la tapa.

Línea en uso (servicio de red): esta configuración sólo aparece si la tarjeta SIM admite dos números de suscriptor, es decir, dos líneas telefónicas. Seleccione la línea telefónica que desea usar para realizar llamadas y enviar mensajes de texto. Podrá contestar las llamadas para ambas líneas sin importar la línea seleccionada. Si selecciona **Línea 2** sin suscribirse a este servicio de red, no podrá realizar llamadas. Cuando selecciona la línea 2, **2** aparece en el modo standby.

Sugerencia: Para intercambiar entre las líneas telefónicas, mantenga pulsada la tecla **#** en el modo standby.

Cambio de línea (servicio de red): para evitar el cambio de línea, seleccione **Cambio de línea** > **Desactivar** si es admitido por su tarjeta SIM. Para cambiar esta configuración, necesitará el código PIN2.

Conexión

Conexiones de datos y puntos de acceso

Su dispositivo admite las conexiones de paquetes de datos ( ), como GPRS en la red GSM.

Glosario: GPRS usa la tecnología de datos de paquete con la cual la información se envía en pequeños paquetes de datos sobre la red móvil.

Para establecer una conexión de datos, se requiere un punto de acceso. Puede fijar varios tipos de puntos de acceso, como:

- Punto de acceso MMS para enviar y recibir mensajes multimedia
- Punto de acceso para la aplicación Web para ver las páginas WML o XHTML
- Punto de acceso a Internet (IAP) para enviar y recibir e-mail

Consulte a su proveedor de servicios sobre el punto de acceso necesario para el servicio al que desea acceder. Para más detalles sobre disponibilidad y suscripción a los servicios de paquetes de datos, comuníquese con su operador de red o proveedor de servicios.

Conexiones de paquetes de datos en las redes GSM y UMTS

Cuando usa su dispositivo en las redes GSM y UMTS, podrá tener activas conexiones múltiples de datos simultáneamente y los puntos de acceso pueden compartir una conexión de datos. En la red UMTS, las conexiones de datos permanecen activas durante las llamadas de voz. Para verificar las conexiones de datos activas, consulte "Administrador de conexiones", pág. 222.

Los siguientes indicadores pueden aparecen debajo del indicador de la señal, según la red que use:

- red GSM, paquetes de datos está disponible en la red.
- red GSM, conexión de paquetes de datos activa, se están transfiriendo datos.
- red GSM, múltiples conexiones de paquetes de datos activas.
- red GSM, conexión de paquetes de datos en espera. (por ejemplo, esto puede suceder durante una llamada de voz)
- red UMTS, paquetes de datos está disponible en la red.
- red UMTS, conexión de paquetes de datos activa, se están transfiriendo datos.
- red UMTS, múltiples conexiones de paquetes de datos activas.
- red UMTS, conexión de paquetes de datos en espera.

Puntos de acceso

Algunos o todos los puntos de accesos pueden estar preconfigurados en su dispositivo por su proveedor de servicios u operador de red y es posible que no pueda cambiarlos, crearlos, editarlos ni eliminarlos. indica un punto de acceso protegido. También puede recibir configuraciones de punto de acceso en un mensaje de su proveedor de servicios. Consulte "Datos y configuraciones", pág. 186.

Sugerencia: También consulte "Recibir configuraciones MMS y de e-mail", pág. 185, "E-mail", pág. 193 y "Punto de acceso para servicios", pág. 211.

indica un punto de acceso para paquetes de datos y un punto de acceso de LAN inalámbrica.

Para crear un punto de acceso nuevo, pulse y seleccione **Herramient.** > **Config.** > **Conexión** > **Puntos de acceso** > **Opciones** > **Nuevo punto acceso**.

Siga las instrucciones facilitadas por su proveedor de servicios.

Nombre de conexión: dé un nombre descriptivo para la conexión.

Portador de datos: dependiendo de la conexión de datos que seleccione, sólo ciertos campos están disponibles. Rellene todos los recuadros marcados con **Debe definirse** o con un asterisco rojo. Podrá dejar sin rellenar los otros campos a no ser que haya recibido otra instrucción por parte de su proveedor de servicios.

Para usar una conexión de datos, el proveedor de servicios de red debe admitir esta opción y si fuera necesario, activarla en su tarjeta SIM.

Paquetes de datos

Siga las instrucciones de su proveedor de servicios.

Nombre punto acceso (sólo para paquetes de datos): el nombre del punto de acceso se necesita para establecer una conexión a las redes de paquetes de datos y UMTS. Usted obtiene el nombre del punto de acceso de su operador de red o proveedor de servicios.

Nombre de usuario: el nombre de usuario puede ser necesario para hacer una conexión de datos y normalmente es provisto por el proveedor de servicios. Por lo general el nombre de usuario distingue entre las letras minúsculas y mayúsculas.

Petición contraseña: seleccione **Sí** si debe ingresar una contraseña nueva cada vez que tenga acceso al servidor, o si no desea guardar su contraseña en el dispositivo.

Contraseña: puede necesitar una contraseña para hacer una conexión de datos y normalmente es facilitada por el proveedor de servicios. Por lo general la contraseña distingue entre las letras minúsculas y mayúsculas.

Autenticación: seleccione **Normal** o **Segura**.

Página de inicio: dependiendo en la configuración que está programando, escriba la dirección Web o la dirección del centro de mensajería multimedia. Complete las **Config. avanzadas** para definir mejor sus detalles de acceso:

Tipo de red: seleccione el tipo de protocolo Internet a usar: **Configuraciones IPv4** o **Configuraciones IPv6**. Las otras configuraciones dependen del tipo de red seleccionada.

Dirección IP teléfono (para IPv4): ingrese la dirección IP de su dispositivo.

Dirección DNS— **DNS primario:**, **DNS secundario:** comuníquese con su proveedor de servicios Internet para obtener estas direcciones.

Glosario: El servicio de nombres de dominios (DNS) es un servicio de Internet que interpreta los nombres de dominios como www.nokia.com en direcciones IP como 192.100.124.195.

Dirección de proxy: defina la dirección para el servidor proxy.

Número puerto proxy: ingrese el número de puerto proxy.

LAN inalámbrica

Siga las instrucciones de su proveedor de servicios.

Nombre de red WLAN: seleccione **Ingresar manual.** o **Buscar redes**. Si selecciona una red existente, **Modo de red WLAN** y **Modo segur. WLAN** son determinados por las configuraciones del dispositivo de punto de acceso.

Modo de red WLAN: seleccione **Ad-hoc** para crear una red ad hoc y para permitir que dispositivos envíen y reciban datos directamente; no se necesita un dispositivo de punto de acceso LAN inalámbrica.

Modo segur. WLAN: seleccione la encriptación usada: **WEP**, **802.1x** (no es para redes ad hoc) o **WPA/WPA2**. Si selecciona **Red abierta**, no se usa la encriptación.

Las funciones WEP, 802.1x y WPA sólo pueden ser usadas si la red las admite.

Config. seguridad WLAN: ingrese la configuraciones para el modo de seguridad seleccionado:

Configuraciones de seguridad para **WEP**:

Clave WEP en uso: seleccione el número de clave WEP. Puede crear hasta cuatro claves WEP. Deberá ingresar las mismas configuraciones en el dispositivo de punto de acceso LAN inalámbrica.

Tipo de autenticación: seleccione **Abierto** o **Compartido** para el tipo de autenticación entre su dispositivo y el dispositivo de punto de acceso LAN inalámbrica.

Config. de clave WEP: ingrese **Encriptación WEP** (longitud de la clave), **Formato clave WEP** (**ASCII** o **Hexadecimal**) y **Clave WEP** (los datos de clave WEP en el formato seleccionado).

Configuraciones de seguridad para **802.1x** y **WPA/WPA2**:

Modo WPA: seleccione los métodos de autenticación: **EAP** para usar un protocolo de autenticación extensible (EAP) plug-in o **Clave prev. compart.** para usar una contraseña. Complete las configuraciones indicadas:

Config. plug-in EAP (sólo para **EAP**): ingrese las configuraciones indicadas por su proveedor de servicios.

Clave precompartida (sólo para **Clave prev. compart.**): ingrese una contraseña. Puede ingresar la misma contraseña en el dispositivo de punto de acceso LAN inalámbrica.

Encriptación TKIP: seleccione si se usa el protocolo de integridad provisional de clave (TKIP).

Página de inicio: defina una página inicial.

Seleccione **Opciones** > **Config. avanzadas** y alguna de las siguientes opciones:

Configuraciones IPv4: **Dirección IP teléfono** (la dirección IP de su dispositivo), **Máscara de subred** (la dirección IP subnet), **Gateway predeterm.** (el puerto) y **Dirección DNS**: ingrese las direcciones IP de los servidores DNS primario y secundario. Comuníquese con su proveedor de servicios Internet para obtener estas direcciones.

Configuraciones IPv6 > **Dirección DNS**: seleccione **Automático**, **Conocido** o **Def. por usuario**.

Canal ad-hoc (sólo para **Ad-hoc**): para ingresar manualmente un número de canal (1-11), seleccione **Definido por usuario**.

Dirección de proxy: ingrese la dirección para el servidor proxy.

Número puerto proxy: ingrese el número de puerto proxy.

Paquetes de datos

Las configuraciones de paquetes de datos afectan todos los puntos de acceso usando una conexión de datos de paquete.

Conex. paq. de datos: si selecciona **Si está disponible** y se encuentra en una red que admite paquetes de datos, el dispositivo se registra a la red de paquetes de datos. Además, iniciar una conexión activa de paquetes de datos, (por ejemplo, para enviar y recibir e-mail) resulta más rápido. Si selecciona **Si es necesario**, el dispositivo sólo usará una conexión de paquetes de datos si inicia una aplicación o acción que necesita dicha conexión. Si no hay cobertura para paquetes de datos y selecciona **Si está disponible**, el dispositivo trata de establecer periódicamente una conexión de paquetes de datos.

Punto de acceso: necesita el nombre para el punto de acceso para usar su dispositivo como módem de paquetes de datos en su PC.

Configuraciones

Puede recibir configuraciones seguras de servidor de su operador de red o proveedor de servicios en un mensaje de configuración o las configuraciones pueden estar almacenadas en su tarjeta SIM o USIM. Puede guardar, ver o borrar estas configuraciones en su dispositivo en **Configuraciones**.

Fecha y hora

Consulte "Configuraciones del reloj", pág. 142.

También consulte las configuraciones de idioma en "General", pág. 227.

Seguridad

Teléfono y SIM

Solicitud código PIN: cuando está activo, el código es requerido cada vez que enciende el teléfono. Es posible que algunas tarjetas SIM no permitan la desactivación del código de número de identificación personal (PIN). Consulte "Glosario de códigos PIN y de bloqueo", pág. 235.

Código PIN, **Código PIN2** y **Código de bloqueo**: puede cambiar el código de bloqueo, PIN o PIN2. Estos códigos pueden incluir sólo los números del **0** al **9**. Consulte "Glosario de códigos PIN y de bloqueo", pág. 235.

No use códigos de acceso similares a los números de emergencia, para evitar la marcación accidental de números de emergencia.

En caso de olvidar cualquiera de estos códigos, comuníquese con su proveedor de servicios.

Período bloq. auto.: puede configurar un tiempo de bloqueo automático, un tiempo de espera tras el cual el dispositivo se bloquea automáticamente y sólo se puede usar si ingresa el código de bloqueo correcto. Ingrese un número para el tiempo de espera en minutos, o seleccione **Ninguno** para desactivar el tiempo de bloqueo automático.

Para desbloquear el dispositivo, ingrese el código de bloqueo.

Cuando el dispositivo está bloqueado, pueden efectuarse llamadas al número de emergencia oficial programado.

Sugerencia: Para bloquear manualmente el dispositivo, pulse . Se abre una lista de comandos. Seleccione **Bloquear teléfono**. Para obtener más información sobre el bloqueo consulte "Bloquear el teclado (Bloqueo de teclas)", pág. 144.

Bloq. si cambia SIM: puede programar su dispositivo para solicitar el código de bloqueo cuando una tarjeta SIM desconocida es insertada en su dispositivo. El dispositivo mantiene una lista de las tarjetas SIM reconocidas como las del dueño.

Grupo cerr. usuarios (servicio de red): puede especificar un grupo de personas a las cuales puede llamar y de las cuales puede recibir llamadas.

Cuando las llamadas están limitadas a un grupo cerrado de usuarios, es posible realizar llamadas al número de emergencia oficial programado en el dispositivo.

Confirmar serv. SIM (servicio de red): puede programar el dispositivo para mostrar los mensajes de confirmación cuando está usando un servicio de tarjeta SIM.

Glosario de códigos PIN y de bloqueo
En caso de olvidar cualquiera de estos códigos, comuníquese con su proveedor de servicios.

Código número de identificación personal (PIN): este código protege su tarjeta SIM contra el uso no autorizado. El código PIN (de cuatro a ocho dígitos) es normalmente provisto con la tarjeta SIM. Tras tres ingresos consecutivos erróneos del código PIN, el código PIN es bloqueado y necesitará desbloquearlo antes de que pueda volver a usar la tarjeta SIM. Consulte la información sobre el código PUK en esta sección.

Código UPIN: este código puede ser provisto con la tarjeta USIM. La tarjeta USIM es una versión mejorada de la tarjeta SIM y es admitida por los teléfonos móviles UMTS. El código UPIN protege la tarjeta USIM contra el uso no autorizado.

Código PIN2: este código (de cuatro a ocho dígitos) es provisto con ciertas tarjetas SIM y es necesario para acceder a ciertas funciones de su dispositivo.

Código de bloqueo (también conocido como el código de seguridad): este código (de cinco dígitos) puede ser usado para bloquear el dispositivo de modo que no admita el uso no autorizado. El código de bloqueo predeterminado es **12345**. Para evitar el uso no autorizado de su dispositivo, cambie el código de bloqueo. Mantenga en secreto el código de bloqueo nuevo, en un lugar seguro y apartado del dispositivo.

Código de clave de desbloqueo personal (PUK) y código PUK2: estos códigos (de ocho dígitos) son necesarios para cambiar un código PIN o PIN2 bloqueado, respectivamente. Si los códigos no son provistos con la tarjeta SIM, comuníquese con el operador de la tarjeta SIM en su dispositivo.

Código UPUK: este código (de ocho dígitos) es necesario para cambiar un código UPIN bloqueado. Si el código no es provisto con la tarjeta USIM, comuníquese con el operador de la tarjeta USIM en su dispositivo.

Administración de certificados

Los certificados digitales no garantizan la seguridad, se usan para verificar el origen del software.

Glosario: Los certificados digitales se usan para verificar el origen de las páginas XHTML o WML y de software instalado. Sin embargo, sólo son de confianza si se sabe que el origen del certificado es auténtico.

En la vista principal de administración de certificados, podrá ver una lista de los certificados de autoridad guardados en su dispositivo. Desplácese a la derecha para ver una lista de los certificados personales, si están disponibles.

Los certificados digitales deberían ser usados si desea conectarse en línea a un banco u otro sitio o servidor remoto para cuestiones que tienen que ver con la transferencia de información confidencial. También deberían ser utilizados si desea reducir el riesgo de virus u otro software perjudicial

y para asegurarse de la autenticidad del software cuando es descargado e instalado.

Importante: A pesar de que el uso de certificados reduce el riesgo al establecer conexiones remotas e instalar software, deberán usarse correctamente para obtener todos los beneficios de seguridad mejorada. La existencia de un certificado no ofrece ninguna clase de protección por sí sola, el administrador de certificados debe poseer certificados correctos, auténticos o fiables para que haya disponible seguridad adicional. Los certificados tienen duración limitada. Si aparece **Certificado caducado** o **Certificado no válido aún** cuando el certificado debería ser válido, compruebe que la hora y fecha actuales del dispositivo sean correctos.

Ver detalles del certificado: verificar la autenticidad

Sólo puede asegurarse de la identidad correcta de un servidor cuando la firma y el período de validez del certificado de servidor han sido verificados.

Se le notifica sobre la identidad del servidor si no es auténtica o si usted no tiene el certificado de seguridad correcto en su dispositivo.

- Para verificar los detalles del certificado, desplácese a un certificado y seleccione **Opciones** > **Detalles certificado**. Cuando abra los detalles de un certificado, la validez del certificado es verificada y aparecerá una de las siguientes notas:
- **Certificado no seguro**: no ha programado ninguna aplicación para que use el certificado. Consulte "Cambiar las configuraciones de seguridad", pág. 236.
- **Certificado caducado**: el periodo de validez ha vencido para el certificado seleccionado.
- **Certificado no válido aún**: el periodo de validez aún no ha empezado para el certificado seleccionado.
- **Certificado dañado**: no se puede usar el certificado. Comuníquese con el emisor del certificado.

Cambiar las configuraciones de seguridad

Antes de cambiar cualquier configuración del certificado, deberá asegurarse de que el propietario del certificado es de plena confianza y de que el certificado pertenece realmente al propietario indicado.

Desplácese a un certificado de autoridad y seleccione **Opciones** > **Config. seguridad**. Dependiendo del certificado, aparecerá una lista de las aplicaciones que pueden utilizar el certificado seleccionado. Por ejemplo:

- **Instalación Symbian**: **Sí**: el certificado puede certificar el origen de una aplicación nueva de sistema operativo Symbian.
- **Internet**: **Sí**: el certificado puede certificar los servidores.
- **Instalación aplicac.**: **Sí**: el certificado puede certificar el origen de una aplicación Java nueva.

Seleccione **Opciones** > **Editar config.** para cambiar el valor.

Rastrear objeto protegido

Algunos archivos de música con restricciones de copyright pueden incluir un identificador de transacción. Seleccione si desea permitir que el identificador sea enviado con el archivo de música si envía o reenvía el archivo. El proveedor del archivo de música puede usar el identificador para dar seguimiento a la distribución de los archivos.

Plug-ins EAP

Puede ver los Plug-ins EAP instalados actualmente en su dispositivo (servicio de red). Algunos puntos de acceso a Internet que usan LAN inalámbrica como portador de datos y el modo de seguridad WPA usan los plug-ins para la autenticación.

Desvío de llamadas

Desvío llam. le permite desviar sus llamadas entrantes a su buzón de voz u otro número telefónico. Para más detalles, comuníquese con su proveedor de servicios.

1 Pulse y seleccione **Herramient.** > **Config.** > **Desvío llam.**.

2 Seleccione las llamadas que desea desviar: **Llamadas de voz**, **Llam. datos/video**, o **Llamadas de fax**.

3 Seleccione la opción de desvío deseada. Para desviar llamadas de voz cuando su número esté ocupado o cuando rechace llamadas, seleccione **Si está ocupado**.

4 Programe la opción de desvío como activada (**Activar**) o desactivada (**Cancelar**) o verifique si la opción está activada (**Verificar estado**). Varias opciones de desvío pueden estar activadas a la vez.

Al desviar las llamadas, aparece en el modo standby.

Las restricciones de llamadas y desvío de llamadas no pueden activarse al mismo tiempo.

Restricción de llamadas

Restric. llam. (servicio de red) le permite restringir las llamadas realizadas o recibidas por el dispositivo. Para cambiar las configuraciones, necesita la contraseña de restricción de su proveedor de servicios. Seleccione la opción de restricción deseada y configúrela como activada (**Activar**) o desactivada (**Cancelar**) o verifique si la opción está activada (**Verificar estado**). **Restric. llam.** afecta a todas las llamadas, incluso las de datos.

Las restricciones de llamadas y desvío de llamadas no pueden activarse al mismo tiempo.

Cuando las llamadas están restringidas, es posible realizar llamadas a ciertos números de emergencia.

Red

Su dispositivo puede cambiar entre las redes GSM y UMTS automáticamente. La red GSM es indicada por en el modo standby. Las red UMTS es indicada por 3G.

Modo de red (sólo aparece si es admitido por el operador de red): seleccione la red que usará. Si selecciona **Modo dual**, el dispositivo usa las redes GSM o UMTS automáticamente, según los parámetros de la red y los acuerdos de roaming entre los operadores de red. Comuníquese con su operador de red para obtener más detalles.

Selección operador: seleccione **Automática** para configurar el dispositivo para buscar y seleccionar una de las redes disponibles o **Manual** para seleccionar manualmente la red de una lista de redes. Si pierde la conexión a la red seleccionada manualmente, el dispositivo emitirá un tono de error y le pedirá que vuelva a seleccionar una red. La red seleccionada debe tener un acuerdo de roaming con su sistema de red local, es decir, con el operador otorgante de la tarjeta SIM en su dispositivo.

Glosario: Un acuerdo de roaming es un acuerdo entre dos o más proveedores de servicios que facilita a los usuarios de un proveedor de servicios el uso de los servicios de otros proveedores.

Mostrar info. célula: seleccione **Activar** para que el dispositivo muestre cuando es usado en una red celular basada en la tecnología red microcelular (MCN-Micro Celular Network) y para activar la recepción de información de célula.

Configuraciones de accesorios

Indicadores en el modo standby:

un auricular está conectado.

el auricular no está disponible, o se ha perdido la conexión a un auricular Bluetooth.

Seleccione **Equipo auricular** o **Man. lib. Bluetooth** y las siguientes opciones están disponibles:

Modo predet.: configure el perfil que desea activar cada vez que conecta cierto accesorio a su dispositivo. Consulte "Modos: configurar tonos", pág. 156.

Respuesta auto.: para programar el dispositivo para que conteste las llamadas entrantes automáticamente después de cinco segundos. Si ha programado el tipo de timbre a **Un bip** o **Silencio**, la respuesta automática queda desactivada.

Comandos de voz

Puede usar los comandos de voz para controlar su dispositivo. Para obtener más información sobre los comandos de voz mejorados admitidos por su dispositivo, consulte "Marcación por voz", pág. 160.

Para activar los comandos de voz mejorados para iniciar aplicaciones o perfiles, deberá abrir la aplicación **Com. voz** y su carpeta **Perfiles**. Pulse y seleccione **Herramient.** > **Com. voz** > **Perfiles**. El dispositivo crea comandos de voz para las aplicaciones y perfiles. Ahora puede usar los comandos de voz mejorados al mantener pulsada la tecla de selección derecha (—) y decir un comando de voz en el modo standby. El comando de voz es el nombre de la aplicación o perfil mostrado en la lista.

Para agregar más aplicaciones a la lista, seleccione **Opciones** > **Nueva aplicación**. Para agregar un segundo comando de voz que se pueda usar para iniciar la aplicación, desplácese a ésta, seleccione **Opciones** > **Cambiar comando** e ingrese el nuevo comando de voz como texto. Evite usar nombres muy cortos, abreviaturas y acrónimos.

Para cambiar las configuraciones de los comandos de voz, seleccione **Opciones** > **Configuraciones**. Para desactivar el sintetizador que reproduce las etiquetas de voz y comandos reconocidos en el idioma seleccionado del dispositivo, seleccione **Sintetizador** > **Desactivar**. Para reprogramar el aprendizaje de reconocimiento de voz, por ejemplo, cuando el usuario principal del teléfono ha cambiado, seleccione **Restablecer adapt. voz**.

Administrador de aplicaciones

Pulse y seleccione **Herramient.** > **Adm. aplic.** Puede instalar dos tipos de aplicaciones y software en su dispositivo.

- Las aplicaciones J2ME™ basadas en la tecnología Java™ con la extensión .jad o .jar ( ).
- Otras aplicaciones y software apropiados para el sistema operativo Symbian ( ). Los archivos de instalación tienen la extensión .sis. Sólo instale software específicamente diseñado para su Nokia N91. Los proveedores de software con frecuencia se refieren al número de modelo oficial de este producto: el Nokia N91-1.

Los archivos de instalación pueden transferirse a su dispositivo desde una computadora compatible, descargados durante la navegación o enviados en un mensaje multimedia, como un archivo adjunto de e-mail o usando la conectividad Bluetooth. Puede usar Nokia Application Installer en Nokia PC Suite para instalar una aplicación en su dispositivo. Si usa Microsoft Windows Explorer para transferir un archivo, guarde el archivo en el disco duro de su dispositivo (disco local).

Ejemplo: Si ha recibido el archivo de instalación como un archivo adjunto de e-mail, desplácese a su buzón de correo, abra el e-mail, abra la vista de adjuntos, desplácese al archivo de instalación y pulse el joystick para iniciar la instalación.

Instalar aplicaciones y software

Sugerencia: También puede usar Nokia Application Installer disponible en Nokia PC Suite para instalar aplicaciones. Consulte el CD-ROM suministrado con el dispositivo.

Los iconos de aplicación son los siguientes:

aplicación .sis

aplicación Java

la aplicación no está completamente instalada.

Importante: Sólo instale y utilice aplicaciones y otros software de fuentes que brinden seguridad y protección adecuada contra software perjudicial.

Antes de instalar, tenga en cuenta lo siguiente:

- Para ver el tipo de aplicación, número de versión, detalles del certificado de seguridad y el suministrador o fabricante de la aplicación, seleccione **Opciones** > **Ver detalles**.
- Si instala un archivo con una actualización o reparación de una aplicación existente, sólo podrá restaurar la aplicación original si tiene el archivo de instalación original o una copia completa de seguridad del paquete de software retirado. Para restaurar la aplicación original, retire primero la aplicación y de nuevo instale la aplicación del archivo de la instalación original o de la copia de seguridad.
- El archivo .jar es necesario para instalar las aplicaciones Java. Al carecer del mismo, el dispositivo le indica que lo descargue. Si no hay punto de acceso definido para la aplicación, se le indica que seleccione uno. Cuando esté descargando el archivo .jar, puede ser necesario que ingrese el nombre y contraseña del usuario para acceder al servidor. Puede solicitar estos datos al proveedor o fabricante de la aplicación.

Para instalar aplicaciones y software complete estos pasos:

1 Abra **Adm. aplic.** y desplácese al archivo de instalación. Como alternativa, busque en la memoria del dispositivo o disco duro en **Adm. arch.** o abra un mensaje en **Mensajes** > **Buzón entrada** que contenga el archivo de instalación.

2 Seleccione la aplicación y pulse el joystick para iniciar la instalación.

Sugerencia: Mientras explora, puede descargar un archivo de instalación e instalarlo sin cerrar la conexión.

Durante la instalación, el dispositivo muestra información sobre el estado de la instalación. Si está instalando una aplicación sin una firma digital o certificado, el dispositivo

muestra un aviso. Sólo continúe la instalación si está seguro del origen y contenido de la aplicación.

Para iniciar una aplicación instalada, localícela en el menú y pulse el joystick.

Para empezar una conexión de red y ver datos adicionales sobre la aplicación, desplácese a ésta y seleccione **Opciones** > **Ir a dirección Web**, si está disponible.

Para ver los paquetes de software que han sido instalados, retirados y cuándo, seleccione **Opciones** > **Ver registro**.

Para enviar su registro de instalación a un escritorio de ayuda para ver qué ha sido instalado o retirado, seleccione **Opciones** > **Enviar registro** > **Vía mensaje corto** o **Vía e-mail** (sólo disponible con las configuraciones de e-mail correctas).

Eliminar aplicaciones y software

Desplácese al paquete de software y seleccione **Opciones** > **Borrar**. Seleccione **Sí** para confirmar.

Cuando elimine el software, sólo podrá reinstalarlo si tiene el paquete de software original o una copia de seguridad completa del paquete de software que eliminó. Si elimina un paquete de software, es posible que ya no pueda abrir los documentos creados con dicho software.

Si otro paquete de software depende del paquete de software que usted eliminó, puede que aquél deje de funcionar. Para detalles, refiérase a la documentación del paquete de software.

Configuraciones de aplicación

Seleccione **Opciones** > **Configuraciones** y alguna de las siguientes opciones:

Verif. certif. en línea: seleccione para verificar los certificados en línea antes de instalar una aplicación.

Dirección Web pred.: programe la dirección predeterminada usada al verificar los certificados en línea.

Ciertas aplicaciones Java pueden requerir una llamada telefónica, enviar un mensaje, o una conexión de red a un punto de acceso específico para descargar componentes o datos adicionales. En la vista principal de **Adm. aplic.** desplácese a la aplicación y seleccione **Opciones** > **Config. Suite** para cambiar las configuraciones asociadas a la aplicación específica.

Memoria del disco duro

Para ver la cantidad de espacio usado y libre en el disco duro del dispositivo, pulse y seleccione **Herramient.** > **Disco duro**. Para ver cuánto espacio del disco las distintas aplicaciones y tipos de datos usan, seleccione **Opciones** > **Detalles de memoria**.

Para cambiar el nombre del disco duro, seleccione **Opciones** > **Nombre de disco duro**.

Herramientas

Use un cargador al escanear, defragmentar o formatear el disco duro.

Escanear el disco duro

Puede escanear su disco duro para encontrar y solucionar posibles errores. Los problemas se arreglan automáticamente.

Para escanear su disco duro, seleccione **Opciones** > **Herramientas** > **Explorar disco duro**. Aparece una barra de progreso.

Tras terminar de escanear se le informan los resultados.

Defragmentar el disco duro

Se recomienda escanear el disco duro antes de defragmentarlo.

Para optimizar el disco duro, seleccione **Opciones** > **Herramientas** > **Defragmentar disco**.

Se le informará cuando termine la defragmentación.

Formatear el disco duro

Cuando se formatea el disco duro, se pierden todos los datos en el disco. Haga una copia de seguridad de los datos que desea conservar antes de formatear el disco duro. Puede usar Nokia PC Suite para hacer una copia de seguridad de sus datos en una PC compatible. Si necesita reparar o reemplazar su dispositivo, se requiere una copia de seguridad para restaurar los datos protegidos por las tecnologías de manejo de derechos digitales (DRM).

Puede ocurrir que el disco duro se corrompa y necesite ser formateado para volver a tener funcionalidad completa. Formatear elimina todos los datos del disco duro. Formatear el disco duro aumenta la demanda de alimentación de la batería. Se le aconseja no usar su dispositivo en ninguna otra operación y que lo conecte a un cargador y mantener el cargador conectado mientras su dispositivo formatea el disco duro.

Para formatear el disco duro, seleccione **Opciones** > **Formato rápido** o **Formato completo**. **Formato completo** escanea la unidad en búsqueda de errores antes de formatear. **Formato rápido** borra toda la información del disco sin antes escanearlo. Use esta opción para un disco ya formateado y si está seguro que el disco no está dañado.

Las tecnologías de manejo de derechos digitales pueden impedir restaurar copias de seguridad. Comuníquese con su proveedor de servicios para obtener más información sobre el uso de DMR para su contenido.

Manejo de derechos digitales

Los propietarios de contenido pueden usar distintos tipos de tecnologías para el manejo de derechos digitales (DRM) para proteger su propiedad intelectual, incluso copyrights. Este dispositivo usa software DRM aplicable para acceder al contenido protegido DRM. Si un software DRM falla

en su protección del contenido, el propietario del contenido puede solicitar que se revoque la capacidad del software de acceder al contenido nuevo protegido por dicho DRM. Revocar no afecta el uso del contenido que ya está en su dispositivo. Con este dispositivo puede acceder a contenido protegido con OMA 1.0 y OMA 2.0.

Para ver las claves de activación de contenido DMR protegido guardado en su dispositivo, pulse y seleccione **Personal** > **Clave activ.** y de las siguientes opciones:

Claves válidas: ver claves que están conectadas a uno o más archivos multimedia y las claves para las cuales no ha comenzado su período de activación.

Claves no válidas: ver claves que no son válidas, el período de uso para el archivo multimedia ha llegado a su límite o hay un archivo multimedia protegido en el dispositivo pero no está conectada la clave de activación.

Claves sin usar: ver claves que no tienen archivos multimedia conectados en el dispositivo.

Para comprar más tiempo de uso o prolongar la duración de uso para un archivo multimedia, seleccione una clave de activación no válida y **Opciones** > **Obtener clave activac.** No puede actualizar las teclas de activación si el servicio Web para recepción de mensajes está desactivado. Consulte "Mensajes de servicio Web", pág. 187.

Para ver información en detalle, como estado de validez y capacidad para enviar el archivo, desplácese a una clave de activación y pulse el joystick.

Solución de problemas

Preguntas y respuestas

Conectividad Bluetooth

P: ¿Por qué no puedo encontrar el dispositivo de mi amigo?

R: Asegúrese de que ambos dispositivos tienen activada la conectividad Bluetooth.

Asegúrese de que la distancia entre los dos dispositivos no es más de 10 metros (33 pies) y que no existen paredes u obstrucciones entre los dispositivos.

Asegúrese de que el otro dispositivo no está programado en el modo oculto.

Verifique que ambos dispositivos sean compatibles.

P: ¿Por qué no puedo finalizar una conexión Bluetooth?

R: Si tiene otro dispositivo conectado a su teléfono, puede finalizar la conexión usando el otro dispositivo o desactivar la conectividad Bluetooth. Seleccione **Conect.** > **Bluetooth** > **Desactivado**.

Mensajes multimedia

P: ¿Qué debería hacer cuando el teléfono no puede recibir un mensaje multimedia puesto que la memoria está llena?

R: La cantidad de memoria necesaria es indicada en este mensaje de error: **No hay memoria suficiente para recuperar el mensaje. Elimine antes algunos datos.** Para ver qué tipo de datos tiene y cuánta memoria consumen los diferentes grupos de datos, seleccione **Adm. arch.** > **Opciones** > **Detalles memoria**.

P: La nota aparece brevemente **Recuperando mensaje**. ¿Qué está sucediendo?

R: El dispositivo está tratando de recuperar un mensaje multimedia del centro de mensajería multimedia. Asegúrese de que las configuraciones de mensajería multimedia estén definidas correctamente y de que no existan errores en los números telefónicos o direcciones. Seleccione **Mensajes** > **Opciones** > **Configuraciones** > **Mensaje multmedia**.

Mensajería

P: ¿Por qué no puedo seleccionar ningún contacto?

R: La tarjeta de contacto no tiene un número de teléfono o una dirección de e-mail. Agregue la información necesaria a la tarjeta de contacto en **Contactos**.

Cámara

P: ¿Por qué aparecen borrosas las imágenes?

R: Asegúrese de que la ventanilla protectora del lente de la cámara esté limpia.

Agenda

Q: ¿Por qué no hay números de semanas?

A: Si cambió las configuraciones de agenda para que empiece en otro día, aparte de lunes, no aparecerán los números.

Servicios del navegador

P: ¿Qué hago si este mensaje aparece: **Sin punto de acceso válido. Defínalo en configuraciones de Servicios.**?

R: Inserte las configuraciones de explorador correctas. Comuníquese con su proveedor de servicios para instrucciones.

Registro

P: ¿Por qué está vacío el Registro?

R: Puede que haya activado un filtro y ningún evento de comunicación aplicable a este filtro haya sido registrado. Para ver todos los eventos, seleccione **Personal** > **Registro** > **Opciones** > **Filtrar** > **Toda comunicación**.

P: ¿Cómo borro mi información de registro?

R: Seleccione **Personal** > **Registro** > **Opciones** > **Borrar registro** o desplácese a **Configuraciones** > **Duración del registro** > **Ningún registro**. Esto borrará permanentemente el contenido del registro, las llamadas recientes y los mensajes de entrega de mensajes.

Conectividad PC

P: ¿Por qué tengo problemas al conectar el dispositivo a mi PC?

R: Asegúrese de que Nokia PC Suite está instalado y está funcionando en su PC. Consulte el manual del usuario para Nokia PC Suite en el CD-ROM. Si Nokia PC Suite está instalado y funcionando, puede usar el asistente Nokia Get connected disponible en Nokia PC Suite para conectarse a su PC. Para obtener más información sobre cómo usar Nokia PC Suite, consulte la función de ayuda en PC Suite o consulte las páginas de soporte en www.nokia.com en EE.UU y http://latinoamerica2.ext.nokia.com/pcsuite para Latinoamérica.

Códigos de acceso

P: ¿Cuál es mi contraseña para los códigos de bloqueo, PIN o PUK?

R: El código de bloqueo original es **12345**. Si no se acuerda o ha extraviado el código de bloqueo, comuníquese con el distribuidor de su teléfono.

Si no se acuerda o pierde un código PIN o PUK o si no lo ha recibido, comuníquese con su proveedor de servicios de red. Para obtener más información sobre contraseñas, comuníquese con su proveedor de punto de acceso, por ejemplo, un proveedor de servicios comercial de Internet (ISP), proveedor de servicios u operador de red.

Una aplicación que no responde

P: ¿Cómo cierro una aplicación que no responde?

R: Abra la ventana para cambiar de aplicaciones por medio de mantener pulsada . Luego desplácese hasta la aplicación y pulse para cerrar la aplicación.

Pantalla del teléfono

P: ¿Por qué aparecen en la pantalla puntitos faltantes, descoloridos o luminosos cada vez que enciendo mi teléfono?

R: Esto es una característica de este tipo de pantalla. Algunas pantallas podrían contener píxeles o puntitos que permanecen activados o desactivados. Esto es normal, no es un defecto.

Memoria baja

P: ¿Qué hago si mi dispositivo tiene la memoria baja?

R: Puede borrar los siguientes elementos regularmente para evitar consumir toda la memoria:

- Mensajes de las carpetas **Buzón entrada**, **Borrador** y **Enviados** en **Mensajes**
- Mensajes de e-mail recuperados de la memoria del dispositivo
- Páginas Web almacenadas
- Imágenes y fotos en **Galería**

Para borrar información de contacto, notas de agenda, cronómetros de llamadas, contadores de costo de llamadas, puntajes de juegos o cualquier otro dato, desplácese a la aplicación respectiva para borrar los datos. Si está borrando elementos múltiples y una de estas notas aparece: **No hay suficiente memoria para efectuar operación. Elimine antes datos.** o **Memoria baja. Elimine datos.**, trate de eliminar los elementos uno a uno (comience con los elementos más pequeños).

P: ¿Cómo puedo guardar mis datos antes de borrarlos?

R: Guarde sus datos usando alguno de los siguientes métodos:

- Use Nokia PC Suite para hacer copias de seguridad de todos los datos en una computadora compatible.
- Envíe imágenes a su dirección de e-mail y luego guarde las imágenes en su computadora.
- Envíe datos usando una conexión Bluetooth a un dispositivo compatible.
- Mueva los datos de la memoria del dispositivo al disco duro de su dispositivo.

Seguridad del producto

Información sobre las baterías

Carga y descarga

Su dispositivo recibe alimentación eléctrica a través de una batería recargable. El rendimiento óptimo de una batería nueva se obtiene solamente después de dos o tres ciclos completos de carga y descarga. La batería se puede cargar y descargar cientos de veces pero con el tiempo se gastará. Cuando los tiempos de conversación y de reserva sean notablemente más cortos de lo normal ya es momento de adquirir una batería nueva. Utilice únicamente las baterías aprobadas por Nokia y recárguelas sólo con los cargadores aprobados por Nokia para este dispositivo.

Desconecte el cargador del enchufe eléctrico y del dispositivo cuando no esté en uso. No deje la batería conectada al cargador tras cargarla completamente, puesto que la sobrecarga podría acortar la vida útil de la batería. Si una batería totalmente cargada no se utiliza por un tiempo, se descargará poco a poco.

Si la batería está completamente descargada, podrían pasar unos pocos minutos antes de que el indicador de carga aparezca en la pantalla o antes de poder realizar alguna llamada.

Utilice la batería solamente para el propósito para el que fue diseñada. No utilice nunca el cargador o la batería si están dañados.

No cause cortocircuito en la batería. Puede producirse un cortocircuito accidental si un objeto metálico como una moneda, clip o bolígrafo ocasiona la conexión directa de los terminales positivos (+) y negativos (-) de la batería. (Estas parecen tiras metálicas en la batería). Puede ocurrir, por ejemplo, cuando lleva una batería de repuesto en su bolsillo o cartera. El cortocircuito de las terminales puede dañar la batería o el objeto que esté conectado a ella.

Dejar la batería en lugares calurosos o fríos, como dentro de un automóvil cerrado y expuesto a la intemperie veraniega o invernal reducirá su capacidad, vida útil y su rendimiento de carga. Intente mantener la batería siempre entre los 15 °C y 25 °C (59 °F y 77 °F). Un dispositivo con la batería demasiado caliente o fría puede dejar de funcionar temporalmente, aun cuando la batería se encuentre completamente cargada. El rendimiento de la batería se ve particularmente limitado en temperaturas inferiores al punto de congelación.

No tire las baterías al fuego, ya que pueden explotar. También pueden explotar si están dañadas. Elimine las baterías conforme a las normativas locales. Recíclelas siempre que sea posible. No deseche las baterías junto con los demás desperdicios domésticos.

Tiempos de funcionamiento

La batería BL-5C provee hasta 3-4 horas de tiempo de conversación y hasta 7-8 días de tiempo de reserva cuando el dispositivo está en operación en una red GSM. La batería BL-5C provee hasta 2,5-3 horas de tiempo de conversación y hasta 7 días de tiempo de reserva cuando el dispositivo está en operación en una red WCDMA. La batería BL-5C provee hasta 10 horas de reproducción de música o radio. Los tiempos de operación son estimaciones y podrían variar dependiendo de las condiciones de la red, carga y uso del dispositivo. Cargar la batería con el cargador AC-4 toma aproximadamente 2 horas mientras el dispositivo está en el modo standby.

Normas de autenticación de baterías Nokia

Siempre use baterías auténticas de Nokia para su seguridad. Para asegurarse de contar con una batería original Nokia, cómprela en un centro de distribución autorizado por Nokia, busque el logotipo Nokia Original Enhancements en el empaque e inspeccione la etiqueta holográfica realizando los siguientes pasos:

Realizar con éxito los cuatro pasos no asegura totalmente la autenticidad de la batería. Si tiene alguna razón para creer que su batería no es batería auténtica y original de Nokia, evite usarla y llévela al centro de servicio autorizado Nokia más cercano o a un distribuidor Nokia para pedir asistencia. Su distribuidor o centro de servicio Nokia inspeccionará la batería para confirmar su autenticidad. Ante la imposibilidad de verificar la autenticidad de la batería, devuélvala al lugar de compra.

Autenticación del holograma

1 Al mirar el holograma en la etiqueta, debería ver el símbolo de Nokia: las manos conectadas desde un ángulo y el logotipo Nokia Original Enhancements desde otro ángulo.

2 Cuando mueve el holograma a un ángulo hacia el lado izquierdo, derecho, arriba y abajo, verá 1, 2, 3 y 4 puntitos en cada lado respectivamente.

3 Raspe el lado de la etiqueta para revelar un código de 20 dígitos, por ejemplo, 12345678919876543210. Gire la batería de tal forma que los números se vean hacia arriba. El código de 20 dígitos se lee empezando con el número de la fila superior seguido por la fila inferior.

4 Confirme la validez del código de 20 dígitos realizando las instrucciones en **www.nokia.com/batterycheck** para EE.UU y **http://latinoamerica2.ext.nokia.com/verifiquesubateria** para Latinoamérica.

¿Qué ocurre si la batería no es auténtica?

Si no puede confirmar que su batería Nokia con el holograma en la etiqueta es una batería auténtica de Nokia, no la use. Llévela al centro de servicio autorizado Nokia más cercano o distribuidor para pedir asistencia. El uso de una batería no aprobada por el fabricante podría ser peligroso y podría resultar en un rendimiento inferior y perjudicar a su dispositivo y sus accesorios. También, podría anular cualquier aprobación o garantía aplicable al dispositivo.

Para obtener más información sobre las baterías originales Nokia, visite **www.nokia.com/batterycheck** para EE.UU y **http://latinoamerica2.ext.nokia.com/verifiquesubateria** para Latinoamérica.

Cuidado y mantenimiento

Su dispositivo es un producto de diseño y fabricación excelentes y debe tratarse con cuidado. Las sugerencias a continuación le ayudarán a proteger la cobertura de su garantía y disfrutar de este dispositivo por muchos años.

- Mantenga el dispositivo seco. Las precipitaciones, la humedad y los líquidos contienen minerales que pueden oxidar los circuitos electrónicos. Si su dispositivo se moja, extraiga la batería y espere que el dispositivo se seque por completo para volver a colocarla.
- No utilice ni guarde el dispositivo en lugares sucios o polvorientos. Las piezas movibles y los componentes electrónicos podrían dañarse.
- No guarde el dispositivo en lugares calurosos. Las altas temperaturas pueden reducir la vida útil de los dispositivos electrónicos, dañar las baterías y deformar o derretir algunos tipos de plástico.
- No guarde el dispositivo en lugares fríos. Cuando el dispositivo recupera su temperatura normal puede formarse humedad en su interior, la que puede dañar las placas de los circuitos electrónicos.
- Abra siempre el dispositivo de acuerdo con las instrucciones que figuran en este manual.

- No deje caer, golpee ni mueva bruscamente el dispositivo. Los manejos bruscos pueden dañar las tarjetas interiores de circuitos y mecanismos delicados.
- No utilice productos químicos abrasivos, solventes de limpieza ni detergentes para limpiarlo.
- No pinte el dispositivo. La pintura puede obstruir las piezas movibles e impedir un funcionamiento apropiado.
- Use sólo un paño suave, limpio y seco para limpiar cualquier lente (como el lente de la cámara y de los detectores de proximidad y de luz).
- Utilice sólo la antena suministrada o un repuesto aprobado. El uso de antenas, modificaciones o elementos no aprobados podrían dañar el dispositivo e infringir los reglamentos sobre el uso de dispositivos de radio.
- No use los cargadores a la intemperie.
- Haga siempre copias de seguridad de los datos que desea guardar (como contactos y notas de agenda) antes de enviar su dispositivo a un centro de servicio.

Todas estas recomendaciones aplican de igual manera a su dispositivo, batería, cargador o cualquier accesorio. Si algún dispositivo no funciona, llévelo a su centro de servicio especializado más cercano.

Información adicional de seguridad

Entorno operativo

Respete la normativa especial vigente en la zona donde se encuentra y siempre apague su dispositivo cuando esté prohibido utilizarlo o cuando su uso pueda causar interferencia o peligro. Use el dispositivo en su posición normal de funcionamiento. Este dispositivo cumple con las normas de exposición RF cuando es usado en su posición normal de funcionamiento cerca del oído o portado a una distancia mínima de 1,5 centímetros (5/8 de pulgada) del cuerpo. Al portar el teléfono en accesorios como un estuche, clip de cinturón o soporte, utilice sólo accesorios que no contengan metal y que coloquen el dispositivo a la distancia mínima del cuerpo arriba indicada.

Para transmitir archivos de datos o mensajes, este dispositivo requiere una conexión de calidad a la red. En algunos casos, la transmisión de archivos de datos o mensajes puede demorarse hasta que esté disponible la conexión adecuada. Asegúrese de seguir las instrucciones anteriores sobre las distancias de separación hasta que la transmisión haya finalizado.

El dispositivo tiene ciertas partes magnéticas. Los materiales metálicos pueden ser atraídos por el dispositivo y las personas que usan dispositivos auditivos no deben sujetar el dispositivo en el oído con el dispositivo auditivo. No coloque tarjetas de crédito u otros artículos de almacenamiento magnético cerca del dispositivo, dado que la información almacenada puede borrarse.

Dispositivos médicos

Los aparatos de radiotransmisión, incluso los teléfonos móviles, pueden interferir con el funcionamiento de dispositivos médicos que no estén protegidos adecuadamente. Consulte con un médico o con el fabricante del dispositivo médico si tiene alguna pregunta o para determinar si está correctamente protegido contra las señales externas de radiofrecuencia. Apague su teléfono en los lugares específicos de los centros de salud donde se le indique. Es probable que en los hospitales y centros de salud se utilicen equipos sensibles a las señales externas de radiofrecuencia.

Marcapasos. Los fabricantes de marcapasos recomiendan que se mantenga una separación mínima de 15,3 cm (6 pulgadas) entre un teléfono móvil y el marcapasos a fin de minimizar la potencia para interferencias. Estos consejos provienen de la investigación independiente y recomendaciones de la Investigación tecnológica en telefonía móvil (Wireless Technology Research). Las personas con marcapasos:

- Deben mantener siempre una distancia de más de 15,3 cm (6 pulgadas) entre su teléfono y el marcapasos
- No deben llevar el teléfono en el bolsillo superior de la camisa y
- Deben utilizar el dispositivo en el oído contrario al marcapasos para minimizar las posibles interferencias.

Si tiene alguna razón para sospechar que existen interferencias, debería apagar inmediatamente su dispositivo.

Aparatos auditivos. Algunos dispositivos móviles digitales pueden interferir con determinados aparatos auditivos. En ese caso, comuníquese con su proveedor de servicio.

Vehículos

Las señales de RF pueden afectar a los sistemas electrónicos que están inadecuadamente instalados o protegidos en los automóviles, por ejemplo, sistemas electrónicos de inyección directa, sistemas electrónicos de frenos antirresbalantes (antibloqueo), sistemas electrónicos de control de velocidad, o sistemas de bolsas de aire. Para obtener más información, verifique estos aspectos con el fabricante o el representante de su vehículo o de los equipos que haya incorporado posteriormente.

Sólo personal especializado debe reparar el dispositivo o instalarlo en su vehículo. Si la instalación o reparación no es correcta puede resultar peligrosa y anular cualquier garantía que se aplique al dispositivo. Compruebe con regularidad que el equipo completo del dispositivo móvil de su vehículo esté correctamente montado y funcionando debidamente. No almacene ni transporte líquidos inflamables, gases o materiales explosivos en el mismo compartimiento destinado al dispositivo, sus piezas o accesorios. Para los automóviles con sistemas de bolsas de aire incorporado, recuerde que la bolsa de aire se infla con mucha fuerza. No coloque objetos, incluso equipos instalados o celulares en el área encima de la bolsa de aire o en la zona donde ésta pueda desplegarse. Si el equipo de telefonía móvil no está bien instalado y las bolsas de aire se despliegan, puede ocasionar lesiones graves.

Se prohíbe el uso de su dispositivo a bordo de un avión. Apague el dispositivo antes de embarcar un avión. Su uso puede resultar peligroso para el funcionamiento del avión, interrumpir la red del teléfono móvil y puede ser ilegal.

Entornos potencialmente explosivos

Apague el dispositivo en áreas donde puedan producirse explosiones y obedezca todas las señales e instrucciones. Las zonas con potencial explosivo incluyen las zonas donde existan avisos reglamentarios requiriéndole que apague el motor de su vehículo. Las chispas en tales zonas pueden producir explosiones o incendios que pueden resultar en daño corporal o incluso la muerte. Apague el dispositivo en lugares de abastecimiento de combustible tales como cerca de estaciones de servicio o gasolineras. Respete las restricciones de uso de equipos de radio en los depósitos de combustible, áreas de almacenamiento y distribución de combustible, plantas químicas o lugares donde se realicen explosiones. Las zonas en entornos de potencia explosiva suelen estar marcadas, pero no siempre con claridad. Éstas incluyen zonas bajo cubierta de los barcos, instalaciones de transferencia o almacenamiento de productos químicos, vehículos que utilizan gas de petróleo licuado (como propano o butano), áreas donde el aire contiene elementos químicos o partículas, como grano, polvo o partículas metálicas.

Llamadas de emergencia

Importante: los teléfonos celulares, incluso este dispositivo, funcionan utilizando señales de radio, redes de servicios celulares y terrestres, así como funciones programadas por el usuario. Por lo tanto, no se puede garantizar que la conexión funcione en todas las condiciones. Nunca dependa únicamente de un dispositivo móvil para las comunicaciones importantes, como emergencias médicas.

Para hacer una llamada de emergencia:

1 Si el teléfono está apagado, enciéndalo. Compruebe que la intensidad de la señal sea adecuada.

Ciertas redes podrían requerir que una tarjeta SIM válida esté debidamente instalada en el dispositivo.

2 Pulse cuantas veces sea necesario para borrar la pantalla y preparar el teléfono para las llamadas.

3 Ingrese el número de emergencia oficial de la localidad en la que se encuentre. Los números de emergencia pueden variar en cada localidad.

4 Pulse la tecla .

Cuando realice una llamada de emergencia, recuerde dar toda la información necesaria de la forma más exacta posible. Su dispositivo móvil puede ser el único medio de comunicación en el lugar de un accidente. No finalice la llamada hasta que reciba instrucciones de hacerlo.

Advertencia: en el perfil desconectado, no se pueden realizar (ni recibir) llamadas, excepto las llamadas de emergencia, ni usar otras funciones que requieran la cobertura de red.

Información de certificación (SAR)

Este dispositivo móvil cumple con las normas sobre exposición a ondas de radio

Su dispositivo móvil es radiotransmisor y receptor. Está diseñado para no sobrepasar los límites de exposición a radiofrecuencia recomendados por las normas internacionales. Estas normas fueron desarrolladas por la organización independiente ICNIRP e incluyen un margen de seguridad diseñado para garantizar la seguridad del público, independientemente de la edad y salud.

La norma de exposición para dispositivos móviles emplea una unidad de medida denominada Specific Absorption Rate (Tasa de Absorbencia Específica - TAE). El límite de SAR estipulado en las normas ICNIRP es de 2,0 vatios/kilogramos (V/kg) promediado sobre 10 gramos de tejido corporal. Las pruebas SAR son efectuadas mediante el uso en posiciones estándares, con el teléfono transmitiendo al nivel más alto de potencia certificado, en todas las bandas de frecuencia comprobadas. El nivel SAR actual de un dispositivo en funcionamiento puede estar muy por debajo del valor máximo debido a que el dispositivo ha sido diseñado para sólo usar la potencia requerida para acceder a la red. La cifra cambia dependiendo de tantos factores como su distancia de acercamiento a la estación base de la red. Según los estándares ICNIRP, el valor SAR máximo cuando el dispositivo fue probado para uso junto al oído es de 0,48 W/kg.

El uso de accesorios de dispositivos podría resultar en diferentes valores SAR. Los valores SAR podrían variar dependiendo de los requerimientos nacionales de prueba y de reportaje y la banda de la red. Información adicional sobre SAR podría ser provista en la sección de productos en **www.nokia.com** (texto en inglés).

Su dispositivo móvil está diseñado para reunir los requerimientos que rigen la exposición a las ondas de radio los cuales fueron establecidos por la Comisión Federal de Comunicaciones (EE.UU.) y la Industria Canadiense. Estos requerimientos establecen un límite SAR de 1,6 W/kg promediado sobre 1 gramo de tejido corporal. El valor SAR máximo según la información suministrada bajo este estándar, durante la certificación del producto, cuando fue probado para uso junto al oído es de 0,74 W/kg y cuando está en contacto con el cuerpo es de 0,58 W/kg. La información sobre este modelo de dispositivo se puede encontrar en **http://www.fcc.gov/oet/fccid** al buscar en el sistema de autorización del equipo usando la ID FCC: QEYRM-43.

Información técnica

Función	Especificación
Peso	164 g (5,78 oz) con la Batería Li-Ion BL-5C
Tamaño	Volumen: 6,7 pulgadas^3 (109 cm^3) Longitud: 4,45 pulgadas (113.1 mm) Ancho: 2,17 pulgadas (55,2 mm) Espesor: 0,87 pulgadas (22 mm)
Rango de frecuencia	GSM 900 880–915 MHz (TX) 925–960 MHz (RX) GSM 1800 1710–1785 MHz (TX) 1805–1880 MHz (RX) GSM 1900 1850–1910 MHz (TX) 1930–1990 MHz (RX)
Potencia de salida del transmisor	Hasta 2 Vatios
Voltaje de la batería	3.7 V cc
Temperatura de funcionamiento	14°F hasta + 131°F (-10°C hasta + 55°C)

Índice

A

accesorios, configuraciones 238
accesos directos
  conexión web 211
administrador de archivos 143
agenda
  alarma 208
  crear ingresos 208
  sincronización PC Suite 209
alarmas
  agenda 209
  reloj de alarma 142
altavoz 143
  etiquetas de voz 161
  mientras escucha la radio 177
aplicación de ayuda 141
aplicaciones
  instalar 240
  Java 239
archivo SIS 239
archivos media
  formatos de archivo 173
archivos multimedia. Consulte *formatos de archivo.*
audífono, configuraciones 238
auricular, configuraciones 238

B

bitrate 154
Bluetooth 217
  apagar 220
  auricular 238
  conectar dos dispositivos, vincular 219
  dirección de dispositivo 219
  seguridad 218
  vincular 220
borrar memoria
  información de registro 245
borrar pantalla. Consulte *modo standby.*
brillo, pantalla 228
buzón de correo 187
  recuperar automáticamente 189
buzón de correo remoto 187

C

cable USB 221
caché, borrar 215
cámara
  ajustar configuraciones 168
  ajustar luz y color 168, 170
  configuraciones de cámara de imagen fija 168
  configuraciones de imagen 168
  configuraciones grabador de video 171
  enviar imágenes 167
  guardar videoclips 171
  modo de secuencia 168
  temporizador 168
canciones
  formatos de archivo 145
CD-ROM 221
certificados 235
chat. Consulte *mensajería instantánea.*
claves de activación 242
codecs 154
código de bloqueo 235
código PIN 235
código UPIN 235
código UPUK 235
códigos 234, 235
comandos USSD 180
conexión
  configuraciones 229
  gestor 222
conexión de computadora 221
conexiones de datos
  detalles 222
  finalizar 222
  indicadores 140

conexiones de datos de paquete 229
conexiones de paquetes de datos 233
conexiones PC 221
configuraciones
  agenda 209
  audífono 238
  auricular 238
  certificados 235
  código de bloqueo 235
  código PIN 235
  código UPIN 235
  código UPUK 235
  códigos de acceso 234
  conexión Bluetooth 218
  conexiones de datos 229
  configuraciones originales 228
  desvío de llamadas 237
  fecha y hora 234
  idioma 227
  MI 197, 198, 201
  pantalla 228
  personalizar el teléfono 156
  protector de pantalla 228
  puntos de acceso 230
  restricciones de llamadas 237
control de volumen
  altavoz 143
  durante una llamada 159
copia de seguridad 241
copiar
  contactos 165
  texto 182
correo de voz 237
cortar texto 182
cualquier tecla contesta 229

D

dirección IP 232
directorio. Consulte *tarjetas de contactos*
DNS 232
DRM 242

E

e-mail 183
  abrir 188
  buzón de correo remoto 187
  configuraciones 193
  desconectado 190
  obtener del buzón de correo 188
  recuperar automáticamente 189
  ver archivos adjuntos 189
enviar
  imágenes 167
  tarjetas de contactos, tarjetas de negocios 163

F

favoritos 211
fecha 142
formatos de archivo
  .jad 239
  .jar 239, 240
  .sis 239
  RealPlayer 173
  reproductor de música 145
formatos de archivo audio 145

G

galería
  agregar archivos a un álbum 173
  crear un nuevo álbum 173
  ver imágenes y videos 173
GPRS. Consulte *conexiones de datos de paquete.* 229

H

hora 142

I

indicadores 140
Internet. Consulte *Web.*

J

Java. Consulte *aplicaciones.*

K

Kb/s 154

L

LAN inalámbrica 138
  configuraciones de punto de acceso 232

llamadas
configuraciones 228
configuraciones de desvío de llamadas 237
desviar 237
duración 206
internacionales 159
llamada de conferencia 160
números marcados 206
recibidas 206
restringir 237
transferencia 162
usar altavoz 143
llamadas de conferencia 160

M

manejo de derechos digitales. Consulte *DRM.*
manos libres. Consulte *altavoz.*
marcación fija 164
marcación rápida 160
marcado rápido. Consulte *marcación rápida.*
memoria
borrar memoria 206
ver consumo de memoria 137
mensajería
mensajes de texto 183
mensajes multimedia 183
redactar e-mail 183
mensajería instantánea
bloquear 199
conectarse a un servidor 197
enviar mensajes instantáneos 198
grabar mensajes 199
grupos 200
mensajes de texto 183
centro de servicio 192
mensajes multimedia 183
modo SIM remoto 218
modo standby 156
multimedia
galería 172
grabar videos 170
RealPlayer 173
reproductor de música 173
sacar fotos 167
música
carátula del álbum 154
detalles 153
formatos de archivo 145
transferir 148

P

pausa 142
PC Suite
datos de agenda 209
sincronización 221
transferir archivos multimedia 138
transferir imágenes a PC 172
ver memoria para datos del teléfono 144
pegar texto 182
perfiles 156
personalización 157
puntos de acceso 229
configuraciones 230
crear 231
puntos de acceso de Internet (IAP). Consulte *puntos de acceso.*

R

radio 176, 177
radio FM 176
Radio Visual 176, 177, 178
recordatorio. Consulte *agenda, alarma.*
registro 207
registro de llamadas 206
registro de llamadas. Consulte *registro.*
reloj 142
reproductor de música
controles 145
ecualizador 146
formatos de archivo
indicadores 146
modos de reproducción 146
opciones 152
reproductor de video. Consulte *RealPlayer.*
restricciones de copyright. Consulte *claves de activación.*

S

seguridad
- código de bloqueo 235
- teléfono y SIM 234

servicio de comandos 180

servicio de nombre de dominio. Consulte *DNS.*

sincronización 223

SMS. Consulte *mensajes de texto.*

software 239

sonido
- grabar sonidos 176

sonidos
- silenciar un tono de timbre 161

T

tarjeta de memoria 241

tarjeta SIM
- copiar contactos 165
- mensajes 191
- nombres y números 164

tarjetas de contactos
- enviar 163
- guardar tonos DTMF 162
- insertar fotos 163

temas 157

tienda de música 147

tonos de timbre
- recibir en un mensaje de texto 187
- tono de timbre personal 165

transferir archivos 239

V

VBR 154

videoclips 172

voz
- buzón de voz 159
- comandos 239
- correo de voz 237
- etiquetas 161, 239
- marcación 160
- mensajes 159

W

Web
- navegador 211

Notas

Notas

Notas

Notas

Notas

https://web.archive.org/web/20070423000553/http://nds2.nokia.com/files/support/nam/phones/guides/N75_US_en.PDF

cingular
raising the bar

User Guide

NOKIA
N75

Some of the contents in this manual may differ from your phone depending on the software of the phone or your service provider.
Printed in Mexico.

Nokia N75 User Guide

9252449
ISSUE 1 EN

© 2006 Nokia. All rights reserved.

Nokia, Nokia Connecting People, Xpress-on, and Pop-Port are trademarks or registered trademarks of Nokia Corporation. Nokia tune is a sound mark of Nokia Corporation. Other product and company names mentioned herein may be trademarks or tradenames of their respective owners.

Reproduction, transfer, distribution, or storage of part or all of the contents in this document in any form without the prior written permission of Nokia is prohibited.

symbian

This product includes software licensed from Symbian Software Ltd (c) 1998-200(6). Symbian and Symbian OS are trademarks of Symbian Ltd.

US Patent No 5818437 and other pending patents. T9 text input software Copyright (C) 1997-2006. Tegic Communications, Inc. All rights reserved.

Java™ and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc.

This product is licensed under the MPEG-4 Visual Patent Portfolio License (i) for personal and noncommercial use in connection with information which has been encoded in compliance with the MPEG-4 Visual Standard by a consumer engaged in a personal and noncommercial activity and (ii) for use in connection with MPEG-4 video provided by a licensed video provider. No license is granted or shall be implied for any other use. Additional information, including that related to promotional, internal, and commercial uses, may be obtained from MPEG LA, LLC. See **http://www.mpegla.com**.

Nokia operates a policy of ongoing development. Nokia reserves the right to make changes and improvements to any of the products described in this document without prior notice.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, UNDER NO CIRCUMSTANCES SHALL NOKIA OR ANY OF ITS LICENSORS BE RESPONSIBLE FOR ANY LOSS OF DATA OR INCOME OR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES HOWSOEVER CAUSED.

THE CONTENTS OF THIS DOCUMENT ARE PROVIDED "AS IS". EXCEPT AS REQUIRED BY APPLICABLE LAW, NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE MADE IN RELATION TO THE ACCURACY, RELIABILITY OR CONTENTS OF THIS DOCUMENT. NOKIA RESERVES THE RIGHT TO REVISE THIS DOCUMENT OR WITHDRAW IT AT ANY TIME WITHOUT PRIOR NOTICE.

The availability of particular products and applications for these products may vary by region. Please check with your Nokia dealer for details, and availability of language options.

Export controls

This device may contain commodities, technology or software subject to export laws and regulations from the US and other countries. Diversion contrary to law is prohibited.

FCC / Industry Canada Notice

Your device may cause TV or radio interference (for example, when using a telephone in close proximity to receiving equipment). The FCC or Industry Canada can require you to stop using your telephone if such interference cannot be eliminated. If you require assistance, contact your local service facility. This device complies with part 15 of the FCC rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. Any changes or modifications not expressly approved by Nokia could void the user's authority to operate this equipment.

Issue 1 EN, 9252449

Contents

For your safety 8

Get connected 11

Welcome 12

Accessibility solutions 12
Find information about your device 12
Contact your service provider 12

Your Nokia device 13

Keys and parts (fold open) 13
Keys and parts (fold closed, front, back and sides) 14
Insert the SmartChip card and battery 15
Switch the device on 15
Modes 16
    Fold closed mode 16
    Fold open mode 17
Essential indicators 18
Useful shortcuts 19
    Idle mode 19
    Edit text and lists 19
Keypad lock (keyguard) 19
Volume and loudspeaker control 20
Headset 20
Clock 21
    Clock settings 21
    World clock 21
Wrist strap 21
Settings 22
Memory card 22
    Memory card tool 23
File manager 23
    View memory consumption 24
    Memory low—free memory 24
Music key 24
Mode key 24
Transfer content from another device 25

Support 27

Nokia support and contact information 27
Help 27
Tutorial 27

Camera 28

Take pictures 28
    Still image camera indicators 29
    Active toolbar 29
    After taking a picture 30
    Still image camera settings 31
    Flash 31
    Adjust color and lighting 31
    Scenes 32
Take pictures in a sequence 32
You in the picture—self-timer 33

Record videos ... 33
Video capture indicators ... 34
After recording a video ... 34
Video settings ... 34
Cingular Music ... 36
Music player ... 36
Play music ... 36
Random and repeat play ... 37
Playlists ... 37
Visualizations ... 39
Modify audio ... 39
Rename items ... 39
Set a song as a ringing tone ... 39
Add music ... 40
Music ID ... 41
MobiRadio ... 41
Shop Music ... 42
Shop settings ... 42
Music Video ... 42
Billboard ... 42
Community ... 43
Music Apps ... 43
Digital Rights Management ... 43
Cingular Video ... 44
Cingular Mall ... 45
My Stuff ... 46
View and browse files ... 46
Images and video clips ... 46
Active toolbar ... 47
Shrink ... 48
Print images ... 48
Albums ... 49
Edit images ... 49
Crop image ... 49
Reduce red eye ... 50
Shortcuts ... 50
Edit video clips ... 50
Add images, video, audio, and text ... 50
Cut video clip ... 51
Send videos ... 51
Movie director ... 52
Create muvees ... 52
Create a custom muvee ... 52
Slide show ... 53
Presentations ... 53
Messaging ... 54
Write text ... 55
Traditional text input ... 55
Predictive text input ... 55
Tips on text input ... 56

Write and send messages ... 57
Write and send postcards ... 58
Receive MMS and e-mail settings ... 59
Inbox—receive messages ... 59
Multimedia messages ... 59
Data and settings ... 60
Web service messages ... 60
Mailbox ... 60
Open the mailbox ... 60
Retrieve e-mail messages ... 61
Delete e-mail messages ... 62
Disconnect from the mailbox ... 62
View messages on a SmartChip card ... 62
Messaging settings ... 62
Text messages ... 63
Multimedia messages ... 63
E-mail ... 64
Web service messages ... 66
Cell broadcast ... 66
Other settings ... 66
Message reader ... 66
Address book ... 67
Contact cards ... 67
Default numbers and addresses ... 68
Contact groups ... 68
Add ringing tones for addresses ... 68
Copy contacts ... 69
SChip directory and other SmartChip services ... 69
Make calls ... 70
Voice calls ... 70
Make a conference call ... 70
1-touch dial a phone number ... 71
Voice dialing ... 71
Answer or decline a call ... 72
Call waiting ... 72
Options during a voice call ... 72
Log ... 73
Call time ... 73
Data counter ... 73
Monitor all communication events ... 74
MEdia Net ... 75
Access point ... 75
Bookmarks view ... 75
Connection security ... 76
Browse ... 76
Download and purchase items ... 77
End a connection ... 77
Empty the cache ... 77
Settings ... 77
Calendar ... 79
Create calendar entries ... 79
Calendar views ... 79
Manage your calendar entries ... 80
Games&Apps ... 81
Games ... 81
MobiTV ... 81

Instant Messaging & Email 82
Instant messaging 82
Receive IM settings 82
Connect to an IM server 82
Modify your IM settings 82
Search for IM groups and users 83
Join IM groups 83
Instant messaging 83
View and start conversations 84
IM contacts 84
IM server settings 84
Mobile email 85

Tools 86
Web 86
Calculator 86
Converter 86
Set base currency and exchange rates 87
Notes 87
Recorder 87
RealPlayer 87
Play video clips 88
Shortcuts during play 88
Stream content over the air 88
Receive RealPlayer settings 88
Flash Player 89
Play flash files 89
Organize flash files 90
Radio 90
Listen to the radio 90
Saved stations 91
Settings 91

Personalize your device 92
Profiles—set tones 92
3-D tones 93
Offline profile 93
Change the look of your device 94
Active idle mode 94

Settings 95
Config 95
Phone 95
Call config. 97
Connection 97
Date and time 99
Security 99
Call forwarding 102
Call restriction 102
Network 102
Enhancement 103
Voice commands 103
Voice aid 104
Application manager 104
Install applications and software 105
Remove applications and software 106
Settings 106

Speech....106
Bluetooth connectivity....107
Settings....107
Send data using Bluetooth connectivity....108
Pair devices....109
Receive data using Bluetooth connectivity....110
Switch off Bluetooth connectivity....110
Infrared connection....110
Data cable....111
PC connections....111
Connection manager....112
Data connections....112
Synchronization....112
Synchronize data....112
Device manager....112
Server profile settings....113
Troubleshooting: Q&A....115
Enhancements....118
Battery information....120
Charging and discharging....120
Nokia battery authentication guidelines....121
Care and maintenance....123
Additional safety information....124
Index....128

For your safety

Read these simple guidelines. Not following them may be dangerous or illegal. Read the complete user guide for further information.

SWITCH ON SAFELY Do not switch the device on when wireless phone use is prohibited or when it may cause interference or danger.

ROAD SAFETY COMES FIRST Obey all local laws. Always keep your hands free to operate the vehicle while driving. Your first consideration while driving should be road safety.

INTERFERENCE All wireless devices may be susceptible to interference, which could affect performance.

SWITCH OFF IN HOSPITALS Follow any restrictions. Switch the device off near medical equipment.

SWITCH OFF IN AIRCRAFT Follow any restrictions. Wireless devices can cause interference in aircraft.

SWITCH OFF WHEN REFUELING Do not use the device at a refueling point. Do not use near fuel or chemicals.

SWITCH OFF NEAR BLASTING Follow any restrictions. Do not use the device where blasting is in progress.

USE SENSIBLY Use only in the normal position as explained in the product documentation. Do not touch the antenna unnecessarily.

QUALIFIED SERVICE Only qualified personnel may install or repair this product.

ENHANCEMENTS AND BATTERIES Use only approved enhancements and batteries. Do not connect incompatible products.

WATER-RESISTANCE Your device is not water-resistant. Keep it dry.

BACK-UP COPIES Remember to make back-up copies or keep a written record of all important information.

CONNECTING TO OTHER DEVICES When connecting to any other device, read its user guide for detailed safety instructions. Do not connect incompatible products.

EMERGENCY CALLS Ensure the phone function of the device is switched on and in service. Press the end key as many times as needed to clear the display and return to the start screen. Enter the emergency number, then press the call key. Give your location. Do not end the call until given permission to do so.

About your device

The wireless device described in this guide is approved for use on the EGSM 850, 900, 1800, 1900, and UMTS 850 and 1900 networks. Contact your service provider for more information about networks.

When taking and using images, video clips or other features, obey all laws, and respect local customs as well as privacy and legitimate rights of others, including copyright.

Warning: To use any features in this device, other than the alarm clock, the device must be switched on. Do not switch the device on when wireless device use may cause interference or danger.

The **Quickoffice** applications in your device support common features of Microsoft Word, PowerPoint, and Excel (Microsoft Office 2000, XP, and 2003). Not all file formats can be viewed or modified.

Network services

To use the phone you must have service from a wireless service provider. Many of the features in this device depend on features in the wireless network to function. These network services may not be available on all networks or you may have to make specific arrangements with your service provider before you can utilize network services. Your service provider may need to give you additional instructions for their use and explain what charges will apply. Some networks may have limitations that affect how you can use network services. For instance, some networks may not support all language-dependent characters and services.

Your service provider may have requested that certain features be disabled or not activated in your device. If so, they will not appear on your device menu. Your device may also have been specially configured. This configuration may include changes in menu names, menu order, and icons. Contact your service provider for more information.

This device supports WAP 2.0 protocols (HTTP and SSL) that run on TCP/IP protocols. Some features of this device, such as MMS, browsing, e-mail, and content downloading using the browser or over MMS, require network support for these technologies.

For availability and information on using SmartChip services, contact your service provider.

Enhancements, batteries, and chargers

Always switch the device off, and disconnect the charger before removing the battery.

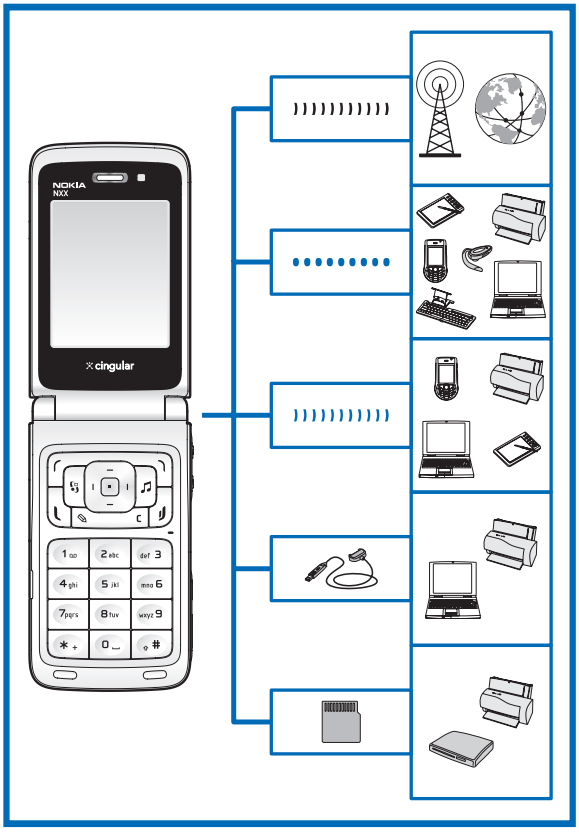
Check the model number of any charger before use with this device. This device is intended for use when supplied with power from an AC-3, AC-4, or AC-5 charger and from an AC-1, ACP-8, ACP-12, LCH-9, or LCH-12 charger when used with the CA-44 charger adapter.

The battery intended for use with this device is BL-5BT.

Warning: Use only batteries, chargers, and enhancements approved by Nokia for use with this particular model. The use of any other types may invalidate any approval or warranty and may be dangerous.

For availability of approved enhancements, please check with your dealer. When you disconnect the power cord of any enhancement, grasp and pull the plug, not the cord.

Get connected



Use your device on the 2G and 3G networks. See "Your Nokia device," p.13

Use Bluetooth technology, for example, to transfer files and connect to compatible enhancements. See "Bluetooth connectivity," p. 107.

Use infrared to transfer and synchronize data between compatible devices, for example. See "Infrared connection," p. 110.

Use Nokia Connectivity Cable CA-53 to connect to compatible devices, such as printers and PCs. See "Data cable," p. 111.

Use a microSD card to transfer data or back up information, for example. See "Memory card tool," p. 23.

Welcome

Congratulations on your purchase of this Nokia device. Your device provides many functions that are practical for daily use, such as a calendar, a clock, an alarm clock, a radio, and a built-in camera. Use the camera for recording video clips and taking pictures that you can, for example, attach as a wallpaper in the idle mode. See "Camera," p. 28.

You can customize your device with ringing tones and themes. See "Personalize your device," p. 92.

Other features on your device are:

- Multimedia messaging. See "Messaging," p. 54.
- Instant messages. See "Games," p. 81.
- Music player. See "Cingular Music," p. 36.
- Wireless Bluetooth technology. See "Bluetooth connectivity," p. 107.
- Java™ applications. See "Application manager," p. 104.
- Web browser. See "MEdia Net," p. 75.

Accessibility solutions

Nokia is committed to making mobile devices easy to use for all individuals, including those with disabilities. For more information, visit **www.nokiaaccessibility.com**.

Find information about your device

Information about your device is provided on the device label, which is on the back of the device under the battery. The label contains the model and serial numbers, as well as the following:

- Device type
- Device model
- International mobile equipment identity (IMEI)
- FCC ID number

Contact your service provider

In many cases, the service provider will make available descriptions of its services and instructions for using features, such as the following:

- Voice mail and voice privacy
- Call waiting, call forwarding, and caller ID
- Messaging
- News and information services
- Selected Internet services

Service providers may differ in their support of features.

Your Nokia device

Keys and parts (fold open)

Light sensor (1)

Display screen (2)

4-way scroll key (3)

Right selection key ( ) (4)

Music key ( ) (5)

End key ( ) (6)

Clear key ( ) (7)

Keypad (8)

Edit key ( ) (9)

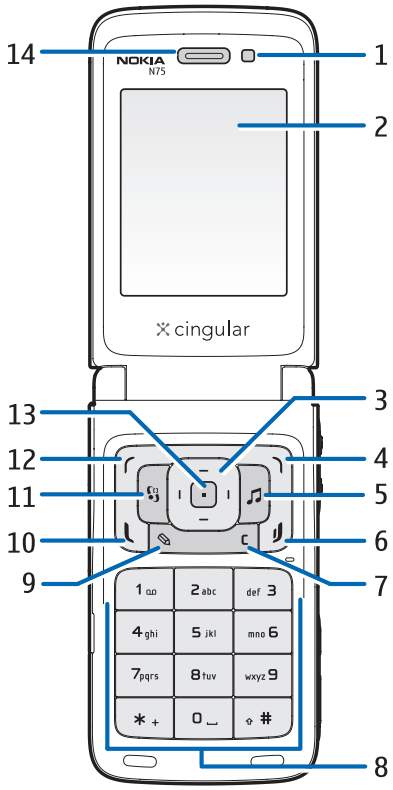
Call key ( ) (10)

Menu key ( ) (11)

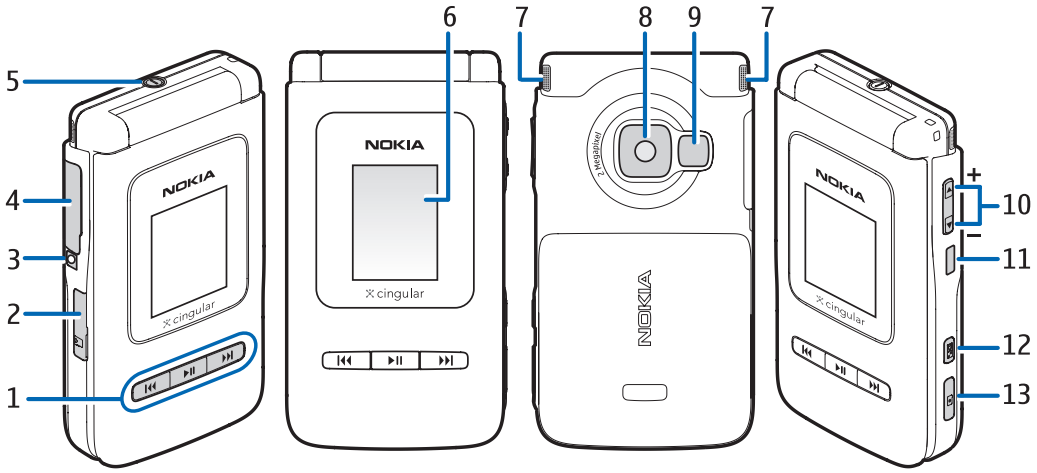
Left selection key ( ) (12)

Center selection key ( ) (13)

Earpiece (14)



Keys and parts (fold closed, front, back and sides)



Player keys (1):
backward ⏮ play/pause ⏯ forward ⏭

Memory card slot (micro SD card) (2)

Charger connector (3)

Pop-Port™ connector (4)

Power button (5)

Cover display (6)

Stereo speakers (7)

Camera (8)

Camera flash (9)

Zoom/Volume key (10)

Infrared (IR) window (11)

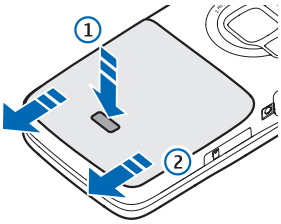
Mode key ( ) (12)

Capture key ( ) (13)

Insert the SmartChip card and battery

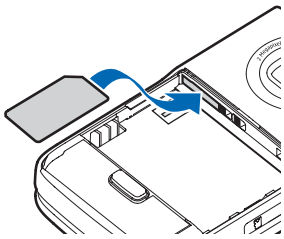
Keep all SmartChip cards out of the reach of small children.

Always switch the device off, and disconnect the charger before removing the battery.

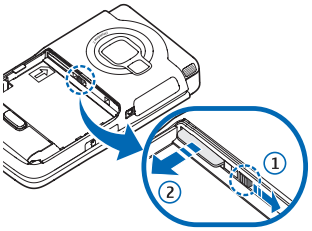


1 With the back of the device facing you, press the release button (1), and slide off the cover (2).

2 Slide the SmartChip card into the holder until it clicks. Make sure that the bevelled corner on the card is facing the left side of the device and that the contact area on the card is facing down.

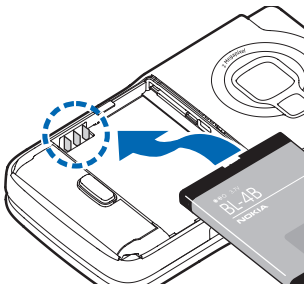


To remove the SmartChip card, slide the lock to the right and pull out the SmartChip card.



3 Insert the battery.

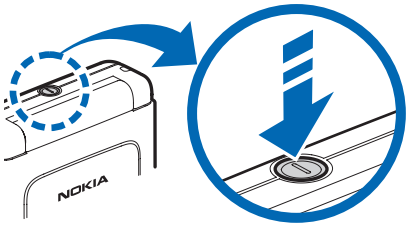
4 Replace the back cover.



Switch the device on

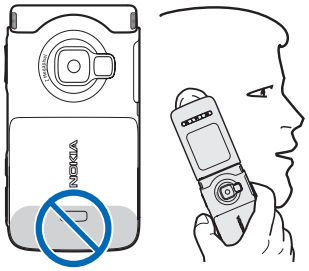
1 Press and hold the power button (⏻).

2 If the device asks for a UPIN code or lock code, enter it, and press ⌐. The factory setting for the lock code is 12345.



Your device has one internal antenna.

Note: As with any other radio transmitting device, do not touch the antenna unnecessarily when the device is switched on. Contact with the antenna affects call quality and may cause the device to operate at a higher power level than otherwise needed. Not touching the antenna area during a phone call optimizes the antenna performance and the battery life.



- When you switch on the device for the first time, it may ask you to set the following information: **Select current city:**, **Time:**, and **Date:**. Use the 4-way scroll key, the center selection key, and the number keys. To find the city, enter the first letters of the city name. The city you select also defines the time zone for the clock in your device.
- Your Nokia N75 normally has MMS, GPRS, streaming, and mobile Internet settings automatically configured in the device.
- Press to open the main menu.

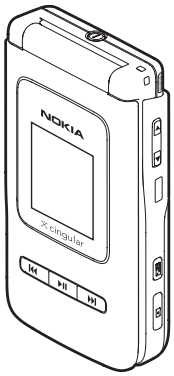
Your service provider may have requested that certain features be disabled or not activated in your device. If so, they will not appear on your device menu. Your device may also have been specially configured. This configuration may include changes in menu names, menu order, and icons. Contact your service provider for more information.

Modes

Your device has two operating modes: Fold closed with access to particular features including **Music player**, **Radio** and **Camera**; Fold open for traditional mobile phone use and access to all the features on your device.

Fold closed mode

When the fold is closed, the main display deactivates and the cover display lights up. Ongoing calls end, unless the loudspeakers are in use or a compatible headset is connected. To change the settings, see "Cover display," p. 96. If you are using **Music player** or **Radio** to listen to an audio clip, it continues to play.



With the fold closed, pressing the mode key switches between 3 views:

- 'Basic' idle: displays clock and status information, signal strength and battery indicators. The following softkeys are also available:

  Left softkey (⏮) selects **Music player**

  Middle softkey (⏯) selects Play/Pause

  Right softkey (⏭) selects **Now playing**
- 'Today' idle: dispalys calendar information
- 'Picture' idle displays a picture of your choice

If you are using **Music player** to listen to an audio clip, volume bars and track information are displayed. To play, pause, or play the next or previous audio clip, use the player keys on the cover. To adjust the volume, use the volume key on the right-hand side of the device.

If you are using **Radio** with the fold closed, volume bars and station information are displayed. To turn the radio on or off, and select the next or previous saved station, use the player keys on the cover. To adjust the volume, use the volume key on the right-hand side of the device.

If you are using **Camera** with the fold closed, hold the device in landscape. The outer display becomes the viewfinder. To activate the camera, press and hold the capture key. To zoom in or out, use the volume key. To take a picture, press the capture key. To switch between **Image mode** and **Video mode**, press the mode key.

To make lighting and color adjustments before taking a picture, press the middle cover key to activate the toolbar, and use the volume key to scroll through the active toolbar. See "Adjust color and lighting," p. 31 and "Scenes," p. 32.

Calendar and clock alarms and notes for missed calls and new received messages are also displayed. To view received text messages and the text and image objects of received multimedia messages, open the fold, and view the message on the main display.

Incoming calls are indicated by their normal tone and a note. In the cover display settings, select **Answer if fold opened** to answer a call when you open the fold. If a compatible headset is connected to the device, press the answer key on the headset. To change the settings, see "Cover display," p. 96.

To make calls or use the menu, open the fold.

Fold open mode

When you open the fold, the main display lights up, the keypad is available, and you can access the menu. To access the menu, press .

The fold opens about 170 degrees, as shown in the picture. Do not force the fold to open more.

Essential indicators

The device is being used in a GSM network.

The device is being used in a UMTS network.

You have one or more unread messages in the **Inbox** folder in **Messag.**.

You have received new e-mail in the remote mailbox.

There are messages waiting to be sent in the **Outbox** folder.

You have missed calls.

Shown if **Ringing type** is set to **Silent** and **Message alert tone**, **IM alert tone**, and **E-mail alert tone** are set to Off.

The device keypad is locked.

A clock alarm is active.

The second phone line is being used (network service).

All calls to the device are forwarded to another number. If you have two phone lines (network service), the indicator for the first line is 1 and for the second 2.

A compatible headset is connected to the device.

The connection to a compatible headset with Bluetooth connectivity has been lost.

A compatible loopset is connected to the device.

A compatible **TTY** is connected to the device.

A data call is active.

A GPRS or EDGE packet data connection is available.

A GPRS or EDGE packet data connection is active.

A GPRS or EDGE packet data connection is on hold.

A UMTS packet data connection is available.

A UMTS packet data connection is active.

A UMTS packet data connection is on hold.

Bluetooth connectivity is on.

Data is being transmitted using Bluetooth connectivity.

A USB connection is active.

An infrared connection is active. If the indicator is blinking, the device is trying to connect to the other device or the connection has been lost.

Useful shortcuts

Use shortcuts to quickly get the most from your device. Refer to the relevant sections in this user guide for further details of the functions.

Idle mode

To switch between applications that are open, press and hold , and select an application. If memory is low, the device may close some applications. The device saves any unsaved data before closing an application.

Leaving applications running in the background increases the demand on battery power and reduces the battery life.

To start the camera, press and hold the capture key for more than 1 second.

To call your voice mailbox (network service), press and hold 1.

To open **Cingular Music** folder, press or press and hold to start **Music player**. See "Music key," p. 24.

To change the profile, press , and select a profile.

To switch between **Normal** and **Silent** profiles, press and hold #. If you have two phone lines (network service), this action switches between the two lines.

To open the last dialed numbers list, press .

To use voice commands, press and hold .

To start a connection to **Services** (network service), press and hold 0 or press . See "MEdia Net," p. 75.

For further shortcuts available in the idle mode, see "Active idle mode," p. 94.

Edit text and lists

To mark an item in a list, scroll to it, and press and at the same time.

To mark multiple items in a list, press and hold while you press or . To end the selection, release , then release .

To copy and paste text: To select letters and words, press and hold . At the same time, press or to highlight text. To copy the text to clipboard, while still holding , select **Copy**. To insert the text into a document, press and hold , and select **Paste**.

Keypad lock (keyguard)

Use the keypad lock to prevent accidental keypresses.

To lock, in the idle mode, press , then *. When the keys are locked, is shown on the display.

To unlock with the fold open, press , then *.

To lock the keypad with the fold open, press the power key, and select **Lock keypad**.

To lock the keypad with the fold closed, press ⏮, then ⏭, or press the power key twice to highlight **Lock keypad**, and press and hold the power key to select it.

To unlock with the fold closed, press ⏭, then ⏮.

When the keypad lock is on, calls may be possible to the official emergency number programmed into your device.

To turn on the display light when the keypad lock is on, press the power key.

Volume and loudspeaker control

To increase or decrease the volume level when you have an active call or are listening to audio, press and hold ◀ ▶ on the right side of the device.

The built-in loudspeakers allow you to speak and listen from a short distance without having to hold the device to your ear. Audio applications use the loudspeakers by default.

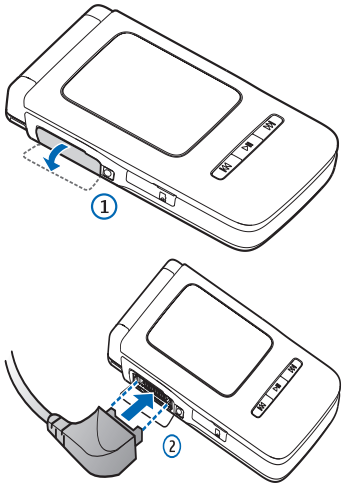
Warning: Listen to music at a moderate level. Continuous exposure to high volume may damage your hearing. Do not hold the device near your ear when the loudspeaker is in use, because the volume may be extremely loud.

To locate the loudspeakers, see "Keys and parts" in the Quick start guide.

To use the loudspeakers during a call, select **Options** > **Activate loudsp.**.

To turn off the loudspeakers, select **Options** > **Activate handset**.

Headset



Connect a compatible headset to the Pop-Port™ connector of your device.

Ensure the headset connector is 'upside down' before connecting to your device. Do not use excessive force.

Warning: When you use the headset, your ability to hear outside sounds may be affected. Do not use the headset where it can endanger your safety.

Clock

Press , and select **Tools** > **Clock**. To set a new alarm, select **Options** > **Set alarm**. When an alarm is active, is shown.

To turn off the alarm, select **Stop**, or to stop the alarm for 5 minutes, select **Snooze**.

If the alarm time is reached while the device is switched off, the device switches itself on and starts sounding the alarm tone. If you select **Stop**, the device asks whether you want to activate the device for calls. Select **No** to switch off the device or **Yes** to make and receive calls. Do not select **Yes** when using a wireless phone may cause interference or danger.

To cancel an alarm, select **Clock** > **Options** > **Remove alarm**.

Clock settings

To change the clock settings, select **Clock** > **Options** > **Settings**.

To change the time or date, select **Time** or **Date**.

To change the clock shown in the idle mode, select **Clock type** > **Analog** or **Digital**.

To allow the mobile phone network to update the time, date, and time zone information to your device (network service), select **Network operator time** > **Auto-update**.

To change the alarm tone, select **Clock alarm tone**.

World clock

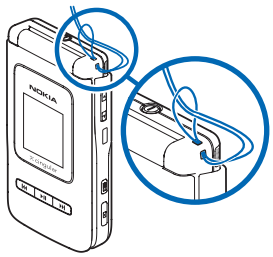
To open the world clock view, select **Clock**, and press . In the world clock view, you can view the time in different cities.

To add cities to the list, select **Options** > **Add city**. Enter the first letters of the city name. The search field appears automatically, and the matching cities are displayed. Select the required city. You can add a maximum of 15 cities to the list.

To set your current city, scroll to a city, and select **Options** > **My current city**. The city is displayed in the clock main view, and the time in your device is changed according to the city selected. Ensure that the time is correct and matches your time zone.

Wrist strap

Thread a strap as shown, and tighten it.



Settings

You may have MMS, GPRS, streaming, and Mobile Internet settings from your service providers already installed in your device, or you may receive the settings from the service providers as a special text message.

Memory card

You can use a compatible microSD card as extra storage space to save the memory on your device. You can also back up information from your device to the memory card.

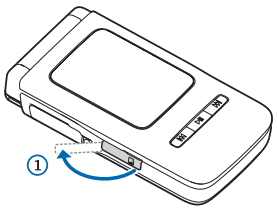
Use only compatible microSD cards with this device. Other memory cards, such as Reduced Size MultiMediaCards, do not fit in the memory card slot and are not compatible with this device. Using an incompatible memory card may damage the memory card as well as the device, and data stored on the incompatible card may be corrupted.

Use only microSD cards approved by Nokia for use with this device. Nokia utilizes the approved industry standards for memory cards but not all other brands may function correctly or be fully compatible with this device.

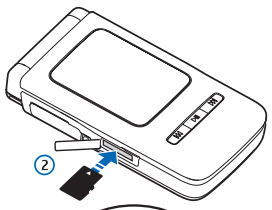
Keep all memory cards out of the reach of small children.

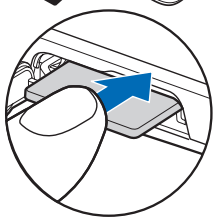
Insert the memory card

1 To insert the memory card, place your finger in the recess of the door of the memory card slot, and open the door.

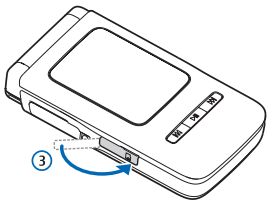


2 Insert the memory card in the slot. Make sure that the contact area on the card is facing down and towards the slot with the bevelled corner facing the base of the device. Push the card in. You can hear a click when the card locks into place.





3 Close the door. You cannot use the memory card if the door is open.



Eject the memory card

1 Before you eject the card, press the power key, and select **Remove mem. card**. All applications are closed.

2 When **Remove memory card and press 'OK'** is displayed, open the door of the memory card slot.

3 Press the memory card to release it from the slot.

4 Remove the memory card.

5 If the device is switched on, select **OK**.

Important: Do not remove the memory card in the middle of an operation when the card is being accessed. Removing the card in the middle of an operation may damage the memory card as well as the device, and data stored on the card may be corrupted.

Memory card tool

Press , and select **Settings** > **Memory**. You can use a compatible memory card as extra storage space and to back up the information in the device memory.

To back up information from the device memory to a compatible memory card, select **Options** > **Backup phone mem.**

To restore information from the compatible memory card to the device memory, select **Options** > **Restore from card**.

Format a memory card

When a memory card is reformatted, all data on the card is permanently lost.

Some memory cards are supplied preformatted, and others require formatting. Consult your retailer to find out if you must format the memory card before you can use it.

To format a memory card, select **Options** > **Format mem. card**. Select **Yes** to confirm.

File manager

Many features of the device, such as address book, messages, images, videos, ringing tones, calendar notes, documents, and downloaded applications, use memory to store data. The free memory available depends on how much data is already saved in the device memory.

You can use a compatible memory card as extra storage space. Memory cards are rewritable, so you can delete old information and save new data on a memory card.

To browse files and folders in the device memory or on a memory card (if inserted), press , and select **Settings** > **File mgr.**. The device memory view ( ) opens. Press to open the memory card view ( ), if available.

To move or copy files to a folder, press and at the same time to mark a file, and select **Options** > **Move to folder** or **Copy to folder**.

To find a file, select **Options** > **Find** and the memory from which to search, and enter a search text that matches the file name.

View memory consumption

To view what types of data you have in the device and how much memory the different data types consume, select **Options** > **Check memory**. The amount of available free memory is shown in **Free memory**.

Memory low—free memory

The device notifies you if the device or the memory card memory is low.

To free device memory, transfer data to a compatible memory card, if you have one. Mark files to be moved, and select **Move to folder** > **Memory card** and a folder. You can also transfer media files to a compatible PC using Nokia PC Suite.

Tip! You can use Nokia Phone Browser available in Nokia PC Suite to view the different memories in your device and transfer data.

To remove data to free memory, use **File mgr.**, or go to the respective application. For example, you can remove the following:

- Messages from **Inbox**, **Drafts**, and **Sent** folders in **Messag.**
- Retrieved e-mail messages from the device memory
- Saved Web pages
- Saved images, videos, or audio files
- Contact information
- Calendar notes
- Downloaded applications. See also "Application manager," p. 104.
- Any other data that you no longer need

You can also shrink images within **My Stuff**, so that they use less memory. See "Shrink," p. 48.

Music key

With the music key, you have quick access to **Music player** and the **Cingular Music** folder containing multimedia content and predefined applications.

To open **Music player**, press and hold . To open the **Cingular Music** folder, press . To open the desired application, press .

Mode key

The mode key is a multi-function key, allowing you to switch easily between selected applications.

Press and hold to open the latest taken photo in **My Stuff**.

In **Camera**, press ▶■ to switch between **Image mode** and **Video mode**.

In **My Stuff**, press ▶■ to switch between Tile and Thumbnail views.

With the fold closed, in **Music player**, press ▶■ to switch between **Now playing** and **Music library** views.

With the fold closed, in **Radio**, press ▶■ to switch between Channel and Channel list views.

With the fold closed and the device in idle mode, press ▶■ to switch between 'basic' idle (clock and softkeys available); 'today' idle (calendar view); and 'picture' idle (where the user can select a picture to display).

Transfer content from another device

You can transfer content, such as an address book (or contacts), between compatible Nokia devices using Bluetooth connectivity or infrared. The type of content that can be transferred depends on the Nokia device you are trying to transfer content to. If the other device supports synchronization, you can also synchronize data between the other device and your Nokia device.

Copyright protections may prevent some images, music (including ringing tones), and other content from being copied, modified, transferred or forwarded.

You can insert your SmartChip card in the other device. When your Nokia device is switched on without a SmartChip card, the offline profile is automatically activated.

Transfer content (to your device)

1. To use the application for the first time, press ☞ on your Nokia device, and select **Settings** > **Transfer**. If you have used the application before and want to start a new transfer, select **Transfer data**.

   In the information view, select **Continue**.

2. Select whether you want to use Bluetooth connectivity or infrared to transfer the data. Both devices must support the selected connection type.

3. If you select Bluetooth connectivity:

   To have your Nokia device search for devices with Bluetooth connectivity, select **Continue**. Select your other device from the list. You are asked to enter a code on your Nokia device. Enter a code (1-16 digits), and select **OK**. Enter the same code on the other device, and select **OK**. The devices are now paired. See "Pair devices," p. 109.

   For some phone models, the **Transfer** application is sent to the other device as a message. To install **Transfer** on the other device, open the message, and follow the instructions on the display.

   If you select infrared, connect the two devices. See "Infrared connection," p. 110.

4 From your Nokia device, select the content you want to copy from the other device.

Content is copied from the memory of the other device to the corresponding location in your Nokia device. Copying time depends on the amount of data to be transferred. You can cancel copying and continue later.

If the other device supports synchronization, you can keep the data up-to-date in both devices.

Tip! Synchronizing data means that data is made the same in each device, including deleted data. Use **Transfer data** to copy data from one device to another.

To start a synchronization with a compatible Nokia device, select **Phones**, scroll to the device, and select **Options** > **Synchronize**. Follow the instructions on the display.

To view the log of a previous transfer, select **Transfer log**.

Support

Nokia support and contact information

Check www.nokia.com/support or your local Nokia Web site for the latest version of this guide, additional information, downloads, and services related to your Nokia device.

On the Web site, you can find information on the use of Nokia products and services. If you need to contact customer service, check the list of local Nokia contact centers at www.nokia.com/customerservice.

For maintenance services, check your nearest Nokia service center at www.nokia.com/repair.

Help

Your device has context-sensitive help. When an application is open, select **Options** > **Help** to access help for the current view.

To switch between help and the application that is open in the background, press and hold .

To open help from the main menu, select **Tools** > **Help**. Select the desired application to view its help topics. To search for help topics using keywords, select **Options** > **Search by keyword**

When you are reading a help topic, scroll left or right to view other related topics.

Tutorial

The tutorial gives you information about some of the features of your device.

To access the tutorial in the menu, press , and select **Tools** > **Tutorial** and the section you want to view.

Camera

Your Nokia device has a high resolution camera on the back of the device that you can use to take still pictures and record videos.

This device supports an image resolution of 1600x1200 pixels (2 Mp). The image resolution in these materials may appear different.

The images and video clips are automatically saved in the **Imgs. & video** folder in **My Stuff**. The camera produces .jpeg images. Video clips are recorded in the .mp4 file format, or in the 3GPP file format with the .3gp file extension (**Sharing** quality). See "Video settings," p. 34.

You can send images and video clips in a multimedia message, as an e-mail attachment, or by using Bluetooth connectivity.

Take pictures

When taking a picture, note the following:

- Use both hands to keep the camera still.
- The quality of a digitally zoomed picture is lower than that of a nonzoomed picture.
- The camera switches to battery saving mode if there are no keypresses for a short while. To continue taking pictures, press .

To take a picture using the cover display as the viewfinder (fold closed), do the following:

1. If the camera is in **Video mode**, press to switch to **Image mode**.
2. Press . Do not move the device before the image is saved.

To take a picture using the main display as the viewfinder, do the following:

1. Open the fold. If the camera is in **Video mode**, press to select the **Image mode** from the active toolbar.
2. Press . Do not move the device before the image is saved.

To zoom in or out, in the main display, press or . In the cover display, press .

To make lighting and color adjustments before taking a picture, in the main display, press or to scroll to items on the active toolbar. In the cover display, press ▶II to activate the toolbar, and to scroll to items. See "Adjust color and lighting," p. 31 and "Scenes," p. 32.

Saving the captured image may take longer if you change the zoom, lighting, or color settings.

To free memory before taking a picture, in the main display, select **Options** > **Go to free memory** (only available if you have backed up images or video clips).

Still image camera indicators

The still image camera viewfinder (main display) displays the following:

- The current capture mode indicator (1)
- The active toolbar (2), which you can scroll through before capturing the image to change **Scene modes**, adjust the flash setting, activate the self-timer or sequence mode, or adjust **Exposure compensation**, **White balance** or **Color tone** (the active toolbar is not displayed during image capture). See "Active toolbar," p. 29.
- The image resolution indicator (3) indicates whether the quality of the image is **Print 2M – Large** (1600x1200 resolution), **Print 1M – Medium** (1152x864 resolution) or **Multimedia message 0.3M – Small** (640x480 resolution).

- The image counter (4) displays the estimated number of images you can capture using the current image quality setting and memory in use (the counter is not displayed during focus and image capture).
- The device memory ( ) and memory card ( ) indicators (5) show where images are saved.

To display all viewfinder indicators, select **Options** > **Show icons**. To display only the camera status indicators, select **Hide icons**.

Active toolbar

You can use the active toolbar as a shortcut to select items and adjust different settings before taking a picture or recording a video.

If you are using the main display as a viewfinder, scroll to the item you want, and press .

If you are using the cover display as a viewfinder, press ▶II to display the active toolbar, use to scroll to the item you want, and press ▶II.

The available options vary depending on the capture mode. You can also define whether the active toolbar is always visible on the display or activated by a key press.

If you want the active toolbar to be visible on the main display before and after taking a picture or recording a video, select **Options** > **Show icons**.