EXHIBIT 1009

Specification Volume 1

SPECIFICATION OF THE *BLUETOOTH* SYSTEM

Experience More



Architecture & Terminology Overview

Covered Core Package version:
4.0

Bluetooth®

Revision History

The Revision History is shown in the [Vol 0] Part C Appendix on page 55.

Contributors

The persons who contributed to this specification are listed in the [Vol 0] Part C Appendix on page 55.

Web Site

This specification can also be found on the official Bluetooth web site:
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 applicable 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.

Copyright © 1999 - 2010

Ericsson AB,
Lenovo,
Intel Corporation,
Microsoft Corporation,
Motorola, Inc.,
Nokia Corporation,
Toshiba Corporation

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

TABLE OF CONTENTS

Part A
ARCHITECTURE

1 General Description 17
1.1 Overview of BR/EDR Operation 18
1.2 Overview Of Bluetooth Low Energy Operation 20
1.3 Overview Of AMP Operation 22
1.4 Nomenclature 24

2 Core System Architecture 30
2.1 Core Architectural Blocks 34
2.1.1 Host Architectural Blocks 34
2.1.1.1 Channel Manager 34
2.1.1.2 L2CAP Resource Manager 34
2.1.1.3 Security Manager Protocol 34
2.1.1.4 Attribute Protocol 35
2.1.1.5 AMP Manager Protocol 35
2.1.1.6 Generic Attribute Profile 35
2.1.1.7 Generic Access Profile 35
2.1.2 BR/EDR/LE Controller Architectural Blocks 35
2.1.2.1 Device Manager 36
2.1.2.2 Link Manager 36
2.1.2.3 Baseband Resource Manager 36
2.1.2.4 Link Controller 37
2.1.2.5 PHY 37
2.1.3 AMP Controller architectural blocks 37
2.1.3.1 AMP HCI 37
2.1.3.2 AMP PAL 38
2.1.3.3 AMP MAC 38
2.1.3.4 AMP PHY 38

3 Data Transport Architecture 39
3.1 Core Traffic Bearers 40
3.1.1 Framed Data Traffic 41
3.1.2 Unframed Data Traffic 42
3.1.3 Reliability of traffic bearers 43
3.1.3.1 BR/EDR Reliability 43
3.1.3.2 LE reliability 44
3.1.3.3 AMP Reliability 45
3.2 Transport Architecture Entities 45
3.2.1 BR/EDR Generic Packet Structure 46
3.2.2 LE Generic Packet Structure 47
3.3 Physical Channels 49

3.3.1 BR/EDR Physical Channels 49
3.3.1.1 Basic Piconet Channel 50
3.3.1.2 Adapted Piconet Channel 51
3.3.1.3 Inquiry scan channel 52
3.3.1.4 Page Scan Channel 53
3.3.2 LE Physical Channels 54
3.3.2.1 LE piconet channel 55
3.3.2.2 Advertisement broadcast channel 56
3.3.3 AMP physical channel 57
3.3.3.1 Overview 57
3.3.3.2 Characteristics 57
3.3.3.3 Topology 57
3.3.3.4 Supported Layers 58
3.4 Physical Links 58
3.4.1 BR/EDR Links Supported By The Basic And Adapted Piconet Physical Channel58
3.4.1.1 Active Physical Link 58
3.4.1.2 Parked Physical Link 59
3.4.2 BR/EDR Links Supported by the Scanning Physical Channels60
3.4.3 LE Links Supported by the LE Physical Channels 60
3.4.3.1 Active physical link 60
3.4.3.2 Advertising physical link 60
3.4.4 Links Supported by the AMP Physical Channels 61
3.5 Logical Links and Logical Transports 61
3.5.1 Casting 62
3.5.2 Scheduling and Acknowledgement Scheme 63
3.5.3 Class of Data 63
3.5.4 Logical Transports 64
3.5.4.1 BR/EDR Asynchronous Connection-oriented (ACL) 64
3.5.4.2 BR/EDR Synchronous Connection-Oriented (SCO) 65
3.5.4.3 BR/EDR Extended Synchronous Connection-Oriented (eSCO) 65
3.5.4.4 BR/EDR Active Slave Broadcast (ASB) 66
3.5.4.5 BR/EDR Parked Slave Broadcast (PSB) 67
3.5.4.6 LE Asynchronous Connection (LE ACL) 68
3.5.4.7 LE Advertising Broadcast (ADVB) 68
3.5.5 Logical Links 69
3.5.5.1 BR/EDR Logical Links 69
3.5.5.2 LE Logical Links 70
3.5.5.3 AMP Logical Links 71
3.6 L2CAP Channels 72

4 Communication Topology and Operation 73
4.1 Piconet Topology 73
4.1.1 BR/EDR Topology 73
4.1.2 LE Topology 75
4.2 Operational Procedures and Modes 76
4.2.1 BR/EDR Procedures 76
4.2.1.1 Inquiry (Discovering) Procedure 76
4.2.1.2 Paging (Connecting) Procedure 77
4.2.1.3 Connected Mode 77
4.2.1.4 Hold Mode 78
4.2.1.5 Sniff Mode 78
4.2.1.6 Parked State 79
4.2.1.7 Role switch procedure 79
4.2.1.8 Enhanced Data Rate 80
4.2.2 LE Procedures 80
4.2.2.1 Device Filtering Procedure 80
4.2.2.2 Advertising Procedure 81
4.2.2.3 Scanning Procedure 81
4.2.2.4 Discovering Procedure 82
4.2.2.5 Connecting Procedure 82
4.2.2.6 Connected Mode 83
4.2.3 AMP Procedures 83
4.2.3.1 AMP Discovery Procedures 83
4.2.3.2 Physical Link Creation Procedure 84
4.2.3.3 Logical Link Creation Procedure 84
5 Security Overview 85
5.1 BR/EDR Secure Simple Pairing 85
5.1.1 Security Goals 85
5.1.2 Passive Eavesdropping Protection 85
5.1.3 Man-In-The-Middle Protection 86
5.1.4 Association Models 87
5.1.4.1 Numeric Comparison 87
5.1.4.2 Just Works 87
5.1.4.3 Out of Band 88
5.1.4.4 Passkey Entry 88
5.1.4.5 Association Model Overview 89
5.2 LE Security 89
5.2.1 Association Models 90
5.2.2 Key Generation 90
5.2.3 Encryption 90
5.2.4 Signed Data 90
5.2.5 Privacy Feature 91
5.3 AMP Security 91

6 Bluetooth Application Architecture 93
6.1 Bluetooth Profiles 93
6.2 Generic Access Profile 93
6.3 Profile Hierarchy 94
6.4 Generic Attribute Profile 95
6.5 GATT-based Profile Hierarchy 96
6.5.1 Service 96
6.5.2 Referenced Services 97
6.5.3 Characteristic 97

Part B
ACRONYMS & ABBREVIATIONS

1 List of Acronyms and Abbreviations 101

Part C
CORE SPECIFICATION CHANGE HISTORY

1 Deprecated Features 116
2 Changes from V1.1 to V1.2 117
2.1 New Features 117
2.2 Structure Changes 117
2.3 Deprecated Features list 117
2.4 Changes in Wording 118
2.5 Nomenclature Changes 118
3 Changes from V1.2 to V2.0 + EDR 119
3.1 New Features 119
3.2 Deprecated Features 119
4 Changes from V2.0 + EDR to V2.1 + EDR 120
4.1 New features 120
4.2 Deprecated Features 120
5 Changes From V2.1 + EDR To V3.0 + HS 121
5.1 New Features 121
5.2 Deprecated Features 121
6 Changes From V3.0 + HS To v4.0 122
6.1 New Features 122
6.2 Deprecated Features 122

Part D
MIXING OF SPECIFICATION VERSIONS

1 Mixing of Specification Versions 126
1.1 Features and their Types 127

1.2 Core Specification Addendums 128

Part E
IEEE LANGUAGE

1 Use of IEEE Language 135
1.1 Shall 135
1.2 Must 136
1.3 Will 136
1.4 Should 136
1.5 May 136
1.6 Can 137

Architecture & Terminology Overview
Part A

ARCHITECTURE

CONTENTS

1 General Description 17
1.1 Overview of BR/EDR Operation 18
1.2 Overview Of Bluetooth Low Energy Operation 20
1.3 Overview Of AMP Operation 22
1.4 Nomenclature 24

2 Core System Architecture 30
2.1 Core Architectural Blocks 34
2.1.1 Host Architectural Blocks 34
2.1.1.1 Channel Manager 34
2.1.1.2 L2CAP Resource Manager 34
2.1.1.3 Security Manager Protocol 34
2.1.1.4 Attribute Protocol 35
2.1.1.5 AMP Manager Protocol 35
2.1.1.6 Generic Attribute Profile 35
2.1.1.7 Generic Access Profile 35
2.1.2 BR/EDR/LE Controller Architectural Blocks 35
2.1.2.1 Device Manager 36
2.1.2.2 Link Manager 36
2.1.2.3 Baseband Resource Manager 36
2.1.2.4 Link Controller 37
2.1.2.5 PHY 37
2.1.3 AMP Controller architectural blocks 37
2.1.3.1 AMP HCI 37
2.1.3.2 AMP PAL 38
2.1.3.3 AMP MAC 38
2.1.3.4 AMP PHY 38

3 Data Transport Architecture 39
3.1 Core Traffic Bearers 40
3.1.1 Framed Data Traffic 41
3.1.2 Unframed Data Traffic 42
3.1.3 Reliability of traffic bearers 43
3.1.3.1 BR/EDR Reliability 43
3.1.3.2 LE reliability 44
3.1.3.3 AMP Reliability 45
3.2 Transport Architecture Entities 45
3.2.1 BR/EDR Generic Packet Structure 46
3.2.2 LE Generic Packet Structure 47
3.3 Physical Channels 49
3.3.1 BR/EDR Physical Channels 49
3.3.1.1 Basic Piconet Channel 50

3.3.1.2 Adapted Piconet Channel 51
3.3.1.3 Inquiry scan channel 52
3.3.1.4 Page Scan Channel 53
3.3.2 LE Physical Channels 54
3.3.2.1 LE piconet channel 55
3.3.2.2 Advertisement broadcast channel 56
3.3.3 AMP physical channel 57
3.3.3.1 Overview 57
3.3.3.2 Characteristics 57
3.3.3.3 Topology 57
3.3.3.4 Supported Layers 58
3.4 Physical Links 58
3.4.1 BR/EDR Links Supported By The Basic And Adapted Piconet Physical Channel58
3.4.1.1 Active Physical Link 58
3.4.1.2 Parked Physical Link 59
3.4.2 BR/EDR Links Supported by the Scanning Physical Channels60
3.4.3 LE Links Supported by the LE Physical Channels 60
3.4.3.1 Active physical link 60
3.4.3.2 Advertising physical link 60
3.4.4 Links Supported by the AMP Physical Channels 61
3.5 Logical Links and Logical Transports 61
3.5.1 Casting 62
3.5.2 Scheduling and Acknowledgement Scheme 63
3.5.3 Class of Data 63
3.5.4 Logical Transports 64
3.5.4.1 BR/EDR Asynchronous Connection-oriented (ACL)64
3.5.4.2 BR/EDR Synchronous Connection-Oriented (SCO)65
3.5.4.3 BR/EDR Extended Synchronous Connection-Oriented (eSCO)65
3.5.4.4 BR/EDR Active Slave Broadcast (ASB) 66
3.5.4.5 BR/EDR Parked Slave Broadcast (PSB) 67
3.5.4.6 LE Asynchronous Connection (LE ACL) 68
3.5.4.7 LE Advertising Broadcast (ADVB) 68
3.5.5 Logical Links 69
3.5.5.1 BR/EDR Logical Links 69
3.5.5.2 LE Logical Links 70
3.5.5.3 AMP Logical Links 71
3.6 L2CAP Channels 72

4 Communication Topology and Operation 73

4.1 Piconet Topology....73
4.1.1 BR/EDR Topology....73
4.1.2 LE Topology....75
4.2 Operational Procedures and Modes....76
4.2.1 BR/EDR Procedures....76
4.2.1.1 Inquiry (Discovering) Procedure....76
4.2.1.2 Paging (Connecting) Procedure....77
4.2.1.3 Connected Mode....77
4.2.1.4 Hold Mode....78
4.2.1.5 Sniff Mode....78
4.2.1.6 Parked State....79
4.2.1.7 Role switch procedure....79
4.2.1.8 Enhanced Data Rate....80
4.2.2 LE Procedures....80
4.2.2.1 Device Filtering Procedure....80
4.2.2.2 Advertising Procedure....81
4.2.2.3 Scanning Procedure....81
4.2.2.4 Discovering Procedure....82
4.2.2.5 Connecting Procedure....82
4.2.2.6 Connected Mode....83
4.2.3 AMP Procedures....83
4.2.3.1 AMP Discovery Procedures....83
4.2.3.2 Physical Link Creation Procedure....84
4.2.3.3 Logical Link Creation Procedure....84

5 Security Overview....85

5.1 BR/EDR Secure Simple Pairing....85
5.1.1 Security Goals....85
5.1.2 Passive Eavesdropping Protection....85
5.1.3 Man-In-The-Middle Protection....86
5.1.4 Association Models....87
5.1.4.1 Numeric Comparison....87
5.1.4.2 Just Works....87
5.1.4.3 Out of Band....88
5.1.4.4 Passkey Entry....88
5.1.4.5 Association Model Overview....89
5.2 LE Security....89
5.2.1 Association Models....90
5.2.2 Key Generation....90
5.2.3 Encryption....90
5.2.4 Signed Data....90
5.2.5 Privacy Feature....91
5.3 AMP Security....91

6 Bluetooth Application Architecture 93

6.1 Bluetooth Profiles 93

6.2 Generic Access Profile 93

6.3 Profile Hierarchy 94

6.4 Generic Attribute Profile 95

6.5 GATT-based Profile Hierarchy[#E3624] 96

6.5.1 Service 96

6.5.2 Referenced Services 97

6.5.3 Characteristic 97

1 GENERAL DESCRIPTION

Bluetooth wireless technology is a short-range communications system intended to replace the cable(s) connecting portable and/or fixed electronic devices. The key features of Bluetooth wireless technology are robustness, low power consumption, and low cost. Many features of the core specification are optional, allowing product differentiation.

There are two forms of Bluetooth wireless technology systems: Basic Rate (BR) and Low Energy (LE). Both systems include device discovery, connection establishment and connection mechanisms. The Basic Rate system includes optional Enhanced Data Rate (EDR) Alternate Media Access Control (MAC) and Physical (PHY) layer extensions. The Basic Rate system offers synchronous and asynchronous connections with data rates of 721.2 kbps for Basic Rate, 2.1 Mbps for Enhanced Data Rate and high speed operation up to 24 Mbps with the 802.11 AMP. The LE system includes features designed to enable products that require lower current consumption, lower complexity and lower cost than BR/EDR. The LE system is also designed for use cases and applications with lower data rates and has lower duty cycles. Depending on the use case or application, one system including any optional parts may be more optimal than the other.

Devices implementing both systems can communicate with other devices implementing both systems as well as devices implementing either system. Some profiles and use cases will be supported by only one of the systems. Therefore, devices implementing both systems have the ability to support the most use cases.

The Bluetooth core system consists of a Host and one or more Controllers. A Host is a logical entity defined as all of the layers below the non-core profiles and above the Host Controller Interface (HCI). A Controller is a logical entity defined as all of the layers below HCI. An implementation of the Host and Controller may contain the respective parts of the HCI. Two types of Controllers are defined in this version of the Core Specification: Primary Controllers and Secondary Controllers.

An implementation of the Bluetooth Core has only one Primary Controller which may be one of the following configurations:

- BR/EDR Controller including the Radio, Baseband, Link Manager and optionally HCI.
- an LE Controller including the LE PHY, Link Layer and optionally HCI.
- a combined BR/EDR Controller portion and LE controller portion (as identified in the previous two bullets) into a single Controller. This configuration has only one Bluetooth device address shared by the combination in the combined Controller.

A Bluetooth core system may additionally have one or more Secondary Controllers described by the following configuration:

- an Alternate MAC/PHY (AMP) Controller including an 802.11 PAL (Protocol Adaptation Layer), 802.11 MAC and PHY, and optionally HCI.

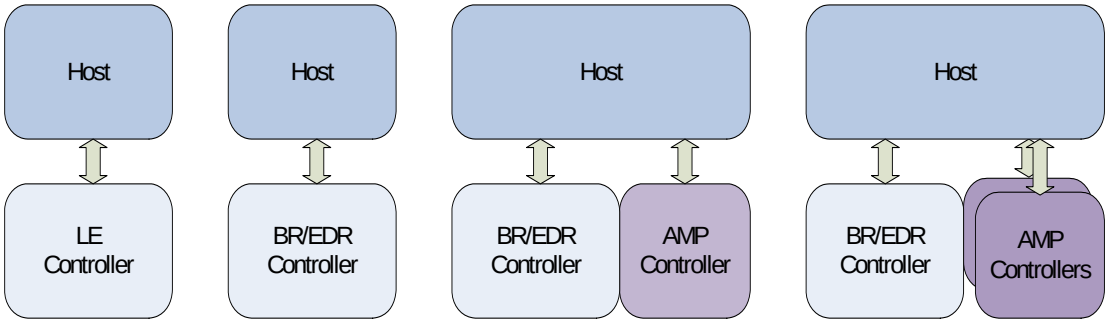


Figure 1.1: Bluetooth Host and Controller Combinations: (from left to right): LE Only Primary Controller, BR/EDR only Primary Controller, BR/EDR Primary Controller, with one AMP Secondary Controller, and BR/EDR with multiple AMP Secondary Controllers

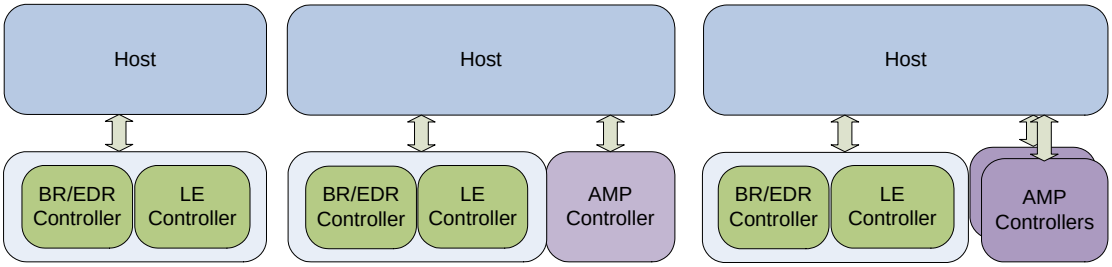


Figure 1.2: Bluetooth Host and Controller Combinations (from left to right): BR/EDR and LE Primary Controller, BR/EDR and LE Primary Controller with one AMP Secondary Controller, and BR/EDR and LE Primary Controller with multiple AMP Secondary Controllers

This volume of the specification provides an overview of the Bluetooth system architecture, communication topologies and data transport features. The text in this volume of the specification should be treated as informational and used as a background and for context-setting.

1.1 OVERVIEW OF BR/EDR OPERATION

The Basic Rate / Enhanced Data Rate (BR/EDR) radio (physical layer or PHY) operates in the unlicensed ISM band at 2.4 GHz. The system employs a frequency hop transceiver to combat interference and fading and provides many FHSS carriers. Basic Rate radio operation uses a shaped, binary frequency modulation to minimize transceiver complexity. The symbol rate is 1 Megasymbol per second (Ms/s) supporting the bit rate of 1 Megabit per second (Mb/s) or, with Enhanced Data Rate, a gross air bit rate of 2 or 3Mb/s. These modes are known as Basic Rate and Enhanced Data Rate respectively.

During typical operation a physical radio channel is shared by a group of devices that are synchronized to a common clock and frequency hopping pattern. One device provides the synchronization reference and is known as the master. All other devices synchronized to a master's clock and frequency hop-

ping pattern are known as slaves. A group of devices synchronized in this fashion form a piconet. This is the fundamental form of communication in the Bluetooth BR/EDR wireless technology.

Devices in a piconet use a specific frequency hopping pattern, which is algorithmically determined by certain fields in the Bluetooth address and clock of the master. The basic hopping pattern is a pseudo-random ordering of the 79 frequencies, separated by 1 MHz, in the ISM band. The hopping pattern can be adapted to exclude a portion of the frequencies that are used by interfering devices. The adaptive hopping technique improves Bluetooth co-existence with static (non-hopping) ISM systems when they are co-located.

The physical channel is sub-divided into time units known as slots. Data is transmitted between Bluetooth devices in packets that are positioned in these slots. When circumstances permit, a number of consecutive slots may be allocated to a single packet. Frequency hopping takes place between the transmission or reception of packets. Bluetooth technology provides the effect of full duplex transmission through the use of a Time-Division Duplex (TDD) scheme.

Above the physical channel there is a layering of links and channels and associated control protocols. The hierarchy of channels and links from the physical channel upwards is physical channel, physical link, logical transport, logical link and L2CAP channel. These are discussed in more detail in Section 3.3 on page 49 to Section 3.6 on page 72 but are introduced here to aid the understanding of the remainder of this section.

Within a physical channel, a physical link is formed between a master device and slave devices. The physical link provides bidirectional packet transport between the master and slave devices. Since a physical channel could include multiple slave devices, there are restrictions on which devices may form a physical link. There is a physical link between each slave and the master. Physical links are not formed directly between the slaves in a piconet.

The physical link is used as a transport for one or more logical links that support unicast synchronous, asynchronous and isochronous traffic, and broadcast traffic. Traffic on logical links is multiplexed onto the physical link by occupying slots assigned by a scheduling function in the resource manager.

A control protocol for the baseband and physical layers is carried over logical links in addition to user data. This is the link manager protocol (LMP). Devices that are active in a piconet have a default asynchronous connection-oriented logical transport that is used to transport the LMP protocol signaling. For historical reasons this is known as the ACL logical transport. The default ACL logical transport is the one that is created whenever a device joins a piconet. Additional logical transports may be created to transport synchronous data streams when this is required.

The Link Manager function uses LMP to control the operation of devices in the piconet and provide services to manage the lower architectural layers (radio

layer and baseband layer). The LMP protocol is only carried on the default ACL logical transport and the default broadcast logical transport.

Above the baseband layer the L2CAP layer provides a channel-based abstraction to applications and services. It carries out segmentation and reassembly of application data and multiplexing and de-multiplexing of multiple channels over a shared logical link. L2CAP has a protocol control channel that is carried over the default ACL logical transport. Application data submitted to the L2CAP protocol may be carried on any logical link that supports the L2CAP protocol.

1.2 OVERVIEW OF BLUETOOTH LOW ENERGY OPERATION

Like the BR/EDR radio, the LE radio operates in the unlicensed 2.4 GHz ISM band. The LE system employs a frequency hopping transceiver to combat interference and fading and provides many FHSS carriers. LE radio operation uses a shaped, binary frequency modulation to minimize transceiver complexity. The symbol rate is 1 Megasymbol per second (Ms/s) supporting the bit rate of 1 Megabit per second (Mb/s).

LE employs two multiple access schemes: Frequency division multiple access (FDMA) and time division multiple access (TDMA). Forty (40) physical channels, separated by 2 MHz, are used in the FDMA scheme. Three (3) are used as advertising channels and 37 are used as data channels. A TDMA based polling scheme is used in which one device transmits a packet at a predetermined time and a corresponding device responds with a packet after a predetermined interval.

The physical channel is sub-divided into time units known as events. Data is transmitted between LE devices in packets that are positioned in these events. There are two types of events: Advertising and Connection events.

Devices that transmit advertising packets on the advertising PHY channels are referred to as **advertisers**. Devices that receive advertising on the advertising channels without the intention to connect to the advertising device are referred to as **scanners**. Transmissions on the advertising PHY channels occur in advertising events. At the start of each advertising event, the advertiser sends an advertising packet corresponding to the advertising event type. Depending on the type of advertising packet, the scanner may make a request to the advertiser on the same advertising PHY channel which may be followed by a response from the advertiser on the same advertising PHY channel. The advertising PHY channel changes on the next advertising packet sent by the advertiser in the same advertising event. The advertiser may end the advertising event at any time during the event. The first advertising PHY channel is used at the start of the next advertising event.

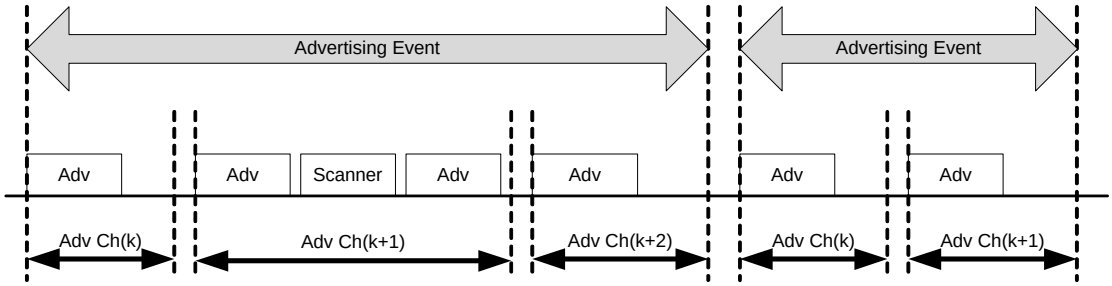


Figure 1.3: Advertising Events

LE devices may fulfill the entire communication in the case of unidirectional or broadcast communication between two or more devices using advertising events. They may also use advertising events to establish pair-wise bi-directional communication between two or more devices using data channels.

Devices that need to form a connection to another device listen for connectable advertising packets. Such devices are referred to as **initiators**. If the advertiser is using a connectable advertising event, an initiator may make a connection request using the same advertising PHY channel on which it received the connectable advertising packet. The advertising event is ended and connection events begin if the advertiser receives and accepts the request for a connection be initiated. Once a connection is established, the initiator becomes the **master** device in what is referred to as a **piconet** and the advertising device becomes the **slave** device. Connection events are used to send data packets between the master and slave devices. In connection events, channel hopping occurs at the start of each connection event. Within a connection event, the master and slave alternate sending data packets using the same data PHY channel. The master initiates the beginning of each connection event and can end each connection event at any time.

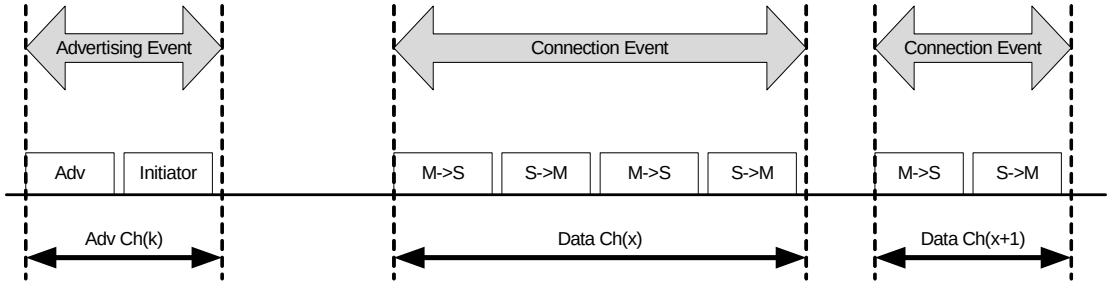


Figure 1.4: Connection Events

Devices in a piconet use a specific frequency hopping pattern, which is algorithmically determined by a field contained in the connection request sent by an initiating device. The initiating device provides the synchronization reference known as a hop interval. The hopping pattern used in LE is a pseudo-random ordering of the 37 frequencies in the ISM band. The hopping pattern can be adapted to exclude a portion of the frequencies that are used by interfering devices. The adaptive hopping technique improves Bluetooth co-existence with

static (non-hopping) ISM systems when these are co-located and have access to information about the local radio environment, or detected by other means.

Above the physical channel there are concepts of links, channels and associated control protocols. The hierarchy is physical channel, physical link, logical transport, logical link and L2CAP channel. These are discussed in more detail in Section 3.3 on page 49 to Section 3.6 on page 72 but are introduced here to aid the understanding of the remainder of this section.

Within a physical channel, a physical link is formed between a master and each slave. Physical links between slaves in a piconet are not supported. At this time, slaves are not permitted to have physical links to more than one master at a time. Role changes between a master and slave device are also not permitted at this time.

The physical link is used as a transport for one or more logical links that support asynchronous traffic. Traffic on logical links is multiplexed onto the physical link assigned by a scheduling function in the resource manager.

A control protocol for the link and physical layers is carried over logical links in addition to user data. This is the link layer protocol (LL). Devices that are active in a piconet have a default LE asynchronous connection logical transport (LE ACL) that is used to transport the LL protocol signaling. The default LE ACL is the one that is created whenever a device joins a piconet.

The Link Layer function uses the LL protocol to control the operation of devices in the piconet and provide services to manage the lower architectural layers (PHY and LL).

Just as in BR/EDR, above the link layer the L2CAP layer provides a channel-based abstraction to applications and services. It carries out fragmentation and de-fragmentation of application data and multiplexing and de-multiplexing of multiple channels over a shared logical link. L2CAP has a protocol control channel that is carried over the default logical transport.

In addition to L2CAP, LE provides two additional protocol layers that reside on top of L2CAP. The Security Manager protocol (SMP) uses a fixed L2CAP channel to implement the security functions between devices. The other is the Attribute protocol (ATT) that provides a method to communicate small amounts of data over a fixed L2CAP channel. The Attribute protocol is also used by devices to determine the services and capabilities of other devices. The Attribute protocol may also be used over BR/EDR.

1.3 OVERVIEW OF AMP OPERATION

Alternate MAC/PHYs (AMP) are secondary Controllers in the Bluetooth core system. The BR/EDR radio, the primary radio, is used to perform discovery, association, connection establishment, and connection maintenance. Once an L2CAP connection has been established between two devices over the BR/

EDR radio, the AMP Managers can discover the AMPs that are available on the other device. When an AMP is common between the two devices, the Core system provides mechanisms for moving data traffic from BR/EDR Controller to an AMP Controller.

Each AMP consists of a Protocol Adaptation Layer (PAL) on top of a MAC and PHY. The PAL is responsible for mapping the Bluetooth protocols and behavior (as specified by HCI) to the specific protocols of the underlying MAC and PHY.

L2CAP channels may be created on, or moved to, an AMP. L2CAP channels may also be moved back to the BR/EDR radio when those capabilities are not necessary or when the AMP physical link has a link supervision timeout. A link supervision timeout on the BR/EDR radio forces a disconnection of all AMP physical links between those devices.

AMPs may be enabled or disabled as needed in order to minimize power consumption in the system.

1.4 NOMENCLATURE

Where the following terms appear in the specification they have the meaning given in Table 1.1 on page 24.

Ad Hoc Network	A network typically created in a spontaneous manner. An ad hoc network requires no formal infrastructure and is limited in temporal and spatial extent.
Advertiser	A Bluetooth low energy device that broadcasts advertising packets during advertising events on advertising channels
Advertising Event	A series of between one and three advertising packets on different advertising channels sent by an advertiser.
Active Slave Broadcast (ASB)	The Active Slave Broadcast logical transport that is used to transport L2CAP user traffic to all active devices in the piconet over the BR/EDR Controller. See Section 3.5.4.4 on page 66
AMP	Alternate Media Access Controller (MAC) and Physical Layer (PHY) or Alternate MAC/PHY.
AMP Controller	A term referring to AMP PHY, AMP MAC, Protocol Adaptation Layer, and HCI layers.
Beacon Train	A pattern of reserved slots within a basic or adapted piconet physical channel. Transmissions starting in these slots are used to resynchronize parked devices.
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 AV and data communications between Bluetooth Hosts. The link protocol is based on time slots.
Bluetooth Baseband	The part of the Bluetooth system that specifies or implements the medium access and physical layer procedures to support the exchange of real-time voice, data information streams, and ad hoc networking between Bluetooth Devices.
Bluetooth Clock	A 28 bit clock internal to a BR/EDR Controller sub-system that ticks every 312.5μs. The value of this clock defines the slot numbering and timing in the various physical channels.
Bluetooth Controller	A generic term referring to a Primary Controller with or without a Secondary Controller.
Bluetooth Device	A device that is capable of short-range wireless communications using the Bluetooth system.
Bluetooth Device Address	A 48 bit address used to identify each Bluetooth device.

Table 1.1: Nomenclature.

BD_ADDR	The Bluetooth device Address, BD_ADDR, is used to identify a Bluetooth device.
BR/EDR	Bluetooth basic rate (BR) and enhanced data rate (EDR).
BR/EDR Controller	A term referring to the Bluetooth Radio, Baseband, Link Manager, and HCI layers.
BR/EDR/LE	Bluetooth basic rate (BR), enhanced data rate (EDR) and low energy (LE).
BR/EDR Piconet Physical Channel	A Channel that is divided into time slots in which each slot is related to an RF hop frequency. Consecutive hops normally correspond to different RF hop frequencies and occur at a standard hop rate of 1600 hops/s. These consecutive hops follow a pseudo-random hopping sequence, hopping through a 79 RF channel set, or optionally fewer channels when Adaptive Frequency Hopping (AFH) is in use.
C-plane	Control plane
Channel	Either a physical channel or an L2CAP channel, depending on the context.
Connect (to service)	The establishment of a connection to a service. If not already done, this also includes establishment of a physical link, logical transport, logical link and L2CAP channel.
Connectable device	A BR/EDR device in range that periodically listens on its page scan physical channel and will respond to a page on that channel. An LE device that is advertising using a connectable advertising event.
Connected devices	Two BR/EDR devices and with a physical link between them.
Connecting	A phase in the communication between devices when a connection between the devices is being established. (Connecting phase follows after the link establishment phase is completed.)
Connection	A connection between two peer applications or higher layer protocols mapped onto an L2CAP channel.
Connection establishment	A procedure for creating a connection mapped onto a channel.
Connection Event	A series of one or more pairs of interleaving data packets sent between a master and a slave on the same physical channel.
Controller	A collective term referring to all of the layers below HCI.
Coverage area	The area where two Bluetooth devices can exchange messages with acceptable quality and performance.

Table 1.1: Nomenclature.

Creation of a secure connection	A procedure of establishing a connection, including authentication and encryption.
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, or pairing, when a link key is not available.
Device discovery	A procedure for retrieving the Bluetooth device address, clock, class-of-device field and used page scan mode from discoverable devices.
Discoverable device	A BR/EDR device in range that periodically listens on an inquiry scan physical channel and will respond to an inquiry on that channel. An LE device in range that is advertising with a connectable or scannable advertising event with a discoverable flag set in the advertising data. This device is in the discoverable mode.
Discoverable Mode	A Bluetooth device that is performing inquiry scans in BR/EDR or advertising with a discoverable or connectable advertising event with a discoverable flag set in LE.
Discovery procedure	A Bluetooth device that is carrying out the inquiry procedure in BR/EDR or scanning for advertisers using a discoverable or connectable advertising event with a discoverable flag set in LE.
HCI	The Host Controller Interface (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.
Host	A logical entity defined as all of the layers below the non-core profiles (e.g.Volume 3) and above the Host Controller Interface (HCI). A Bluetooth Host attached to a Bluetooth Controller may communicate with other Bluetooth Hosts attached to their Controllers as well.
Initiator	A Bluetooth low energy device that listens on advertising channels for connectable advertising events to form connections.
Inquiring device	A BR/EDR device that is carrying out the inquiry procedure. This device is performing the discovery procedure.
Inquiry	A procedure where a Bluetooth device transmits inquiry messages and listens for responses in order to discover the other Bluetooth devices that are within the coverage area.
Inquiry scan	A procedure where a Bluetooth device listens for inquiry messages received on its inquiry scan physical channel.

Table 1.1: Nomenclature.

Interoperability	The ability of two or more systems or components to exchange information and to use the information that has been exchanged.
Isochronous data	Information in a stream where each information entity in the stream is bound by a time relationship to previous and successive entities.
Known device	A Bluetooth device for which at least the BD_ADDR is stored.
L2CAP	Logical Link Control and Adaptation Protocol
L2CAP Channel	A logical connection on L2CAP level between two devices serving a single application or higher layer protocol.
L2CAP Channel establishment	A procedure for establishing a logical connection on L2CAP level.
LE	Bluetooth low energy (LE)
Link establishment	A procedure for establishing the default ACL link and hierarchy of links and channels between devices.
Link	Shorthand for a logical link.
Link key	A secret key that is known by two devices and is used to authenticate the link.
LMP authentication	An LMP level procedure for verifying the identity of a remote device.
LMP pairing	A procedure that authenticates two devices and creates a common link key that can be used as a basis for a trusted relationship or a (single) secure connection.
Logical link	The lowest architectural level used to offer independent data transport services to clients of the Bluetooth system.
Logical transport	Shared acknowledgement protocol and link identifiers between different logical links.
Name discovery	A procedure for retrieving the user-friendly name (the Bluetooth device name) of a connectable device.
Packet	Format of aggregated bits that are transmitted on a physical channel.
Page	The initial phase of the connection procedure where a device transmits a train of page messages until a response is received from the target device or a time-out occurs.
Page scan	A procedure where a device listens for page messages received on its page scan physical channel.
Paging device	A Bluetooth device that is carrying out the page procedure.

Table 1.1: Nomenclature.

Paired device	A Bluetooth device for which a link key has been created (either before connection establishment was requested or during connecting phase).
Parked device	A device operating in a basic mode piconet that is synchronized to the master but has given up its default ACL logical transport.
Passkey	A 6-digit number used to authenticate connections when Secure Simple Pairing is used.
Physical Channel	Characterized by synchronized occupancy of a sequence of RF carriers by one or more devices. A number of physical channel types exist with characteristics defined for their different purposes.
Physical Link	A Baseband or Link Layer level connection between two devices.
Piconet	A collection of devices occupying a shared physical channel where one of the devices is the Piconet Master and the remaining devices are connected to it.
Piconet Master	The BR/EDR device in a piconet whose Bluetooth Clock and Bluetooth Device Address are used to define the piconet physical channel characteristics.
Piconet Slave	Any BR/EDR device in a piconet that is not the Piconet Master, but is connected to the Piconet Master.
PIN	A user-friendly number that can be used to authenticate connections to a device before pairing has taken place.
PMP	A Participant in Multiple Piconets. A device that is concurrently a member of more than one piconet, which it achieves using time division multiplexing (TDM) to interleave its activity on each piconet physical channel.
The Parked Slave Broadcast (PSB)	The Parked Slave Broadcast logical transport that is used for communications between the master and parked devices. These communications may also be received by active devices. See Section 3.5.4.5 on page 67.
Scanner	A Bluetooth low energy device that listens for advertising events on the advertising channels.
Scatternet	Two or more piconets that include one or more devices acting as PMPs.
Service Layer Protocol	A protocol that uses an L2CAP channel for transporting PDUs.
Service discovery	Procedures for querying and browsing for services offered by or through another Bluetooth device.
Silent device	A Bluetooth enabled device appears as silent to a remote device if it does not respond to inquiries made by the remote device.

Table 1.1: Nomenclature.

U-plane	User plane
Unknown device	A Bluetooth device for which no information (Bluetooth Device Address, link key or other) is stored.

Table 1.1: Nomenclature.

2 CORE SYSTEM ARCHITECTURE

The Bluetooth Core system consists of a Host, a Primary Controller and zero or more Secondary Controllers. A minimal implementation of the Bluetooth BR/EDR core system covers the four lowest layers and associated protocols defined by the Bluetooth specification as well as one common service layer protocol; the Service Discovery Protocol (SDP) and the overall profile requirements are specified in the Generic Access Profile (GAP). The BR/EDR Core system includes support of Alternate MAC/PHYs (AMPs) including an AMP Manager Protocol and Protocol Adaptation Layers (PALs) supporting externally referenced MAC/PHYs. A minimal implementation of a Bluetooth LE only core system covers the four lowest layers and associated protocols defined by the Bluetooth specification as well as two common service layer protocols; the Security Manager (SM) and Attribute Protocol (ATT) and the overall profile requirements are specified in the Generic Attribute Profile (GATT) and Generic Access Profile (GAP). Implementations combining Bluetooth BR/EDR and LE include both of the minimal implementations described above.

A complete Bluetooth application requires a number of additional service and higher layer protocols that are defined in the Bluetooth specification, but are not described here. The core system architecture is shown in Figure 2.1 on page 31.

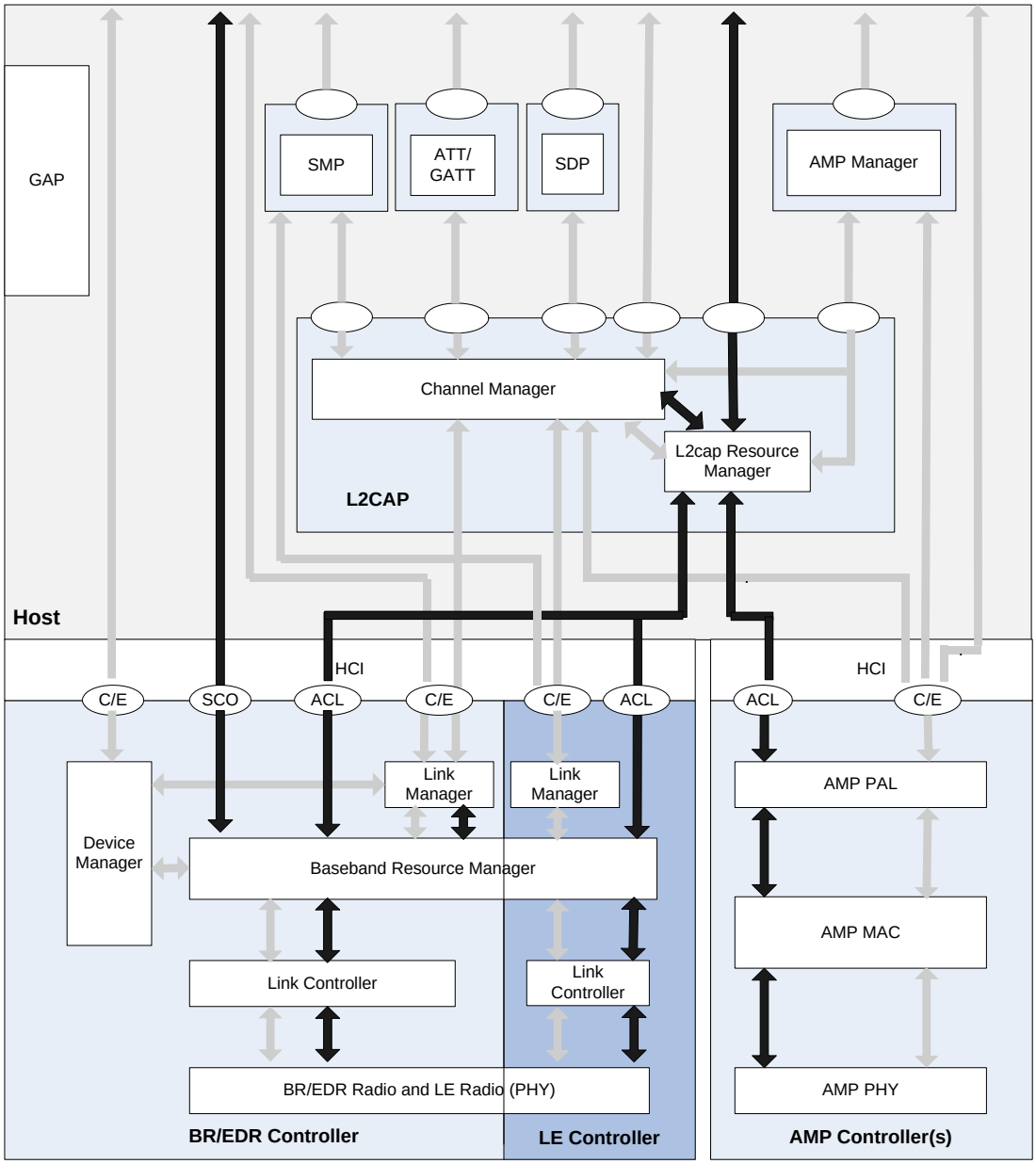


Figure 2.1: Bluetooth core system architecture

Figure 2.1 on page 31 shows the Core blocks, each with its associated communication protocol. Link Manager, Link Controller and BR/EDR Radio blocks comprise a BR/EDR Controller. An AMP PAL, AMP MAC, and AMP PHY comprise an AMP Controller. Link Manager, Link Controller and LE Radio blocks comprise an LE Controller. L2CAP, SDP and GAP blocks comprise a BR/EDR Host. L2CAP, SMP, Attribute protocol, GAP and Generic Attribute Profile (GATT) blocks comprise an LE Host. A BR/EDR/LE Host combines the set of blocks from each respective Host. This is a common implementation involving a standard physical communication interface between the Controller and the Host. Although this interface is optional the architecture is designed to allow for its existence and characteristics. The Bluetooth specification enables interoperability between independent Bluetooth systems by defining the protocol mes-

sages exchanged between equivalent layers, and also interoperability between independent Bluetooth subsystems by defining a common interface between Bluetooth controllers and Bluetooth hosts.

A number of functional blocks and the path of services and data between them are shown. The functional blocks shown in the diagram are informative; in general the Bluetooth specification does not define the details of implementations except where this is required for interoperability. Thus the functional blocks in Figure 2.1 on page 31 are shown in order to aid description of the system behavior. An implementation may be different from the system shown in Figure 2.1 on page 31.

Standard interactions are defined for all inter-device operation, where Bluetooth devices exchange protocol signaling according to the Bluetooth specification. The Bluetooth core system protocols are the Radio (PHY) protocol, Link Control (LC) and Link Manager (LM) protocol or Link Layer (LL) protocol, AMP PAL, Logical Link Control and Adaptation protocol (L2CAP), and AMP Manager Protocol, all of which are fully defined in subsequent parts of the Bluetooth specification. In addition, the Service Discovery Protocol (SDP) and the Attribute Protocol (ATT) are service layer protocols that may be required by some Bluetooth applications.

The Bluetooth core system offers services through a number of service access points that are shown in the diagram as ellipses. These services consist of the basic primitives that control the Bluetooth core system. The services can be split into three types. There are device control services that modify the behavior and modes of a Bluetooth device, transport control services that create, modify and release traffic bearers (channels and links), and data services that are used to submit data for transmission over traffic bearers. It is common to consider the first two as belonging to the C-plane and the last as belonging to the U-plane.

A service interface to the Bluetooth Controller subsystem is defined such that the Primary Controller may be considered a standard part. In this configuration the Bluetooth Controller operates the lowest four layers. The Bluetooth Host operates the L2CAP layer and other higher layers. The standard interface is called the Host Controller Interface (HCI) and its service access points are represented by the ellipses on the upper edge of the Bluetooth controller subsystem in Figure 2.1 on page 31. Implementation of this standard service interface is optional.

As the Bluetooth architecture is defined with the possibility of separate host and controller(s) communicating through one or more HCI transports, a number of general assumptions are made. Bluetooth controllers are assumed to have limited data buffering capabilities in comparison with the host. Therefore the L2CAP layer is expected to carry out some simple resource management when submitting L2CAP PDUs to the controller for transport to a peer device. This includes segmentation of L2CAP SDUs into more manageable PDUs and then the fragmentation of PDUs into start and continuation packets of a size

suitable for the controller buffers, and management of the use of controller buffers to ensure availability for channels with Quality of Service (QoS) commitments.

The BR/EDR Baseband, LE Link Layer, and AMP MAC layers provides the basic acknowledgement/repeat request (ARQ) protocol in Bluetooth. The L2CAP layer can optionally provide a further error detection and retransmission to the L2CAP PDUs. This feature is recommended for applications with requirements for a low probability of undetected errors in the user data. A further optional feature of L2CAP is a window-based flow control that can be used to manage buffer allocation in the receiving device. Both of these optional features augment the QoS performance in certain scenarios. Not all of the L2CAP capabilities are available when using the LE system.

Although these assumptions may not be required for embedded Bluetooth implementations that combine all layers in a single system, the general architectural and QoS models are defined with these assumptions in mind, in effect a lowest common denominator.

Automated conformance testing of implementations of the Bluetooth core system is required. This is achieved by allowing the tester to control the implementation through the PHY interface, and test interfaces such as Direct Test Mode (DTM), Generic Test Methodology (GTM), Test Control Interface (TCI), and test commands and events over HCI which are only required for conformance testing.

The tester exchanges with the Implementation Under Test (IUT) through the PHY interface to ensure the correct responses to requests from remote devices. The tester controls the IUT through HCI, DTM, GTM, or TCI to cause the IUT to originate exchanges through the PHY interface so that these can also be verified as conformant.

The TCI uses a different command-set (service interface) for the testing of each architectural layer and protocol. A subset of the HCI command-set is used as the TCI service interface for each of the layers and protocols within the BR/EDR Controller subsystem. A separate service interface is used for testing the L2CAP layer and protocol. As an L2CAP service interface is not defined in the Bluetooth core specification it is defined separately in the Test Control Interface specification. Implementation of the L2CAP service interface is only required for conformance testing.

2.1 CORE ARCHITECTURAL BLOCKS

This section describes the function and responsibility of each of the blocks shown in Figure 2.1 on page 31. An implementation is not required to follow the architecture described above, though every implementation shall conform to the protocol specifications described in subsequent parts of the Bluetooth specification, and shall implement the behavioral aspects of the system outlined below and specified in subsequent parts of the Bluetooth specification.

2.1.1 Host Architectural Blocks

2.1.1.1 Channel Manager

The channel manager is responsible for creating, managing and closing L2CAP channels for the transport of service protocols and application data streams. The channel manager uses the L2CAP protocol to interact with a channel manager on a remote (peer) device to create these L2CAP channels and connect their endpoints to the appropriate entities. The channel manager interacts with its local link manager or an AMP PAL to create new logical links (if necessary) and to configure these links to provide the required quality of service for the type of data being transported.

2.1.1.2 L2CAP Resource Manager

The L2CAP resource manager block is responsible for managing the ordering of submission of PDU fragments to the baseband and some relative scheduling between channels to ensure that L2CAP channels with QoS commitments are not denied access to the physical channel due to Controller resource exhaustion. This is required because the architectural model does not assume that a Controller has limitless buffering, or that the HCI is a pipe of infinite bandwidth.

L2CAP Resource Managers may also carry out traffic conformance policing to ensure that applications are submitting L2CAP SDUs within the bounds of their negotiated QoS settings. The general Bluetooth data transport model assumes well-behaved applications, and does not define how an implementation is expected to deal with this problem.

2.1.1.3 Security Manager Protocol

The Security Manager Protocol (SMP) is the peer-to-peer protocol used to generate encryption keys and identity keys. The protocol operates over a dedicated fixed L2CAP channel. The SMP block also manages storage of the encryption keys and identity keys and is responsible for generating random addresses and resolving random addresses to known device identities. The SMP block interfaces directly with the controller to provide stored keys used for encryption and authentication during the encryption or pairing procedures.

This block is only used in LE systems. Similar functionality in the BR/EDR system is contained in the Link Manager block in the Controller. SMP functionality

is in the Host on LE systems to reduce the implementation cost of the LE only controllers.

2.1.1.4 Attribute Protocol

The Attribute Protocol (ATT) block implements the peer-to-peer protocol between an attribute server and an attribute client. The ATT client communicates with an ATT server on a remote device over a dedicated fixed L2CAP channel. The ATT client sends commands, requests, and confirmations to the ATT server. The ATT server sends responses, notifications and indications to the client. These ATT client commands and requests provide a means to read and write values of attributes on a peer device with an ATT server.

2.1.1.5 AMP Manager Protocol

The AMP manager is a layer that uses L2CAP to communicate with a peer AMP Manager on a remote device. It also interfaces directly with the AMP PAL for AMP control purposes. The AMP manager is responsible for discovering remote AMP(s) and determining their availability. It also collects information about remote AMP(s). This information is used to set up and manage AMP physical links. The AMP manager uses a dedicated L2CAP signaling channel to communicate with remote AMP manager(s).

2.1.1.6 Generic Attribute Profile

The Generic Attribute Profile (GATT) block represents the functionality of the attribute server and, optionally, the attribute client. The profile describes the hierarchy of services, characteristics and attributes used in the attribute server. The block provides interfaces for discovering, reading, writing and indicating of service characteristics and attributes. GATT is used on LE devices for LE profile service discovery.

2.1.1.7 Generic Access Profile

The Generic Access Profile (GAP) block represents the base functionality common to all Bluetooth devices such as modes and access procedures used by the transports, protocols and application profiles. GAP services include device discovery, connection modes, security, authentication, association models and service discovery.

2.1.2 BR/EDR/LE Controller Architectural Blocks

In implementations where the BR/EDR and LE systems are combined, the architectural blocks may be shared between systems or each system may have their own instantiation of the block.

2.1.2.1 Device Manager

The device manager is the functional block in the baseband that controls the general behavior of the Bluetooth device. It is responsible for all operations of the Bluetooth system that are not directly related to data transport, such as inquiring for the presence of nearby Bluetooth devices, connecting to Bluetooth devices, or making the local Bluetooth device discoverable or connectable by other devices.

The device manager requests access to the transport medium from the baseband resource controller in order to carry out its functions.

The device manager also controls local device behavior implied by a number of the HCI commands, such as managing the device local name, any stored link keys, and other functionality.

2.1.2.2 Link Manager

The link manager is responsible for the creation, modification and release of logical links (and, if required, their associated logical transports), as well as the update of parameters related to physical links between devices. The link manager achieves this by communicating with the link manager in remote Bluetooth devices using the Link Management Protocol (LMP) in BR/EDR and the Link Layer Protocol (LL) in LE.

The LM or LL protocol allows the creation of new logical links and logical transports between devices when required, as well as the general control of link and transport attributes such as the enabling of encryption on the logical transport, the adapting of transmit power in BR/EDR on the physical link, or the adjustment of QoS settings in BR/EDR for a logical link.

2.1.2.3 Baseband Resource Manager

The baseband resource manager is responsible for all access to the radio medium. It has two main functions. At its heart is a scheduler that grants time on the physical channels to all of the entities that have negotiated an access contract. The other main function is to negotiate access contracts with these entities. An access contract is effectively a commitment to deliver a certain QoS that is required in order to provide a user application with an expected performance.

The access contract and scheduling function must take account of any behavior that requires use of the Primary Controller. This includes (for example) the normal exchange of data between connected devices over logical links, and logical transports, as well as the use of the radio medium to carry out inquiries, make connections, be discoverable or connectable, or to take readings from unused carriers during the use of adaptive frequency hopping mode.

In some cases in BR/EDR systems the scheduling of a logical link results in changing a logical link to a different physical channel from the one that was

previously used. This may be (for example) due to involvement in scatternet, a periodic inquiry function, or page scanning. When the physical channels are not time slot aligned, then the resource manager also accounts for the realignment time between slots on the original physical channel and slots on the new physical channel. In some cases the slots will be naturally aligned due to the same device clock being used as a reference for both physical channels.

2.1.2.4 Link Controller

The link controller is responsible for the encoding and decoding of Bluetooth packets from the data payload and parameters related to the physical channel, logical transport and logical link.

The link controller carries out the link control protocol signaling in BR/EDR and link layer protocol in LE (in close conjunction with the scheduling function of the resource manager), which is used to communicate flow control and acknowledgement and retransmission request signals. The interpretation of these signals is a characteristic of the logical transport associated with the baseband packet. Interpretation and control of the link control signaling is normally associated with the resource manager's scheduler.

2.1.2.5 PHY

The PHY block is responsible for transmitting and receiving packets of information on the physical channel. A control path between the baseband and the PHY block allows the baseband block to control the timing and frequency carrier of the PHY block. The PHY block transforms a stream of data to and from the physical channel and the baseband into required formats.

2.1.3 AMP Controller architectural blocks

2.1.3.1 AMP HCI

The AMP HCI is the logical interface between an AMP Controller and Host (L2CAP and AMP manager). HCI is an optional layer used when the Host and AMP Controller(s) are physically separated. Support for AMPs requires additional HCI commands and events. These new commands and events are related to AMP physical and logical link management, QoS, as well as flow control.

There is one logical instantiation of an HCI per AMP Controller, and another logical instantiation for the BR/EDR Controller. The physical HCI transport layer manages the multiplexing of several Controllers over the same physical transport bus in cases where multiple Controllers exist in a single physical unit.

2.1.3.2 AMP PAL

The AMP PAL is the AMP layer interfacing the AMP MAC with the Host (L2CAP and AMP Manager). It translates commands from the Host into the specific MAC service primitives and primitives into commands, and it translates primitives from the AMP MAC into understandable event(s) for the Host. The AMP PAL provides support for AMP channel management, data traffic according to specified flow specifications, and power efficiency.

There is not a requirement that the AMP PAL has exclusive access to the AMP itself. Implementers may choose to allow other protocols to be concurrent clients of the AMP. Specific mechanisms for allowing other protocols to share access to the AMP are outside the scope of this specification.

A generic PAL architecture is shown in Figure 2.2.

.

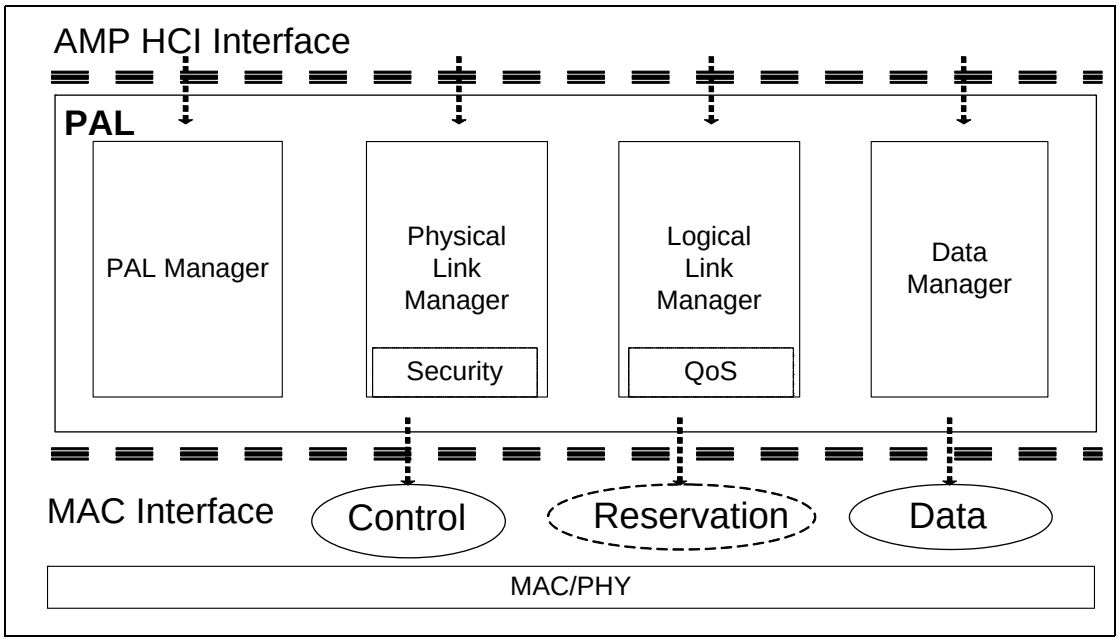


Figure 2.2: Generic PAL architecture

Capabilities of the PAL and the underlying MAC/PHY are exchanged via the AMP_Info and AMP_Assoc structures by the AMP Manager Protocol. The AMP_Info contains generic capabilities of the PAL such as bandwidth availability, maximum PDU size, PAL capabilities, length of the AMP_Assoc and flush timeout information. The contents of the AMP_Assoc are dependent on the PAL and are related to the underlying MAC/PHY

2.1.3.3 AMP MAC

The AMP MAC is the MAC layer as defined in the IEEE 802 reference layer model. It provides services such as addressing and mechanisms to control and access channels. The AMP MAC is in between AMP PHY and AMP PAL layers.

2.1.3.4 AMP PHY

The AMP PHY is the AMP physical layer.

3 DATA TRANSPORT ARCHITECTURE

The Bluetooth data transport system follows a layered architecture. This description of the Bluetooth system describes the Bluetooth core transport layers up to and including L2CAP channels. All Bluetooth operational modes follow the same generic transport architecture, which is shown in Figure 3.1 on page 39.

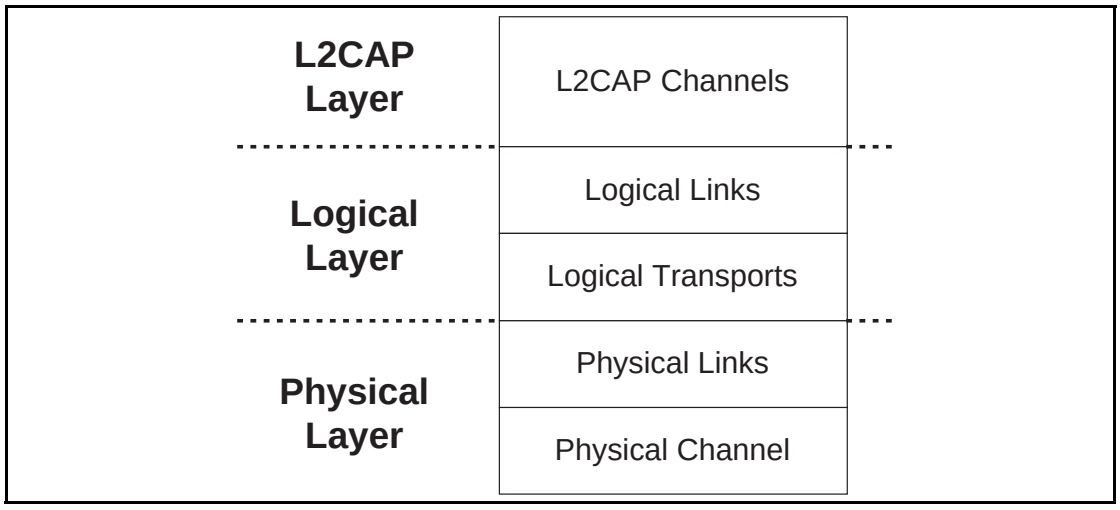


Figure 3.1: Bluetooth generic data transport architecture

For efficiency and legacy reasons, the Bluetooth transport architecture includes a sub-division of the logical layer, distinguishing between logical links and logical transports. This sub-division provides a general (and commonly understood) concept of a logical link that provides an independent transport between two or more devices. The logical transport sub-layer is required to describe the inter-dependence between some of the logical link types (mainly for reasons of legacy behavior.)

Prior to Version 1.2, the Bluetooth Core specification described the ACL and SCO links as physical links. With the addition of Extended SCO (eSCO) and for future expansion it is better to consider these as logical transport types, which more accurately encapsulates their purpose. However, they are not as independent as might be desired, due to their shared use of resources such as the LT_ADDR and acknowledgement/repeat request (ARQ) scheme. Hence the architecture is incapable of representing these logical transports with a single transport layer. The additional logical transport layer goes some way towards describing this behavior.

3.1 CORE TRAFFIC BEARERS

The Bluetooth core system provides a number of standard traffic bearers for the transport of service protocol and application data. These are shown in Figure 3.2 on page 40 below (for ease of representation this is shown with higher layers to the left and lower layers to the right).

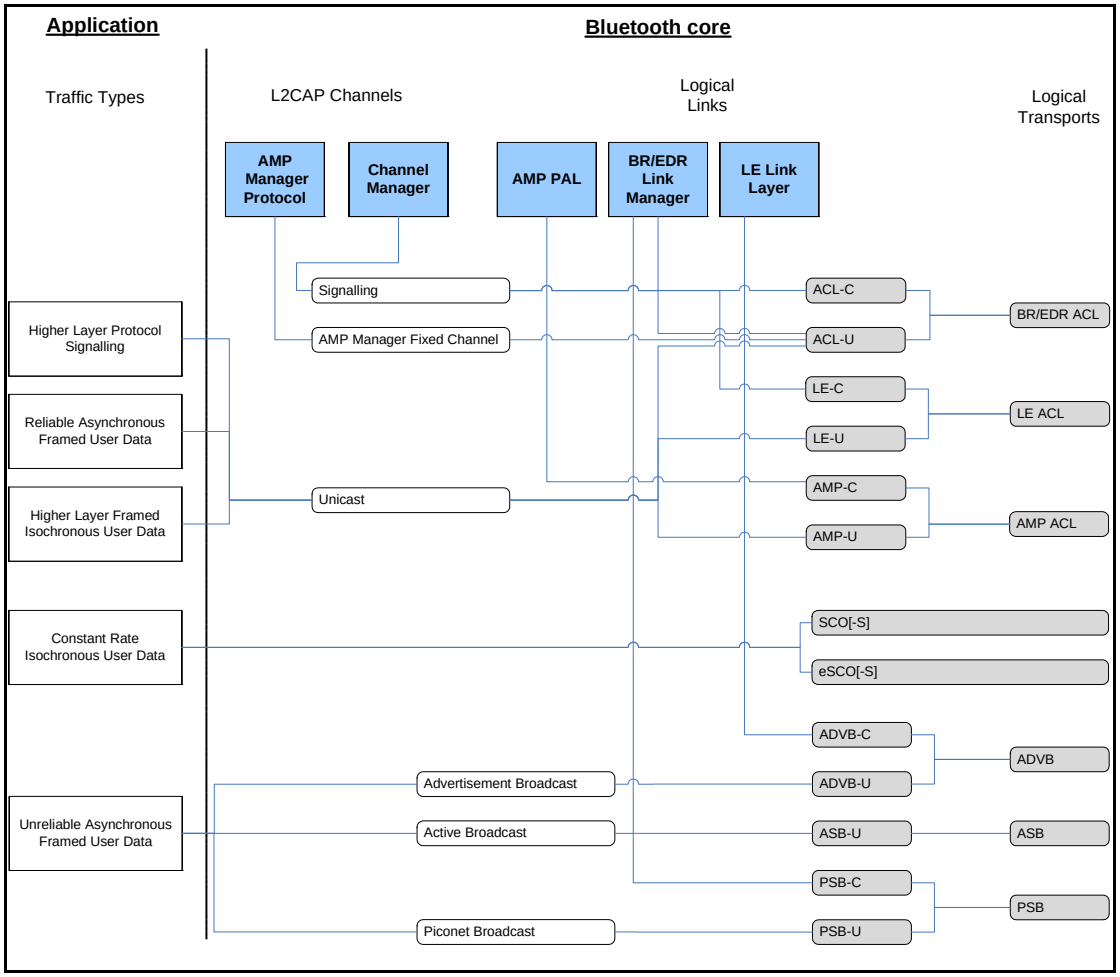


Figure 3.2: Bluetooth traffic bearers

The core traffic bearers that are available to applications are shown in Figure 3.2 on page 40 as the shaded rounded rectangles. The architectural layers that are defined to provide these services are described in Section 2 on page 30. A number of data traffic types are shown on the left of the diagram linked to the traffic bearers that are typically suitable for transporting that type of data traffic.

The logical links are named using the names of the associated logical transport and a suffix that indicates the type of data that is transported. (C for control links carrying LMP or LL messages, U for L2CAP links carrying user data (L2CAP PDUs) and S for stream links carrying unformatted synchronous or isochronous data.) It is common for the suffix to be removed from the logical link without introducing ambiguity, thus a reference to the default ACL logical transport can be resolved to mean the ACL-C logical link in cases where the

LMP protocol is being discussed, the LE-C logical link in cases where LL protocol is being discussed, the AMP-C logical link in cases where a PAL protocol is being discussed, or the ACL-U,LE-U, or AMP-U logical links when the L2CAP layer is being discussed.

The mapping of application traffic types to Bluetooth core traffic bearers in Figure 3.2 on page 40 is based on matching the traffic characteristics with the bearer characteristics. It is recommended to use these mappings as they provide the most natural and efficient method of transporting the data with its given characteristics.

However, an application (or an implementation of the Bluetooth core system) may choose to use a different traffic bearer, or a different mapping to achieve a similar result. For example, in a BR/EDR piconet with only one slave, the master may choose to transport L2CAP broadcasts over the ACL-U logical link rather than over the ASB-U or PSB-U logical links. This will probably be more efficient in terms of bandwidth (if the physical channel quality is not too degraded.) Use of alternative transport paths to those in Figure 3.2 on page 40 is only acceptable if the characteristics of the application traffic type are preserved.

Figure 3.2 on page 40 shows a number of application traffic types. These are used to classify the types of data that may be submitted to the Bluetooth core system. The original data traffic type may not be the same as the type that is submitted to the Bluetooth core system if an intervening process modifies it. For example, video data is generated at a constant rate but an intermediate coding process may alter this to variable rate, e.g. by MPEG4 encoding. For the purposes of the Bluetooth core system, only the characteristic of the submitted data is of interest.

3.1.1 Framed Data Traffic

The L2CAP layer services provide a frame-oriented transport for asynchronous and isochronous user data. The application submits data to this service in variable-sized frames (up to a negotiated maximum for the channel) and these frames are delivered in the same form to the corresponding application on the remote device. There is no requirement for the application to insert additional framing information into the data, although it may do so if this is required (such framing is invisible to the Bluetooth core system.)

Connection-oriented BR/EDR L2CAP channels may be created for transport of unicast (point-to-point) data between two Bluetooth devices. Connection-oriented channels provide a context within which specific properties may be applied to data transported on the channel. For example, quality of service parameters or flow and error control modes may be applied. Connection-oriented L2CAP channels are created using the L2CAP connection procedure.

A connectionless BR/EDR L2CAP channel exists for broadcasting data or for transport of unicast data. In the case of piconet topologies the master device is always the source of broadcast data and the slave device(s) are the recipients.

Broadcast traffic on the connectionless L2CAP channel is uni-directional. Unicast data sent on the connectionless L2CAP channels may be uni-directional or bi-directional. Unicast data sent on the L2CAP connectionless channel provides an alternate mechanism to send data with the same level of reliability as an L2CAP connection-oriented channel operating in Basic mode but without the additional latency incurred by opening an L2CAP connection-oriented channel.

BR/EDR L2CAP channels have an associated QoS setting that defines constraints on the delivery of the frames of data. These QoS settings may be used to indicate (for example) that the data is isochronous, and therefore has a limited lifetime after which it becomes invalid, or that the data should be delivered within a given time period, or that the data is reliable and should be delivered without error, however long this takes.

Only fixed channels are supported in the LE system; connection-oriented and connectionless L2CAP channels are not supported.

Some L2CAP channels are fixed channels created when the ACL-U and/or LE-U logical links are established. These fixed channels have fixed channel identifiers and fixed configurations and do not permit negotiation of the configuration after they are created. These fixed channels are used for BR/EDR and LE L2CAP signaling (ACL-U or LE-U), connectionless channel (ACL-U), AMP manager protocol (ACL-U), Security Manager Protocol (LE-U), Attribute Protocol (ACL-U or LE-U) and AMP Test Manager (ACL-U).

The L2CAP channel manager is responsible for arranging to transport the L2CAP channel data frames on an appropriate baseband logical link, possibly multiplexing this onto the baseband logical link with other L2CAP channels with similar characteristics.

3.1.2 Unframed Data Traffic

If the application does not require delivery of data in frames, possibly because it includes in-stream framing, or because the data is a pure stream, then it may avoid the use of L2CAP channels and make direct use of a baseband logical link.

The Bluetooth core system supports the direct transport of application data that is isochronous and of a constant rate (either bit-rate, or frame-rate for pre-framed data), using a SCO-S or eSCO-S logical link. These logical links reserve physical channel bandwidth and provide a constant rate transport locked to the piconet clock. Data is transported in fixed size packets at fixed intervals with both of these parameters negotiated during channel establishment. eSCO links provide a greater choice of bit-rates and also provide greater reliability by using limited retransmission in case of error. Enhanced Data Rate operation is supported for eSCO, but not for SCO logical transports. SCO and eSCO logical transports do not support multiplexed logical links or any further layering within the Bluetooth core. An application may choose to layer a num-

ber of streams within the submitted SCO/eSCO stream, provided that the submitted stream is, or has the appearance of being, a constant rate stream.

The application chooses the most appropriate type of logical link from those available at the baseband, and creates and configures it to transport the data stream, and releases it when completed. (The application will normally also use a framed L2CAP unicast channel to transport its C-plane information to the peer application on the remote device.)

If the application data is isochronous and of a variable rate, then this may only be carried by the L2CAP unicast channel, and hence will be treated as framed data.

Unframed data traffic is not supported in the LE system.

3.1.3 Reliability of traffic bearers

3.1.3.1 BR/EDR Reliability

Bluetooth is a wireless communications system. In poor RF environments, this system should be considered inherently unreliable. To counteract this the system provides levels of protection at each layer. The baseband packet header uses forward error correcting (FEC) coding to allow error correction by the receiver and a header error check (HEC) to detect errors remaining after correction. Certain Baseband packet types include FEC for the payload. Furthermore, some Baseband packet types include a cyclic redundancy error check (CRC).

On ACL logical transports the results of the error detection algorithm are used to drive a simple ARQ protocol. This provides an enhanced reliability by retransmitting packets that do not pass the receiver's error checking algorithm. It is possible to modify this scheme to support latency-sensitive packets by discarding an unsuccessfully transmitted packet at the transmitter if the packet's useful life has expired. eSCO links use a modified version of this scheme to improve reliability by allowing a limited number of retransmissions.

The resulting reliability gained by this ARQ scheme is only as dependable as the ability of the HEC and CRC codes to detect errors. In most cases this is sufficient, however it has been shown that for the longer packet types the probability of an undetected error is too high to support typical applications, especially those with a large amount of data being transferred.

The L2CAP layer provides an additional level of error control that is designed to detect the occasional undetected errors in the baseband layer and request retransmission of the affected data. This provides the level of reliability required by typical Bluetooth applications.

Broadcast links have no feedback route, and are unable to use the ARQ scheme (although the receiver is still able to detect errors in received packets.)

Instead each packet is transmitted several times in the hope that the receiver is able to receive at least one of the copies successfully. Despite this approach there are still no guarantees of successful receipt, and so these links are considered unreliable.

In summary, if a link or channel is characterized as reliable this means that the receiver is capable of detecting errors in received packets and requesting retransmission until the errors are removed. Due to the error detection system used some residual (undetected) errors may still remain in the received data. For L2CAP channels the level of these is comparable to other communication systems, although for logical links the residual error level is somewhat higher.

The transmitter may remove packets from the transmit queue such that the receiver does not receive all the packets in the sequence. If this happens detection of the missing packets is delegated to the L2CAP layer.

On an unreliable link the receiver is capable of detecting errors in received packets but cannot request retransmission. The packets passed on by the receiver may be without error, but there is no guarantee that all packets in the sequence are received. Hence the link is considered fundamentally unreliable. There are limited uses for such links, and these uses are normally dependent on the continuous repetition of data from the higher layers while it is valid.

Stream links have a reliability characteristic somewhere between a reliable and an unreliable link, depending on the current operating conditions.

3.1.3.2 LE reliability

Like BR/EDR, in poor RF environments, the LE system should be considered inherently unreliable. To counteract this, the system provides levels of protection at each layer. The LL packet uses a 24-bit cyclic redundancy error check (CRC) to cover the contents of the packet payload. If the CRC verification fails on the packet payload, the packet is not acknowledged by the receiver and the packet gets retransmitted by the sender.

Broadcast links have no feedback route, and are unable to use the acknowledgement scheme (although the receiver is still able to detect errors in received packets.) Instead, each packet is transmitted several times to increase the probability that the receiver is able to receive at least one of the copies successfully. Despite this approach there are still no guarantees of successful receipt, and so these links are considered unreliable.

In summary, if a link or channel is characterized as reliable this means that the receiver is capable of detecting errors in received packets and requesting retransmission until the errors are removed.

The transmitter may remove packets from the transmit queue such that the receiver does not receive all the packets in the sequence. If this happens detection of the missing packets is delegated to the L2CAP layer.

On an unreliable link the receiver is capable of detecting errors in received packets but cannot request retransmission. The packets passed on by the receiver may be without error, but there is no guarantee that all packets in the sequence are received. Hence the link is considered fundamentally unreliable. There are limited uses for such links, and these uses are normally dependent on the continuous repetition of data from the higher layers while it is valid.

3.1.3.3 AMP Reliability

The reliability of each AMP depends on the underlying MAC/PHY technology. Some AMPs only flush user traffic when marked as flushable whereas other AMPs can flush user traffic.

The Bluetooth Core maintains a level of reliability for protocols and profiles above the Core by mandating the use of Enhanced Retransmission Mode or Streaming Mode for any L2CAP channel used over the AMP.

3.2 TRANSPORT ARCHITECTURE ENTITIES

The Bluetooth transport architecture entities are shown in Figure 3.3 on page 45 and are described from the lowest layer upwards in the subsequent sections.

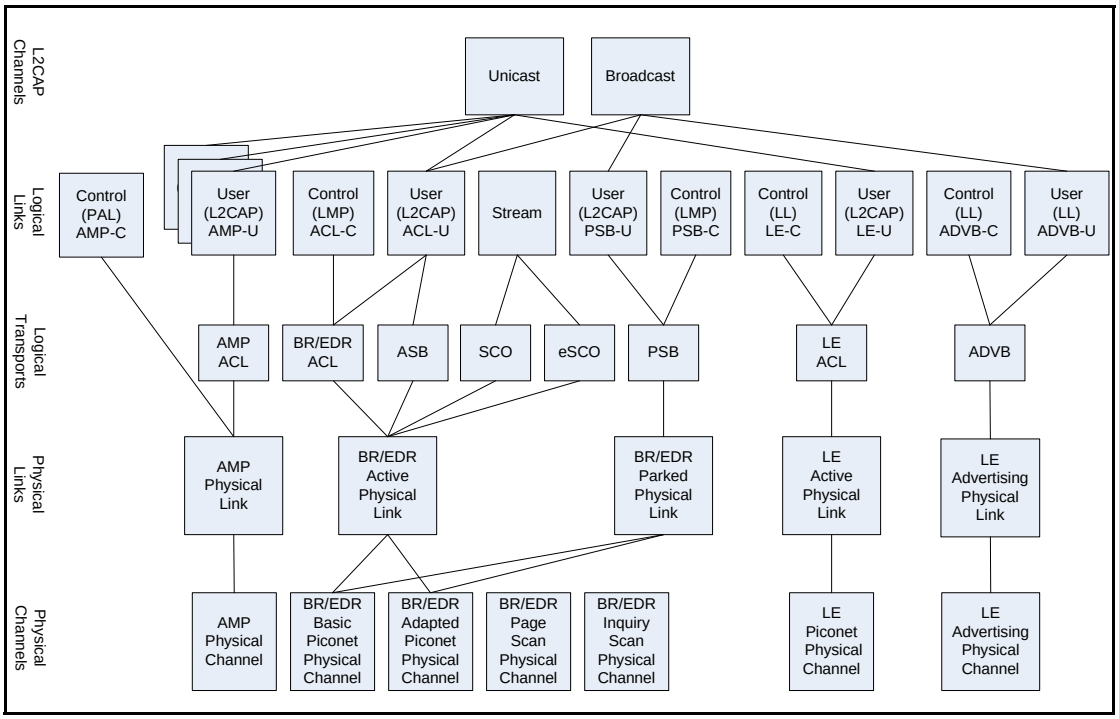


Figure 3.3: Overview of transport architecture entities and hierarchy

3.2.1 BR/EDR Generic Packet Structure

The general packet structure nearly reflects the architectural layers found in the Bluetooth BR/EDR system. The BR/EDR packet structure is designed for optimal use in normal operation. It is shown in Figure 3.4 on page 46.

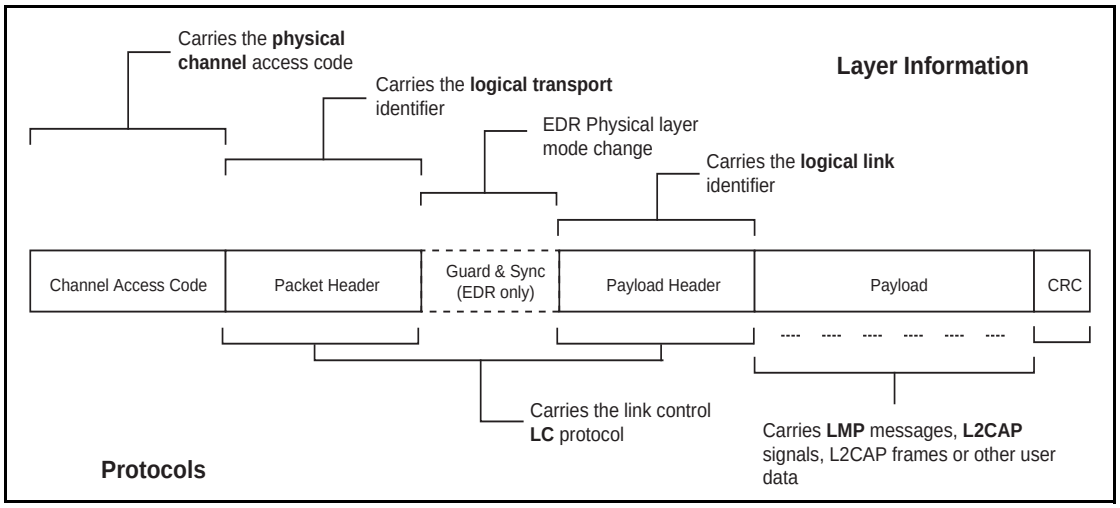


Figure 3.4: BR/EDR packet structure

Packets normally only include the fields that are necessary to represent the layers required by the transaction. Thus a simple inquiry request over an inquiry scan physical channel does not create or require a logical link or higher layer and therefore consists only of the channel access code (associated with the physical channel.) General communication within a piconet uses packets that include all of the fields, as all of the architectural layers are used.

All packets include the channel access code. This is used to identify communications on a particular physical channel, and to exclude or ignore packets on a different physical channel that happens to be using the same RF carrier in physical proximity.

There is no direct field within the BR/EDR packet structure that represents or contains information relating to physical links. This information is implied in the logical transport address (LT_ADDR) carried in the packet header.

Most BR/EDR packets include a packet header. The packet header is always present in packets transmitted on physical channels that support physical links, logical transports and logical links. The packet header carries the LT_ADDR, which is used by each receiving device to determine if the packet is addressed to the device and is used to route the packet internally.

The BR/EDR packet header also carries part of the link control (LC) protocol that is operated per logical transport (except for ACL and SCO transports that operate a shared LC protocol carried on either logical transport.)

The Enhanced Data Rate (EDR) packets have a guard time and synchronization sequence before the payload. This is a field used for physical layer change of modulation scheme.

The payload header is present in all packets on logical transports that support multiple logical links. The payload header includes a logical link identifier field used for routing the payload, and a field indicating the length of the payload. Some packet types also include a CRC after the packet payload that is used to detect most errors in received packets.

EDR packets have a trailer after the CRC.

The packet payload is used to transport the user data. The interpretation of this data is dependent on the logical transport and logical link identifiers. For ACL logical transports Link Manager Protocol (LMP) messages and L2CAP signals are transported in the packet payload, along with general user data from applications. For SCO and eSCO logical transports the payload contains the user data for the logical link.

3.2.2 LE Generic Packet Structure

The general structure of the link layer air interface packet closely reflects the architectural layers found in the LE system. The LE packet structure is designed for optimal use in normal operation. It is shown in Figure 3.5.

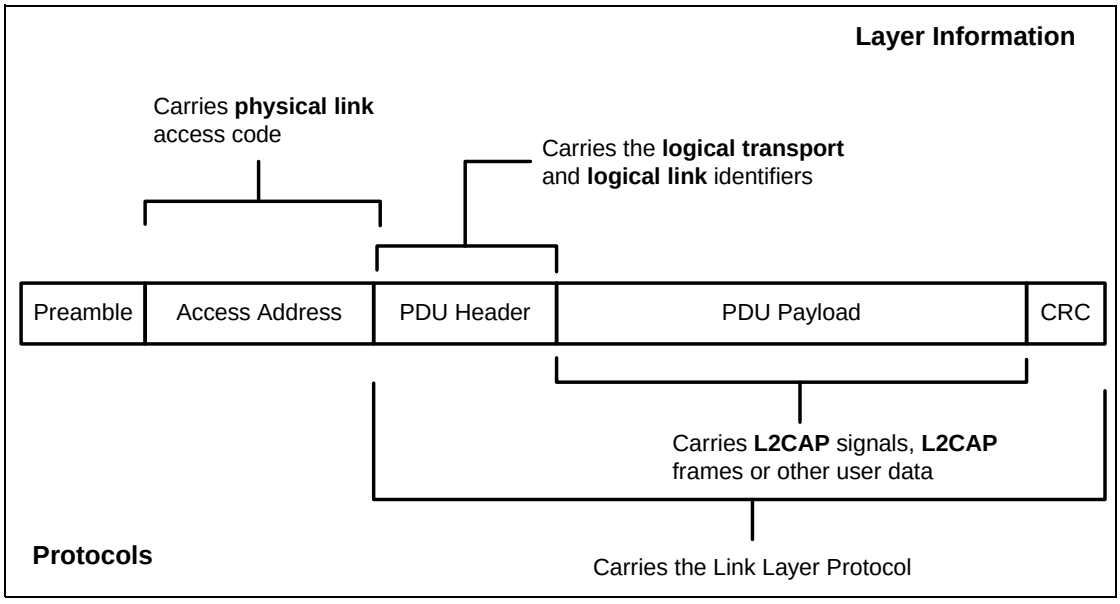


Figure 3.5: LE packet structure

The physical channel identifier is not contained in the link layer air interface packet. The physical channel identifiers are either fixed or are determined at connection setup. All LE packets include the Access Address. This is used to identify communications on a physical link, and to exclude or ignore packets on different physical links that are using the same PHY channels in physical prox-

imity. The Access Address determines whether the packet is directed to the advertising physical link or to an active physical link to a device. All LE advertising physical links use a fixed Access Address. LE active physical links use a randomly generated 32-bit value as their Access Address. This provides an high number of active devices that can be addressed in an LE piconet.

All LE packets include a PDU header. The PDU header determines the type of advertisement broadcast or logical link carried over the physical channel.

For advertising channel PDUs, the PDU header contains the type of advertisement payload, the device address type for addresses contained in the advertisement and advertising channel PDU payload length. Most advertising channel PDU payloads contain the advertiser's address and advertising data. One advertising channel PDU payload only contains the advertiser's device address and the initiator's device address in which the advertisement is directed. Advertising channel PDUs with scan requests payloads contain the scanner's device address and the advertiser's device address. Advertising channel PDUs with scan responses contain advertiser's device address and the scan response data. Advertising channel PDUs with connection request payloads contain the initiator's device address, advertiser's device address and connection setup parameters.

For data channel PDUs, the PDU header contains the Logical Link Identifier (LLID), the Next Expected Sequence Number (NESN), Sequence Number (SN), More Data (MD) and payload length. For data channel PDUs that contain control commands, the data channel PDU payload contains a command opcode and control data that is specific to the command. There is an optional Message Integrity Check (MIC) value that is used to authenticate the data PDU. For data channel PDUs that are data, the data channel PDU payload contains L2CAP data.

3.3 PHYSICAL CHANNELS

The lowest architectural layer in the Bluetooth system is the physical channel. A number of types of physical channel are defined. All Bluetooth physical channels are characterized by a PHY frequency combined with temporal parameters and restricted by spatial considerations. For the basic and adapted piconet physical channels frequency hopping is used to change frequency periodically to reduce the effects of interference and for regulatory reasons.

The Bluetooth BR/EDR system and LE system differ slightly in the way they use physical channels.

3.3.1 BR/EDR Physical Channels

In the BR/EDR core system, peer devices use a shared physical channel for communication. To achieve this their transceivers need to be tuned to the same PHY frequency at the same time, and they need to be within a nominal range of each other.

Given that the number of RF carriers is limited and that many Bluetooth devices may be operating independently within the same spatial and temporal area there is a strong likelihood of two independent Bluetooth devices having their transceivers tuned to the same RF carrier, resulting in a physical channel collision. To mitigate the unwanted effects of this collision each transmission on a physical channel starts with an access code that is used as a correlation code by devices tuned to the physical channel. This channel access code is a property of the physical channel. The access code is present at the start of every transmitted packet.

Four BR/EDR channels are defined. Each is optimized and used for a different purpose. Two of these physical channels (the basic piconet channel and adapted piconet channel) are used for communication between connected devices and are associated with a specific piconet. The remaining BR/EDR physical channels are used for discovering Bluetooth devices and for connecting Bluetooth devices. In BR/EDR this is accomplished through the use of inquiry scan and page scan channels respectively.

A Bluetooth device can only use one BR/EDR physical channel at any given time. In order to support multiple concurrent operations the device uses time-division multiplexing between the channels. In this way a Bluetooth device can appear to operate simultaneously in several piconets, as well as being discoverable and connectable.

Whenever a Bluetooth device is synchronized to the timing, frequency and access code of a physical channel it is said to be ‘connected’ to this channel (whether or not it is actively involved in communications over the channel). The Bluetooth specification assumes that a device is only capable of connecting to one physical channel at any time. Advanced devices may be capable of con-

necting simultaneously to more than one physical channel, but the specification does not assume that this is possible.

3.3.1.1 Basic Piconet Channel

3.3.1.1.1 Overview

The basic piconet channel is used for communication between connected devices during normal operation.

3.3.1.1.2 Characteristics

The basic piconet channel is characterized by a pseudo-random sequence hopping through the PHY 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. All Bluetooth devices participating in the piconet are time- and hop-synchronized to the channel.

The channel is divided into time slots where each slot corresponds to an PHY hop frequency. Consecutive hops correspond to different PHY hop frequencies. The time slots are numbered according to the Bluetooth clock of the piconet master. Packets are transmitted by Bluetooth devices participating in the piconet aligned to start at a slot boundary. Each packet starts with the channel's access code, which is derived from the Bluetooth device address of the piconet.

On the basic piconet channel the master controls access to the channel. The master starts its transmission in even-numbered time slots only. Packets transmitted by the master are aligned with the slot start and define the piconet timing. Packets transmitted by the master may occupy up to five time slots depending on the packet type.

Each master transmission is a packet carrying information on one of the logical transports. Slave devices may transmit on the physical channel in response. The characteristics of the response are defined by the logical transport that is addressed.

For example, on the asynchronous connection-oriented logical transport (ACL), the addressed slave device responds by transmitting a packet containing information for the same logical transport that is nominally aligned with the next (odd-numbered) slot start. Such a packet may occupy up to five time slots, depending on the packet type. On a broadcast logical transport no slaves are allowed to respond.

A special characteristic of the basic piconet physical channel is the use of some reserved slots to transmit a beacon train. The beacon train is only used if the piconet physical channel has parked slaves connected to it. In this situation the master transmits a packet in the reserved beacon train slots (these packets are used by the slave to resynchronize to the piconet physical channel.) The

master may transmit packets from any logical transport onto these slots, providing there is a transmission starting in each of the slots. In the case where there is information from the parked slave broadcast (PSB) logical transport to be transmitted then this is transmitted in the beacon train slots and takes priority over any other logical transport.

3.3.1.1.3 Topology

A basic piconet channel may be shared by any number of Bluetooth devices, limited only by the resources available on the piconet master device. Only one device is the piconet master, all others being piconet slaves. All communication is between the master and slave devices. There is no direct communication between slave devices on the piconet channel.

There is, however, a limitation on the number of logical transports that can be supported within a piconet. This means that although there is no theoretical limit to the number of Bluetooth devices that share a channel there is a limit to the number of these devices that can be actively involved in exchanging data with the master.

3.3.1.1.4 Supported layers

The basic piconet channel supports a number of physical links, logical transports, logical links and L2CAP channels used for general purpose communications.

3.3.1.2 Adapted Piconet Channel

3.3.1.2.1 Overview

The adapted piconet channel differs from the basic piconet channel in two ways. First, the frequency on which a slave transmits is the same as the frequency used by its master in the preceding transmission. In other words the frequency is not recomputed between master and subsequent slave packets. Second, the adapted piconet channel may be based on fewer than the full 79 frequencies. A number of frequencies may be excluded from the hopping pattern by being marked as “unused”. The remainder of the 79 frequencies are included. The two sequences are the same except that whenever the basic pseudo-random hopping sequence selects an unused frequency, it is replaced with an alternative chosen from the used set.

Because the adapted piconet channel uses the same timing and access code as the basic piconet channel, the two channels are often coincident. This provides a deliberate benefit as it allows slaves in either the basic piconet channel or the adapted piconet channel to adjust their synchronization to the master.

The topology and supported layers of the adapted piconet physical channel are identical to the basic piconet physical channel.

3.3.1.3 Inquiry scan channel

3.3.1.3.1 Overview

In order for a device to be discovered, an inquiry scan channel is used. A discoverable device listens for inquiry requests on its inquiry scan channel and then sends a response to that request. In order for a device to discover other devices, it iterates (hops) through all possible inquiry scan channel frequencies in a pseudo-random fashion, sending an inquiry request on each frequency and listening for any response.

3.3.1.3.2 Characteristics

Inquiry scan channels follow a slower hopping pattern and use an access code to distinguish between occasional occupancy of the same radio frequency by two co-located devices using different physical channels.

The access code used on the inquiry scan channel is taken from a reserved set of inquiry access codes that are shared by all Bluetooth devices. One access code is used for general inquiries, and a number of additional access codes are reserved for limited inquiries. Each device has access to a number of different inquiry scan channels. As all of these channels share an identical hopping pattern, a device may concurrently occupy more than one inquiry scan channel if it is capable of concurrently correlating more than one access code.

A device using one of its inquiry scan channels remains passive on that channel until it receives an inquiry message on this channel from another Bluetooth device. This is identified by the appropriate inquiry access code. The inquiry scanning device will then follow the inquiry response procedure to return a response to the inquiring device.

In order for a device to discover other Bluetooth devices it uses the inquiry scan channel to send inquiry requests. As it has no prior knowledge of the devices to discover, it cannot know the exact characteristics of the inquiry scan channel.

The device takes advantage of the fact that inquiry scan channels have a reduced number of hop frequencies and a slower rate of hopping. The inquiring device transmits inquiry requests on each of the inquiry scan hop frequencies and listens for an inquiry response. Transmissions are done at a faster rate, allowing the inquiring device to cover all inquiry scan frequencies in a reasonably short time period.

3.3.1.3.3 Topology

Inquiring and discoverable devices use a simple exchange of packets to fulfill the inquiring function. The topology formed during this transaction is a simple and transient point-to-point connection.

3.3.1.3.4 Supported Layers

During the exchange of packets between an inquiring and discoverable device it may be considered that a temporary physical link exists between these devices. However, the concept is quite irrelevant as it has no physical representation but is only implied by the brief transaction between the devices. No further architectural layers are considered to be supported.

3.3.1.4 Page Scan Channel

3.3.1.4.1 Overview

A connectable device (one that is prepared to accept connections) does so using a page scan channel. A connectable device listens for a page request on its page scan channel and, once received, enters into a sequence of exchanges with this device. In order for a device to connect to another device, it iterates (hops) through all page scan channel frequencies in a pseudo-random fashion, sending a page request on each frequency and listening for a response.

3.3.1.4.2 Characteristics

The page scan channel uses an access code derived from the scanning device's Bluetooth device address to identify communications on the channel. The page scan channel uses a slower hopping rate than the hop rate of the basic and adapted piconet channels. The hop selection algorithm uses the Bluetooth device clock of the scanning device as an input.

A device using its page scan channel remains passive until it receives a page request from another Bluetooth device. This is identified by the page scan channel access code. The two devices will then follow the page procedure to form a connection. Following a successful conclusion of the page procedure both devices switch to the basic piconet channel that is characterized by having the paging device as master.

In order for a device to connect to another Bluetooth device it uses the page scan channel of the target device in order to send page requests. If the paging device does not know the phase of the target device's page scan channel it therefore does not know the current hop frequency of the target device. The paging device transmits page requests on each of the page scan hop frequencies and listens for a page response. This is done at a faster hop rate, allowing the paging device to cover all page scan frequencies in a reasonably short time period.

The paging device may have some knowledge of the target device's Bluetooth clock (indicated during a previous inquiry transaction between the two devices, or as a result of a previous involvement in a piconet with the device), in this case it is able to predict the phase of the target device's page scan channel. It

may use this information to optimize the synchronization of the paging and page scanning process and speed up the formation of the connection.

3.3.1.4.3 Topology

Paging and connectable devices use a simple exchange of packets to fulfill the paging function. The topology formed during this transaction is a simple and transient point-to-point connection.

3.3.1.4.4 Supported Layers

During the exchange of packets between a paging and connectable device it may be considered that a temporary physical link exists between these devices. However, the concept is quite irrelevant as it has no physical representation but is only implied by the brief transaction between the devices. No further architectural layers are considered to be supported.

3.3.2 LE Physical Channels

In the LE core system, two Bluetooth devices use a shared physical channel for communication. To achieve this, their transceivers need to be tuned to the same PHY frequency at the same time, and they need to be within a nominal range of each other.

Given that the number of PHY channels is limited, and many Bluetooth devices can be operating independently within the same spatial and temporal area, there is a strong likelihood of two pairs of independent Bluetooth devices having their transceivers tuned to the same PHY channel, resulting in a physical channel collision. Unlike BR/EDR where an access code is used to identify the piconet, LE uses a randomly generated Access Address to identify a physical link between devices. In the event that two devices happen to share the same PHY channel in the same area, the targeted device Access Address is used as a correlator to determine to which device the communication is directed.

Two LE physical channels are defined. Each is optimized and used for a different purpose. The LE piconet channel is used for communication between connected devices and is associated with a specific piconet. The LE advertisement broadcast channel is used for broadcasting advertisements to LE devices. These advertisements can be used to discover, connect or send user data to scanner or initiator devices.

An LE device can only use one of these LE physical channels at any given time. In order to support multiple concurrent operations the device uses time-division multiplexing between the channels. In this way a Bluetooth device can appear to support connected devices while simultaneously sending advertising broadcasts.

Whenever an LE device is synchronized to the timing and frequency of the physical channel it is said to be connected to this channel (whether or not it is

actively involved in communications over the channel). The Bluetooth specification assumes that a device is only capable of connecting to one physical channel at a time. Advanced devices may be capable of connecting simultaneously to more than one physical channel, but the specification does not make this assumption.

3.3.2.1 LE piconet channel

3.3.2.1.1 Overview

The LE piconet channel is used for communication between connected LE devices during normal operation.

3.3.2.1.2 Characteristics

The LE piconet channel is characterized by a pseudo-random sequence of PHY channels and by three additional parameters provided by the master. The first is the channel map that indicates the set of PHY channels used in the piconet. The second is a pseudo random number used as an index into the complete set of PHY channels. The third is the timing of the first data packet sent by the master after the connection request.

The channel is divided into connection events where each connection event corresponds to a PHY hop channel. Consecutive connection events correspond to different PHY hop channels. The first packet sent by the master after the connection establishment sets an anchor point for the timing of all future connection events. In a connection event the master transmits packets to a slave in the piconet and the slave may or may not respond with a packet of its own.

On the LE piconet channel the master controls access to the channel. The master starts its transmission in a connection event that occurs at regular intervals. Packets transmitted by the master are aligned with the connection event start and define the piconet timing. Packets sent by the master may vary in length from 10 to 47 octets.

Each master transmission contains a packet carrying information on one of the logical transports. Slave devices can transmit on the physical channel in response. The characteristics of the response are defined by the logical transport that is addressed.

The LE piconet channel is similar to the BR/EDR adapted piconet channel in that the set of PHY channels used can be modified to avoid interference. The set of used channels in the channel map is established by the master during connection setup. While in a connection the master can change the channel map when necessary to avoid new interferers.

There are 37 LE piconet channels. The master can reduce this number through the channel map indicating the used channels. When the hopping pattern hits

an unused channel the unused channel is replaced with an alternate from the set of used channels.

3.3.2.1.3 Topology

An LE piconet channel can be shared by any number of LE devices, limited only by the resources available on the piconet master device. Only one device is the piconet master; all others being piconet slaves. All communication is between the master and slave devices. There is no direct communication between slave devices on the piconet channel.

There is virtually no limit on the number of logical transports that can be supported within a piconet. This means that there is no theoretical limit to the number of Bluetooth devices that share a channel with the master.

At present an LE device is only permitted to belong to one piconet at a time.

3.3.2.1.4 Supported layers

The LE piconet channel supports a number of physical links, logical transports, logical links and L2CAP channels used for general purpose communications.

3.3.2.2 Advertisement broadcast channel

3.3.2.2.1 Overview

The LE advertising channel is used to set up connections between two devices or to communicate broadcast information between unconnected devices.

3.3.2.2.2 Characteristics

The LE advertisement broadcast channel is a set of three fixed PHY channels spread evenly across the LE frequency spectrum. The number of advertising PHY channels can be reduced by the advertising device in order to reduce interference. All advertising packets use a fixed Access Address.

The channel is divided into advertising events where each advertising event can hop on all three advertising PHY channels. Consecutive advertising events begin on the first advertising PHY channel. The advertising events occur at regular intervals which are slightly modified with a random delay to aid in interference avoidance.

On the LE advertising broadcast channel the advertising device controls access to the channel. The advertising device starts its transmission in an advertising event and transmits advertising packets on one or more of the three advertising PHY channels. Each advertising packet is sent on a different advertising PHY channel at a fixed interval. Four types of advertising events can be

used, with each advertising event type having different sized advertising packets. These advertising packets can vary in length from 8 to 39 octets.

Some advertising events sent by the advertising device permit the listening device to concurrently send scan requests or connection requests packets on the same advertising PHY channel in which the advertising packet was received. The advertising device can send a scan response packet again on the same advertising PHY channel within the same advertising event. The scan response packet can vary in length from 8 to 39 octets.

3.3.2.2.3 Topology

An LE advertising channel can be shared by any number of LE devices. Any number of LE devices can transmit advertising packets while sharing the same three advertising PHY channels. Any number of scanning devices can listen on the advertising channel. An advertising device can advertise and be connected on an LE piconet channel simultaneously. If the advertising device is connected on an LE piconet channel then restrictions apply to the types of advertising events the device can use. Scanning devices may also be connected to an LE piconet channel simultaneously; however, if the device is connected to an LE piconet channel as a slave, it cannot make connection requests

3.3.3 AMP physical channel

3.3.3.1 Overview

An AMP physical channel is used for communication between connected devices during normal operation. It is used in parallel with an adapted piconet channel between the associated BR/EDR Controllers.

3.3.3.2 Characteristics

The AMP physical channel characteristics depend on the referenced MAC and PHY. See Volume 5 for the referenced MAC and PHY for each PAL.

3.3.3.3 Topology

An AMP physical channel may be shared by any number of Bluetooth devices, limited only by the resources available on the piconet BR/EDR master device and the number of LT_ADDRs available. Roles within the AMP physical channel are not used. All communication is between the BR/EDR piconet master and a single BR/EDR piconet slave. There is no direct communication between BR/EDR piconet slave devices over the AMP physical channel.

Although there is no theoretical limit to the number of Bluetooth devices that can share an AMP physical channel, similar to the basic and adapted piconet channels, there is a limit to the number of these devices that can be actively involved in exchanging data with the master.

3.3.3.4 Supported Layers

The AMP physical channel supports a number of physical links, logical transports, logical links and L2CAP channels used for general purpose communications.

3.4 PHYSICAL LINKS

A physical link represents a baseband connection between Bluetooth devices. A physical link is always associated with exactly one physical channel (although a physical channel may support more than one physical link.) Within the Bluetooth system a physical link is a virtual concept that has no direct representation within the structure of a transmitted packet.

In BR/EDR the access code packet field, together with the clock and address of the master Bluetooth device, is used to identify a physical channel. However, in BR/EDR, there is no subsequent part of the packet that directly identifies the physical link. Instead the physical link may be identified by association with the logical transport, as each logical transport is only received on one physical link.

In LE the hop index and channel map are used to identify a physical channel. The physical link is identified by the access address in the LE packet.

Some physical link types have properties that may be modified. An example of this is the transmit power for the link. Other physical link types have no modifiable properties. In the case of physical links with modifiable properties the LM protocol (BR/EDR) is used to adapt these properties. As the LM protocol (BR/EDR) is supported at a higher layer (by a logical link) the appropriate physical link is identified by implication from the logical link that transports the LM signaling.

In the situation where a transmission is broadcast over a number of different physical links, then the transmission parameters are selected to be suitable for all of the physical links.

3.4.1 BR/EDR Links Supported By The Basic And Adapted Piconet Physical Channel

The basic and adapted piconet physical channels support a physical link which may be active or parked. The physical link is a point-to-point link between the master and a slave. It is always present when the slave is synchronized in the piconet.

3.4.1.1 Active Physical Link

The physical link between a master and a slave device is active if a default ACL logical transport exists between the devices. Active physical links have no direct identification of their own, but are identified by association with the default ACL logical transport ID with which there is a one-to-one correspondence.

An active physical link has the associated property of radio transmit power in each direction. Transmissions from slave devices are always directed over the active physical link to the master, and use the transmit power that is a property of this link in the slave to master direction. Transmissions from the master may be directed over a single active physical link (to a specific slave) or over a number of physical links (to a group of slaves in the piconet.) In the case of point-to-point transmissions the master uses the appropriate transmit power for the physical link in question. (In the case of point-to-multipoint transmissions the master uses a transmit power appropriate for the set of devices addressed.)

Active physical links may be placed into Hold or Sniff mode. The effect of these modes is to modify the periods when the physical link is active and may carry traffic. Logical transports that have defined scheduling characteristics are not affected by these modes and continue according to their pre-defined scheduling behavior. The default ACL logical transport and other links with undefined scheduling characteristics are subject to the mode of the active physical link.

3.4.1.2 Parked Physical Link

The physical link between a master and a slave device is parked when the slave remains synchronized in the piconet, but has no default ACL logical transport. Such a slave is also said to be parked. A beacon train is used to provide regular synchronization to all parked slaves connected to the piconet physical channel. A parked slave broadcast (PSB) logical transport is used to allow communication of a subset of LMP signaling and broadcast L2CAP to parked slaves. The PSB logical transport is closely associated with the beacon train.

A slave is parked (its active link is changed to a parked link) using the park procedure. The master is not allowed to park a slave that has any user created logical transport supported by the physical link. These logical transports are first removed, and any L2CAP channels that are built on these logical transports are also removed. The broadcast logical transport and default ACL logical transports are not considered as user created and are not explicitly removed. When the active link is replaced with a parked link the default ACL logical transport is implicitly removed. The supported logical links and L2CAP channels remain in existence, but become suspended. It is not possible to use these links and L2CAP channels to transport signaling or data while the active link is absent.

A parked slave may become active using the unpark procedure. This procedure is requested by the slave at an access window and initiated by the master. Following the unpark procedure the parked physical link is changed to an active physical link and the default ACL logical transport is re-created. L2CAP channels that were suspended during the most recent park procedure are associated with the new default ACL logical transport and become active again.

Parked links do not support radio power control, as there is no feedback path from parked slaves to the piconet master that can be used to provide the received signal strength at the slave or for the master to measure received signal strength from the slave. Transmissions are carried out at nominal power on parked links.

Parked links use the same physical channel as their associated active link. If a master manages a piconet that contains parked slaves using the basic piconet physical channel and also parked slaves using the adapted piconet physical channel then it must create a parked slave broadcast logical transport (and associated transport) for each of these physical channels.

A parked slave may use the inactive periods of the parked slave broadcast logical transport to save power, or it may carry out activities on other physical channels unrelated to the piconet within which it is parked.

3.4.2 BR/EDR Links Supported by the Scanning Physical Channels

In the case of inquiry scan and page scan channels, the physical link exists for a relatively short time and cannot be controlled or modified in any way. These types of physical links are not further elaborated.

3.4.3 LE Links Supported by the LE Physical Channels

The LE piconet physical channels support an LE active physical link. The physical link is a point-to-point link between the master and a slave. It is always present when the slave is in a connection with the master.

The LE advertising physical channels support an LE advertising physical link. The physical link is a broadcast between the advertiser device and one or more scanner or initiator devices. It is always present when the advertiser is broadcasting advertisement events.

3.4.3.1 Active physical link

The physical link between a master and a slave device is active if a default LE ACL logical transport exists between the devices. Active physical links are identified by the randomly generated Access Address used in the Link Layer packet. Each Access Address has a one-to-one relationship with the master and the slave of the active physical link.

3.4.3.2 Advertising physical link

An advertising physical link between an advertising device and an initiating device for the purposes of forming a connection (active physical link) can exist for a relatively short period of time. These advertising physical links cannot be controlled or modified in any way and these types of physical links are not further elaborated.

An advertising physical link between an advertising device and a scanning device used for periodic broadcasting of user data can exist for longer periods of time. There is no identification information about the physical link within the protocol. The relationship between the advertising and scanning device shall be established through the use of the Bluetooth device address.

3.4.4 Links Supported by the AMP Physical Channels

An AMP physical link is always associated with exactly one AMP physical channel (although an AMP physical channel may support more than one AMP physical link). The AMP physical link is a point-to-point link between two PALs. The characteristics of an AMP physical link depend on the underlying AMP technology.

3.5 LOGICAL LINKS AND LOGICAL TRANSPORTS

A variety of logical links are available to support different application data transport requirements. Each logical link is associated with a logical transport, which has a number of characteristics. These characteristics include flow control, acknowledgement/repeat mechanisms, sequence numbering and scheduling behavior. Logical transports are able to carry different types of logical links (depending on the type of the logical transport). In the case of some of the Bluetooth logical links these are multiplexed onto the same logical transport. Logical transports may be carried by active physical links on either the basic or the adapted piconet physical channel.

Logical transport identification and real-time (link control) signaling are carried in the packet header, and for some logical links identification is carried in the payload header. Control signaling that does not require single slot response times is carried out using the LMP protocol.

Table 3.1 on page 61 lists all of the logical transport types, the supported logical link types, which type of physical links and physical channels can support them, and a brief description of the purpose of the logical transport.

Logical transport	Links supported	Supported by	Bearer	Overview
Asynchronous Connection-Oriented (ACL)	Control (LMP) ACL-C or (PAL) AMP-C User (L2CAP) ACL-U or AMP-U	BR/EDR active physical link, BR/EDR basic or adapted piconet physical channel, AMP physical link, AMP physical channel.	BR/EDR, AMP	Reliable or time-bounded, bi-directional, point-to-point.

Table 3.1: Logical transport types.

Logical transport	Links supported	Supported by	Bearer	Overview
Synchronous Connection-Oriented (SCO)	Stream (unframed) SCO-S	BR/EDR active physical link, BR/EDR basic or adapted piconet physical channel	BR/EDR	Bi-directional, symmetric, point-to-point, AV channels. Used for 64Kb/s constant rate data.
Extended Synchronous Connection-Oriented (eSCO)	Stream (unframed) eSCO-S	BR/EDR active physical link, BR/EDR basic or adapted piconet physical channel	BR/EDR	Bi-directional, symmetric or asymmetric, point-to-point, general regular data, limited retransmission. Used for constant rate data synchronized to the master Bluetooth clock.
Active Slave Broadcast (ASB)	User (L2CAP) ASB-U	BR/EDR active physical link, basic or adapted physical channel.	BR/EDR	Unreliable, uni-directional broadcast to any devices synchronized with the physical channel. Used for broadcast L2CAP groups.
Parked Slave Broadcast (PSB)	Control (LMP) PSB-C, User (L2CAP) PSB-U	BR/EDR parked physical link, basic or adapted piconet physical channel.	BR/EDR	Unreliable, uni-directional broadcast to all piconet devices. Used for LMP and L2CAP traffic to parked devices, and for access requests from parked devices.
LE asynchronous connection (LE ACL	Control (LL) LE-C, User (L2CAP) LE-U	LE active physical link, LE piconet physical channel.	LE	Reliable or time-bounded, bi-directional, point-to-point.
LE Advertising Broadcast (ADVB)	Control (LL) ADVB-C, User (LL) ADVB-U	LE advertising physical link, LE piconet physical channel.	LE	LE advertising physical link, LE piconet physical channel.

Table 3.1: Logical transport types.

The classification of each link type follows from a selection procedure within three categories.

3.5.1 Casting

The first category is that of casting. This may be either unicast or broadcast.

- *Unicast links.* Unicast links exist between exactly two endpoints. Traffic may be sent in either direction on unicast links.
- *Broadcast links.* Broadcast links exist between one source device and zero or more receiver devices. Traffic is unidirectional, i e., only sent from the source devices to the receiver devices. Broadcast links are connectionless, meaning there is no procedure to create these links, and data may be sent over them at any time. Broadcast links are unreliable, and there is no guarantee that the data will be received.

3.5.2 Scheduling and Acknowledgement Scheme

The second category relates to the scheduling and acknowledgement scheme of the link, and implies the type of traffic that is supported by the link. These are synchronous, isochronous or asynchronous. There are no specific isochronous links defined, though the default ACL link can be configured to operate in this fashion.

- *Synchronous links.* Synchronous links provide a method of associating the transported data with the Bluetooth piconet clock. This is achieved by reserving regular slots on the physical channel, and transmitting fixed size packets at these regular intervals. Such links are suitable for constant rate isochronous data.
- *Asynchronous links.* Asynchronous links provide a method for transporting data that has no time-based characteristics. The data is normally expected to be retransmitted until successfully received, and each data entity can be processed at any time after receipt, without reference to the time of receipt of any previous or successive entity in the stream (providing the ordering of data entities is preserved.)
- *Isochronous links.* Isochronous links provide a method for transporting data that has time-based characteristics. The data may be retransmitted until received or expired. The data rate on the link need not be constant (this being the main difference from synchronous links.)

3.5.3 Class of Data

The final category is related to the class of data that is carried by the link. This is either control (LMP or PAL) data or user data. The user data category is subdivided into L2CAP (or framed) data and stream (or unframed) data.

- *Control links.* Control links are only used for transporting LMP messages between two link managers. These links are invisible above the baseband layer, and cannot be directly instantiated, configured or released by applications, other than by the use of the connection and disconnection services that have this effect implicitly. Control links are always multiplexed with an equivalent L2CAP link onto an ACL logical transport. Subject to the rules defining the ARQ scheme, the control link traffic always takes priority over the L2CAP link traffic.

- *L2CAP links.* L2CAP links are used to transport L2CAP PDUs, which may carry the L2CAP signaling channel (on the default ACL-U logical link only) or framed user data submitted to user-instantiated L2CAP channels. L2CAP frames submitted to the baseband may be larger than the available baseband packets. A link control protocol embedded within the LLID field preserves the frame-start and frame-continuation semantics when the frame is transmitted in a number of fragments to the receiver.
- *Stream links.* Stream links are used to transport user data that has no inherent framing that should be preserved when delivering the data. Lost data may be replaced by padding at the receiver.

3.5.4 Logical Transports

3.5.4.1 BR/EDR Asynchronous Connection-oriented (ACL)

The asynchronous connection-oriented (ACL) logical transport is used to carry LMP and L2CAP control signaling and best effort asynchronous user data. The ACL logical transport uses a 1-bit ARQN/SEQN scheme to provide simple channel reliability. Every active slave device within a piconet has one ACL logical transport to the piconet master, known as the default ACL.

The default ACL is created between the master and the slave when a device joins a piconet (connects to the basic piconet physical channel). This default ACL is assigned a logical transport address (LT_ADDR) by the piconet master. This LT_ADDR is also used to identify the active physical link when required (or as a piconet active member identifier, effectively for the same purpose.)

The LT_ADDR for the default ACL is reused for synchronous connection-oriented logical transports between the same master and slave. (This is for reasons of compatibility with earlier Bluetooth specifications.) Thus the LT_ADDR is not sufficient on its own to identify the default ACL. However the packet types used on the ACL are different from those used on the synchronous connection-oriented logical transport. Therefore, the ACL logical transport can be identified by the LT_ADDR field in the packet header in combination with the packet type field.

The default ACL may be used for isochronous data transport by configuring it to automatically flush packets after the packets have expired. Asynchronous traffic can be sent over an ACL logical transport configured for isochronous traffic by marking the asynchronous packets as non-automatically-flushable. This allows both isochronous and asynchronous traffic to be transferred at the same time to a single device.

If the default ACL is removed from the active physical link then all other logical transports that exist between the master and the slave are also removed. In the case of unexpected loss of synchronization to the piconet physical channel the physical link and all logical transports and logical links cease to exist at the time that this synchronization loss is detected.

A device may remove its default ACL (and by implication its active physical link) but remain synchronized to the piconet. This procedure is known as parking, and a device that is synchronized to the piconet, but has no active physical link is parked within that piconet.

When the device transitions to the parked state the default ACL logical links that are transported on the default ACL logical transport remain in existence, but become suspended. No data may be transferred across a suspended logical link. When the device transitions from the parked state back into connection state, a new default ACL logical transport is created (it may have a different LT_ADDR from the previous one) and the suspended logical links are attached to this default ACL and become active once again.

3.5.4.2 BR/EDR Synchronous Connection-Oriented (SCO)

The synchronous connection-oriented (SCO) logical transport is a symmetric, point-to-point transport between the master and a specific slave. The SCO logical transport reserves slots on the physical channel and can therefore be considered as a circuit-switched connection between the master and the slave. SCO logical transports carry 64 kb/s of information synchronized with the piconet clock. Typically this information is an encoded voice stream. Three different SCO configurations exist, offering a balance between robustness, delay and bandwidth consumption.

Each SCO-S logical link is supported by a single SCO logical transport, which is assigned the same LT_ADDR as the default ACL logical transport between the devices. Therefore the LT_ADDR field is not sufficient to identify the destination of a received packet. Because the SCO links use reserved slots, a device uses a combination of the LT_ADDR, the slot numbers (a property of the physical channel) and the packet type to identify transmissions on the SCO link.

Although slots are reserved for the SCO, it is permissible to use a reserved slot for traffic from another channel that has a higher priority. This may be required as a result of QoS commitments, or to send LMP signaling on the default ACL when the physical channel bandwidth is fully occupied by SCOs. As SCOs carry different packet types to ACLs, the packet type is used to identify SCO traffic (in addition to the slot number and LT_ADDR.)

There are no further architectural layers defined by the Bluetooth core specification that are transported over an SCO link. A number of standard formats are defined for the 64 kb/s stream that is transported, or an unformatted stream is allowed where the application is responsible for interpreting the encoding of the stream.

3.5.4.3 BR/EDR Extended Synchronous Connection-Oriented (eSCO)

The extended synchronous connection-oriented (eSCO) logical transport is a symmetric or asymmetric, point-to-point transport between the master and a

specific slave. The eSCO reserves slots on the physical channel and can therefore be considered as a circuit-switched connection between the master and the slave. eSCO links offer a number of extensions over the standard SCO links, in that they support a more flexible combination of packet types and selectable data contents in the packets and selectable slot periods, allowing a range of synchronous bit rates to be supported.

eSCO links also can offer limited retransmission of packets (unlike SCO links where there is no retransmission.) If these retransmissions are required they take place in the slots that follow the reserved slots, otherwise the slots may be used for other traffic.

Each eSCO-S logical link is supported by a single eSCO logical transport, identified by a LT_ADDR that is unique within the piconet for the duration of the eSCO. eSCO-S links are created using LM signaling and follow scheduling rules similar to SCO-S links.

There are no further architectural layers defined by the Bluetooth core specification that are transported over an eSCO-S link. Instead applications may use the data stream for whatever purpose they require, subject to the transport characteristics of the stream being suitable for the data being transported.

3.5.4.4 BR/EDR Active Slave Broadcast (ASB)

The active slave broadcast logical transport is used to transport L2CAP user traffic to all devices in the piconet that are currently connected to the physical channel that is used by the ASB. There is no acknowledgement protocol and the traffic is uni-directional from the piconet master to the slaves. The ASB channel may be used for L2CAP group traffic (a legacy of the 1.1 specification), and is never used for L2CAP connection-oriented channels, L2CAP control signaling or LMP control signaling.

The ASB logical transport is inherently unreliable because of the lack of acknowledgement. To improve the reliability, each packet is transmitted a number of times. An identical sequence number is used to assist with filtering retransmissions at the slave device.

The ASB logical transport is identified by a reserved LT_ADDR. (The reserved LT_ADDR address is also used by the PSB logical transport.) An active slave will receive traffic on both logical transports, and cannot readily distinguish between them. As the ASB logical transport does not carry LMP traffic an active slave can ignore packets received over the LMP logical link on the ASB logical transport. However L2CAP traffic transmitted over the PSB logical transport is also received by active slaves on the ASB logical transport and cannot be distinguished from L2CAP traffic sent on the ASB transport.

An ASB is implicitly created whenever a piconet exists, and there is always one ASB associated with each of the basic and adapted piconet physical channels that exist within the piconet. Because the basic and adapted piconet physical

channels are mostly coincident, a slave device cannot distinguish which of the ASB channels is being used to transmit the packets. This adds to the general unreliability of the ASB channel. (Although it is, perhaps, no more unreliable than general missed packets.)

A master device may decide to use only one of its two possible ASBs (when it has both a basic and adapted piconet physical channel), as with sufficient retransmissions it is possible to address both groups of slaves on the same ASB channel.

The ASB channel is never used to carry LMP or L2CAP control signals.

3.5.4.5 BR/EDR Parked Slave Broadcast (PSB)

The parked slave broadcast logical transport is used for communications between the master and slaves that are parked (have given up their default ACL logical transport.) The parked slave broadcast link is the only logical transport that exists between the piconet master and parked slaves.

The PSB logical transport is more complex than the other logical transports as it consists of a number of phases, each having a different purpose. These phases are the control information phase (used to carry the LMP logical link), the user information phase (used to carry the L2CAP logical link), and the access phase (carrying baseband signaling.) The control information and broadcast information phases are usually mutually exclusive as only one of them can be supported in a single beacon interval. (Even if there is no control or user information phase, the master is still required to transmit a packet in the beacon slots so that the parked slaves can resynchronize.) The access phase is normally present unless cancelled in a control information message.

The control information phase is used for the master to send information to the parked slaves containing modifications to the PSB transport attributes, modifications to the beacon train attributes, or a request for a parked slave to become active in the piconet (known as unparking). This control information is carried in LMP messages on the LMP logical link. (The control information phase is also present in the case of a user information phase where the user information requires more than one baseband packet.)

Packets in the control information phase are always transmitted in the physical channel beacon train slots, and cannot be transmitted on any other slots. The control information occupies a single DM1 packet and is repeated in every beacon train slot within a single beacon interval. (If there is no control information then there may be a user information phase that uses the beacon slots. If neither phase is used then the beacon slots are used for other logical transport traffic or for NULL packets.)

The user information phase is used for the master to send L2CAP packets that are destined for all piconet slaves. User information may occupy one or more

baseband packets. If the user information occupies a single packet then the user information packet is repeated in each of the piconet channel beacon train slots.

If the user information occupies more than one baseband packet then it is transmitted in slots after the beacon train (the broadcast scan window) and the beacon slots are used to transmit a control information phase message that contains the timing attributes of this broadcast scan window. This is required so that the parked slaves remain connected to the piconet physical channel to receive the user information.

The access phase is normally present unless temporarily cancelled by a control message carried in the control information broadcast phase. The access window consists of a sequence of slots that follow the beacon train. In order for a parked slave to become active in the piconet, it may send such an access request to the piconet master during the access window. Each parked slave is allocated an access request address (not necessarily unique) that controls when during the access window the slave requests access.

The PSB logical transport is identified by the reserved LT_ADDR of 0. This reserved LT_ADDR address is also used by the ASB logical transport. Parked slaves are not normally confused by the duplicated use of the LT_ADDR as they are only connected to the piconet physical channel during the time that the PSB transport is being used.

3.5.4.6 LE Asynchronous Connection (LE ACL)

The LE asynchronous connection (LE ACL) logical transport is used to carry LL and L2CAP control signaling and best effort asynchronous user data. The LE ACL logical transport uses a 2-bit NESN/SN scheme to provide simple channel reliability. Every active slave device within an LE piconet has one LE ACL logical transport to the piconet master, known as the default LE ACL.

The default LE ACL is automatically created between the master and the slave when a device joins a piconet (connects to the LE piconet physical channel). This default LE ACL is assigned an access channel number by the piconet master. This access address is also used to identify the active physical link when required.

If the default LE ACL is removed from the LE active physical link then all other LE logical transports that exist between the master and the slave are also removed. In the case of unexpected loss of synchronization to the LE piconet physical channel the LE physical link and all LE logical transports and LE logical links cease to exist at the time that this synchronization loss is detected.

3.5.4.7 LE Advertising Broadcast (ADVB)

The LE advertising broadcast logical transport is used to transport broadcast control and user data to all scanning devices in a given area. There is no acknowledgement protocol and the traffic is predominately unidirectional from

the advertising device. A scanning device can send requests over the logical transport to get additional broadcast user data, or to form an LE ACL logical transport connection. The LE Advertising Broadcast logical transport data is carried only over the LE advertising broadcast link.

The ADVB logical transport is inherently unreliable because of the lack of acknowledgement. To improve the reliability, each packet is transmitted a number of times over the LE advertising broadcast link.

An ADVB is created whenever an advertising device begins advertising. The ADVB logical transport is identified by the advertiser's Bluetooth device address.

3.5.5 Logical Links

Some logical transports are capable of supporting different logical links, either concurrently multiplexed, or one of the choice.

3.5.5.1 BR/EDR Logical Links

Within BR/EDR logical transports, the logical link is identified by the logical link identifier (LLID) bits in the payload header of baseband packets that carry a data payload. The logical links distinguish between a limited set of core protocols that are able to transmit and receive data on the logical transports. Not all of the logical transports are able to carry all of the logical links (the supported mapping is shown in Figure 3.2 on page 40.) In particular the BR/EDR SCO and eSCO logical transports are only able to carry constant data rate streams, and these are uniquely identified by the LT_ADDR. Such logical transports only use packets that do not contain a payload header, as their length is known in advance, and no LLID is necessary.

3.5.5.1.1 ACL Control Logical Link (ACL-C)

The ACL Control Logical Link (ACL-C) is used to carry BR/EDR LMP signaling between devices in the piconet. The control link is only carried on the default ACL logical transport and on the PSB logical transport (in the control information phase). The ACL-C link is always given priority over the ACL-U (see below) link when carried on the same logical transport.

3.5.5.1.2 User Asynchronous/Isochronous Logical Link (ACL-U)

The user asynchronous/isochronous logical link (ACL-U) is used to carry all asynchronous and isochronous framed user data. The ACL-U link is carried on all but the synchronous logical transports. Packets on the ACL-U link are identified by one of two reserved LLID values. One value is used to indicate whether the baseband packet contains the start of an L2CAP frame and the other indicates a continuation of a previous frame. This ensures correct synchronization of the L2CAP reassembler following flushed packets. The use of

this technique removes the need for a more complex L2CAP header in every baseband packet (the header is only required in the L2CAP start packets), but adds the requirement that a complete L2CAP frame shall be transmitted before a new one is transmitted. (An exception to this rule being the ability to flush a partially transmitted L2CAP frame in favor of another L2CAP frame.)

3.5.5.1.3 User Synchronous/Extended Synchronous Logical Links (SCO-S/ eSCO-S)

Synchronous (SCO-S) and extended synchronous (eSCO-S) logical links are used to support isochronous data delivered in a stream without framing. These links are associated with a single logical transport, where data is delivered in constant sized units at a constant rate. There is no LLID within the packets on these transports, as only a single logical link can be supported, and the packet length and scheduling period are pre-defined and remain fixed during the life-time of the link.

Variable rate isochronous data cannot be carried by the SCO-S or eSCO-S logical links. In this case the data must be carried on ACL-U logical links, which use packets with a payload header.

3.5.5.2 LE Logical Links

Within LE logical transports, the logical link is identified by the logical link identifier (LLID) bits in the payload header of baseband packets that carry a data payload. The logical links distinguish between a limited set of core protocols that are able to transmit and receive data on the logical transports. Not all of the logical transports are able to carry all of the logical links (the supported mapping is shown in Figure 3.2 on page 40).

3.5.5.2.1 Control Logical Link (LE-C)

The LE ACL Control Logical Link (LE-C) is used to carry LE LL signaling between devices in the piconet. The control link is only carried on the default LE ACL logical transport.

3.5.5.2.2 User Asynchronous Logical Link (LE-U)

The user asynchronous logical link (LE-U) is used to carry all asynchronous and framed user data. The LE-U link is carried on the LE logical transport. Packets on the LE-U link are identified by one of two reserved LLID values. One value is used to indicate whether the baseband packet contains the start of an L2CAP frame and the other indicates a continuation of a previous frame or empty PDU. This ensures correct synchronization of the L2CAP re-assembler. The use of this technique removes the need for a more complex L2CAP header in every baseband packet, but adds the requirement that a complete L2CAP frame shall be transmitted before a new one is transmitted.

3.5.5.2.3 Advertising Broadcast Control Logical Link (ADVB-C)

The LE Advertising Broadcast Control Logical Link (ADVB-C) is used to carry LE LL signaling between unconnected devices in a given area. This signaling is the control commands for gathering additional broadcast user data (scan requests) or connection requests. The control link is only carried on the default LE Advertising Broadcast logical transport.

3.5.5.2.4 Advertising Broadcast User Data Logical Link (ADVB-U)

The LE Advertising Broadcast User Data Logical Link (ADVB-U) is used to carry LE advertisement broadcast user data used between devices without the need for a connection or LE-U between the devices. The user data link is only carried on the default LE Advertising Broadcast logical transport.

3.5.5.3 AMP Logical Links

3.5.5.3.1 AMP Control Logical Link (AMP-C)

Each PAL functions differently with respect to a control logical link. When one exists, this logical link is referred to as the AMP-C logical link.

3.5.5.3.2 AMP User Asynchronous/Isochronous Logical Link (AMP-U)

The AMP user asynchronous/isochronous logical link (AMP-U) is used to carry all asynchronous and isochronous framed user data over an AMP. Unlike the ACL-U logical link, the AMP-U supports multiple logical link handles per physical link and each logical link handle may have different quality of service capabilities. The details of how the AMP-U logical link is mapped to the underlying MAC/PHY is described by the PAL.

3.6 L2CAP CHANNELS

L2CAP provides a multiplexing role allowing many different applications to share an ACL-U or AMP-U logical link. Applications and service protocols interface with L2CAP using a channel-oriented interface to create connections to equivalent entities on other devices.

L2CAP channel endpoints are identified to their clients by a Channel Identifier (CID). This is assigned by L2CAP, and each L2CAP channel endpoint on any device has a different CID.

L2CAP channels may be configured to provide an appropriate Quality of Service (QoS) to the application. L2CAP maps the channel onto the ACL-U logical link, LE-U, or one of the AMP-U logical links.

L2CAP supports channels that are connection-oriented and others that are group-oriented. Group-oriented channels may be mapped onto the ASB-U logical link, or implemented as iterated transmission to each member in turn over an ACL-U logical link.

Apart from the creation, configuration and dismantling of channels, the main role of L2CAP is to multiplex service data units (SDUs) from the channel clients onto the ACL-U, LE-U, or AMP-U logical links, and to carry out a simple level of scheduling, selecting SDUs according to relative priority.

L2CAP can provide per channel flow control with the peer L2CAP layer. This option is selected by the application when the channel is established. L2CAP can also provide enhanced error detection and retransmission to (a) reduce the probability of undetected errors being passed to the application and (b) recover from loss of portions of the user data when the Baseband layer performs a flush on the ACL-U logical link.

In the case where an HCI is present, the L2CAP is also required to segment L2CAP SDUs into fragments that will fit into the baseband buffers, and also to operate a token based flow control procedure over the HCI, submitting fragments to the baseband only when allowed to do so. This may affect the scheduling algorithm.

4 COMMUNICATION TOPOLOGY AND OPERATION

4.1 PICONET TOPOLOGY

4.1.1 BR/EDR Topology

Any time a link is created using the BR/EDR Controller it is within the context of a piconet. A piconet consists of two or more devices that occupy the same BR/EDR physical channel.

Connected BR/EDR devices communicate on the same physical channel by synchronizing with a common clock and hopping sequence. The common (piconet) clock is identical to the Bluetooth clock of one of the devices in the piconet, known as the master of the piconet, and the hopping sequence is derived from the master's clock and the master's Bluetooth device address. All other synchronized devices are referred to as slaves in the piconet.

The terms master and slave are only used when describing these roles in a piconet.

A number of independent piconets may exist in close proximity. Each piconet has a different physical channel (that is a different master device and an independent timing and hopping sequence.)

A Bluetooth device may participate concurrently in two or more piconets. It does this on a time-division multiplexing basis. A Bluetooth device can never be a master of more than one piconet. (Since in BR/EDR the piconet is defined by synchronization to the master's Bluetooth clock it is impossible to be the master of two or more piconets.) A Bluetooth device may be a slave in many independent piconets.

A Bluetooth device that is a member of two or more piconets is said to be involved in a scatternet. Involvement in a scatternet does not necessarily imply any network routing capability or function in the Bluetooth device. The Bluetooth core protocols do not, and are not intended to offer such functionality, which is the responsibility of higher level protocols and is outside the scope of the Bluetooth core specification.

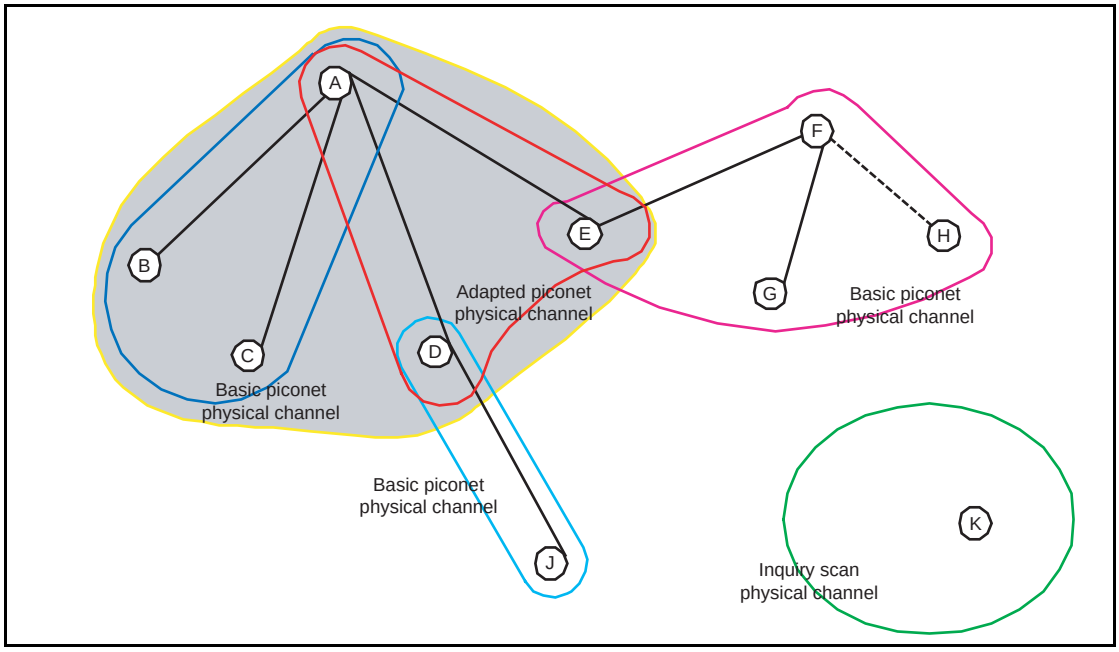


Figure 4.1: Example Bluetooth BR/EDR topology

In Figure 4.1 on page 74 an example topology is shown that demonstrates a number of the architectural features described below. Device A is a master in a piconet (represented by the shaded area, and known as piconet A) with devices B, C, D and E as slaves. Two other piconets are shown: a) one piconet with device F as master (known as piconet F) and devices E, G and H as slaves and b) one piconet with device D as master (known as piconet D) and device J as slave.

In piconet A there are two physical channels. Devices B and C are using the basic piconet physical channel (represented by the blue enclosure) as they do not support adaptive frequency hopping. Devices D and E are capable of supporting adaptive frequency hopping, and are using the adapted piconet physical channel (represented by the red enclosure.) Device A is capable of adaptive frequency hopping, and operates in a TDM basis on both physical channels according to which slave is being addressed.

Piconet D and piconet F are both using only a basic piconet physical channel (represented by the cyan and magenta enclosures respectively.) In the case of piconet D this is because device J does not support the adaptive hopping mode. Although device D supports adaptive hopping it cannot use it in this piconet. In piconet F device F does not support adaptive hopping, and therefore it cannot be used in this piconet.

Device K is shown in the same locality as the other devices. It is not currently a member of a piconet, but has services that it offers to other Bluetooth devices. It is currently listening on its inquiry scan physical channel (represented by the green enclosure), awaiting an inquiry request from another device.

4.1.2 LE Topology

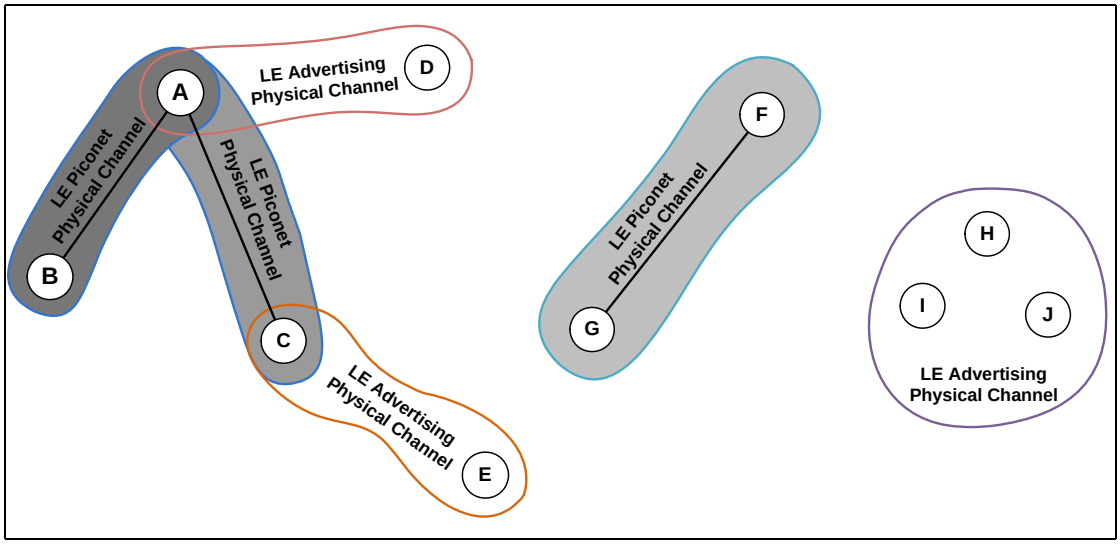


Figure 4.2: Example of Bluetooth LE topology

In Figure 4.2 on page 75 an example topology is shown that demonstrates a number of the LE architectural features described below. Device A is a master in a piconet (represented by the shaded area, and known as piconet A) with devices B and C as slaves, unlike BR/EDR the slaves do not share a common physical channel with the master. Each slave communicates on a separate physical channel with the master. One other piconet is shown with device F as master (known as piconet F) and device G as a slave. There are three other broadcast groups shown: a) device D is an advertiser and device A is also an initiator (known as broadcast group D), b) device E is a scanner and device C is also an advertiser (known as broadcast group C), c) device H is an advertiser and devices I and J are scanners (known as broadcast group H).

Devices A and B are using one LE physical channel (represented by the blue enclosure and a dark gray background). Devices B and C are using another LE piconet physical channel (represented by the blue enclosure and a lighter gray background). In broadcast group D, device D is advertising using a connectable event on the advertising physical channel (represented by the green enclosure) and device A is an initiator. Device A can form a connection with device D and add the device to piconet A. In broadcast group E, device C is also advertising on the advertising physical channel (represented by the orange enclosure) using non-connectable events which are being captured by device E as a scanner. Broadcast group D and broadcast group E may be using different advertising PHY channels or different timings to avoid collisions.

In piconet F there is one physical channel. Devices F and G are using an LE piconet physical channel (represented by the aqua enclosure). Device F is the master and device G is the slave.

In broadcast group H there is one physical channel. Devices H, I and J are using an LE advertising physical channel (represented by the purple enclosure). Devices H and I are advertisers and device J is a scanner.

4.2 OPERATIONAL PROCEDURES AND MODES

The typical operational mode of a Bluetooth device is to be connected to other Bluetooth devices (in a piconet) and exchanging data with those Bluetooth devices. As Bluetooth is an ad-hoc wireless communications technology, there are a number of operational procedures that enable piconets to be formed so that the subsequent communications can take place. Procedures and modes are applied at different layers in the architecture and therefore a device may be engaged in a number of these procedures and modes concurrently.

4.2.1 BR/EDR Procedures

4.2.1.1 Inquiry (Discovering) Procedure

Bluetooth devices use the inquiry procedure to discover nearby devices, or to be discovered by devices in their locality.

The inquiry procedure is asymmetrical. A Bluetooth device that tries to find other nearby devices is known as an inquiring device and actively sends inquiry requests. Bluetooth devices that are available to be found are known as discoverable devices and listen for these inquiry requests and send responses. The inquiry procedure uses a special physical channel for the inquiry requests and responses.

Both inquiring and discoverable devices may already be connected to other Bluetooth devices in a piconet. Any time spent inquiring or occupying the inquiry scan physical channel needs to be balanced with the demands of the QoS commitments on existing logical transports.

The inquiry procedure does not make use of any of the architectural layers above the physical channel, although a transient physical link may be considered to be present during the exchange of inquiry and inquiry response information.

4.2.1.1.1 Extended Inquiry Response

An Extended Inquiry Response can be used to provide miscellaneous information during the inquiry response procedure. Data types are defined for such things as local name and supported services, information that otherwise would have to be obtained by establishing a connection. A device that receives a local name and a list of supported services in an extended inquiry response does not have to connect to do a remote name request and an SDP service search, thereby shortening the time to useful information. It is recommended that a device includes all supported services and a significant portion of its local

name, if that name is too long to be sent in its entirety, in the extended inquiry response.

The extended inquiry response procedure is backwards compatible with the standard inquiry response procedure.

4.2.1.2 Paging (Connecting) Procedure

The procedure for forming connections is asymmetrical and requires that one Bluetooth device carries out the page (connection) procedure while the other Bluetooth device is connectable (page scanning.) The procedure is targeted, so that the page procedure is only responded to by one specified Bluetooth device.

The connectable device uses a special physical channel to listen for connection request packets from the paging (connecting) device. This physical channel has attributes that are specific to the connectable device, hence only a paging device with knowledge of the connectable device is able to communicate on this channel.

Both paging and connectable devices may already be connected to other Bluetooth devices. Any time spent paging or occupying the page scan physical channel needs to be balanced with the demands of the QoS commitments on existing logical transports.

4.2.1.3 Connected Mode

After a successful connection procedure over the BR/EDR Controller, there is a piconet physical channel to which both devices are connected, there is a physical link between the devices, and there are default ACL-C and ACL-U logical links. When in the connected mode it is possible to create and release additional logical links, and to change the modes of the physical and logical links while remaining connected to the piconet physical channel. It is also possible for the device to carry out inquiry, paging or scanning procedures or to be connected to other piconets without needing to disconnect from the original piconet physical channel.

Additional logical links are created using the Link Manager that exchanges Link Manager Protocol messages with the remote Bluetooth device to negotiate the creation and settings for these links. One of these links (ACL-C) transports the LMP control protocol and is invisible to the layers above the Link Manager. The other link (ACL-U) transports the L2CAP signaling protocol and any multiplexed L2CAP best-effort channels. It is common to refer to a default ACL logical transport, which can be resolved by context, but typically refers to the default ACL-U logical link. Also note that these two logical links share a logical transport.

AMP physical links may be established in connected mode. Once an AMP physical link is created, one or more AMP-U logical links may be established to transport L2CAP user data.

During the time that a slave device is actively connected to a piconet there is always a default ACL logical transport between the slave and the master device. There are two methods of deleting the default ACL logical transport. The first method is to detach the device from the piconet physical channel, at which time the entire hierarchy of L2CAP channels, logical links, and logical transports between the devices is deleted.

The second method is to place the physical link to the slave device in the park state, at which time it gives up its default ACL logical transport. This is only allowed if all other logical transports have been deleted (except for the ASB logical transport that cannot be explicitly created or deleted.) It is not allowed to park a device while it has any logical transports other than the default ACL and ASB logical transports.

When the slave device physical link is parked, its default ACL logical transport is released and the ASB logical transport is replaced with a PSB logical transport. The ACL-C and ACL-U logical links that are multiplexed onto the default ACL logical transport remain in existence but cannot be used for the transport of data. The Link Manager on the master device restricts itself to the use of LMP messages that are allowed to be transported over the PSB-C logical link. The Channel Manager and L2CAP Resource Manager ensure that no L2CAP unicast data traffic is submitted to the controller while the device is parked. The Channel Manager may decide to manage the parking and unparking of the device as necessary to allow data to be transported.

4.2.1.4 Hold Mode

Hold mode is not a general device mode, but applies to unreserved slots on the physical link. When in this mode, the physical link is only active during slots that are reserved for the operation of the synchronous link types SCO and eSCO. All asynchronous links are inactive. Hold modes operate once for each invocation and are then exited when complete, returning to the previous mode.

4.2.1.5 Sniff Mode

Sniff mode is not a general device mode, but applies to the default ACL logical transports. When in this mode the availability of these logical transports is modified by defining a duty cycle consisting of periods of presence and absence. Devices that have their default ACL logical transports in sniff mode may use the absent periods to engage in activity on another physical channel, or to enter reduced power mode. Sniff mode only affects the default ACL logical transports (i.e. their shared ACL logical transport), and does not apply to any additional SCO or eSCO logical transports that may be active. The periods of presence and absence of the physical link on the piconet physical channel is derived as a union of all logical transports that are built on the physical link.

Sniff subrating provides a mechanism for further reducing the active duty cycle, thereby enhancing the power-saving capability of sniff mode. Sniff subrating allows a Host to create a guaranteed access-like connection by specifying maximum transmit and receive latencies. This allows the basebands to optimize the low power performance without having to exit and re-enter sniff mode using Link Manager commands.

Note that broadcast logical transports have no defined expectations for presence or absence. A master device should aim to schedule broadcasts to coincide with periods of physical link presence within the piconet physical channel, but this may not always be possible or practical. Repetition of broadcasts is defined to improve the possibilities for reaching multiple slaves without overlapping presence periods. However, broadcast logical transports cannot be considered to be reliable.

4.2.1.6 Parked State

A slave device may remain connected to a piconet but have its physical link in the parked state. In this state the device cannot support any logical links to the master with the exception of the PSB-C and PSB-U logical links that are used for all communication between the piconet master and the parked slave.

When the physical link to a slave device is parked this means that there are restrictions on when the master and slave may communicate, defined by the PSB logical transport parameters. During times when the PSB logical transport is inactive (or absent) then the devices may engage in activity on other physical channels, or enter reduced power mode.

4.2.1.7 Role switch procedure

The role switch procedure is a method for swapping the roles of two devices connected in a piconet. The procedure involves moving from the physical channel that is defined by the original master device to the physical channel that is defined by the new master device. In the process of swapping from one physical channel to the next, the hierarchy of physical links and logical transports over the BR/EDR Controller are removed and rebuilt, with the exception of the ASB and PSB logical transports that are implied by the topology and are not preserved. Note that the role switch procedure does not affect AMP physical channels. After the role switch, the original piconet physical channel may cease to exist or may be continued if the original master had other slaves that are still connected to the channel.

The procedure only copies the default ACL logical links and supporting layers to the new physical channel. Any additional logical transports are not copied by this procedure, and if required this must be carried out by higher layers. The LT_ADDRs of any affected transports may not be preserved as the values may already be in use on the new physical channel.

If there are any QoS commitments or modes such as sniff mode on the original logical transports, then these are not preserved after a role switch. These must be renegotiated after the role switch has completed.

4.2.1.8 Enhanced Data Rate

Enhanced Data Rate is a method of extending the capacity and types of Bluetooth packets for the purposes of increasing the maximum throughput, providing better support for multiple connections, and lowering power consumption, while the remainder of the architecture is unchanged.

Enhanced Data Rate may be selected as a mode that operates independently on each logical transport. Once enabled, the packet type bits in the packet header are interpreted differently from their meaning in Basic Rate mode. This different interpretation is clarified in conjunction with the logical transport address field in the header. The result of this interpretation allows the packet payload header and payload to be received and demodulated according to the packet type. Enhanced Data Rate can be enabled only for ACL-U, eSCO-S logical transports and cannot be enabled for ACL-C, SCO-S, and the broadcast logical transports.

4.2.2 LE Procedures

4.2.2.1 Device Filtering Procedure

The device filtering procedure is a method for controllers to reduce the number of devices requiring communication responses. Since it is not required to respond to requests from every device, it reduces the number of transmissions an LE Controller is required to make which reduces power consumption. It also reduces the communication the controller would be required to make with the host. This results in additional power savings since the Host does not have to be involved.

An advertising or scanning device may employ device filtering to restrict the devices in which it receives advertising packets, scan requests or connection requests. In LE, some advertising packets received by a scanning device require that the scanning device send a request to the advertising device. This advertisement can be ignored if device filtering is used and the advertising device is being filtered. A similar situation occurs with connection requests. Connection requests must be responded to by initiators unless a device filter is used to limit the devices the initiator is required to respond. Advertisers can also use device filters to limit the devices in which it will accept a scan request or connection request.

This device filtering is accomplished through the use of a “White List” located in the LL block of the controller. A white list enumerates the remote devices that are allowed to communicate with the local device. When a white list is in effect, transmissions from devices that are not in the white list will be ignored by the

LL. Since device filtering occurs in the LL it can have a significant impact on power consumption by filtering (or ignoring) advertising packets, scan requests or connection requests from being sent to the higher layers for handling.

The use of device filtering during certain procedures needs to be evaluated carefully to ensure devices are not unintentionally ignored, which may cause interoperability problems when attempting to establish connections or receive advertising broadcasts.

4.2.2.2 Advertising Procedure

An advertiser uses the advertising procedure to perform unidirectional broadcasts to devices in the area. The unidirectional broadcast occurs without a connection between the advertising device and the listening devices. The advertising procedure can be used to establish connections with nearby initiating devices or used to provide periodic broadcast of user data to scanning devices listening on the advertising channel. The advertising procedure uses the advertising physical channel for all advertising broadcasts.

Advertising devices can be connected to other LE devices in an LE piconet. However advertisers are limited to using non-connectable advertising events if they are already connected to another LE device. Time spent advertising while connected needs to be balanced with the connection requirements needed to maintain the already established connection(s) (if the device is a slave in the piconet then it needs to maintain its connection with the master and if the device is the master it needs to maintain its connection(s) with the one or more slaves in the piconet).

Advertising devices may receive scan requests from listening devices in order to get additional user data from the advertising device. Scan responses are sent by the advertising device to the device making the scan request over the same advertising physical channel. Whereas the broadcast user data sent as part of the advertising packets is typically dynamic in nature, scan response data is generally static in nature.

An advertising device may receive connection requests from initiator devices on the advertising broadcast physical channel. If the advertising device was using a connectable advertising event and the initiating device is not being filtered by the device filtering procedure, the advertising device ceases advertising and enters the connected mode. The device can begin advertising again after it is in the connected mode but it can only use non-connected advertising events.

4.2.2.3 Scanning Procedure

A scanning device uses the scanning procedure to listen for unidirectional broadcasts of user data from advertising devices using the advertising physical channel. A scanning device can request additional user data from an advertising device by making a scan request over the advertising physical channel.

The advertising device responds to these requests with additional user data sent to the scanning device over the advertising physical channel.

The scanning procedure can be used while connected to other LE devices in an LE piconet. Time spent scanning while connected needs to be balanced with the connection requirements needed to maintain the already established connection with the other LE devices in the piconet.

If the broadcasts are connectable advertising events and the scanning device is in the initiator mode, it can initiate a connection by sending a connection request on the advertising broadcast physical channel to the advertising device. Once the connection request is sent, the scanning device ceases the initiator mode scanning for additional broadcasts and enters the connected mode. The device can use the scanning procedure after it enters the connected mode. For a master device, using the initiator mode and scanning for connectable advertisements is how additional devices can be added to the master's LE piconet.

4.2.2.4 Discovering Procedure

Bluetooth devices use the advertising procedure and scanning procedure to discover nearby devices, or to be discovered by devices in a given area.

The discovery procedure is asymmetrical. A Bluetooth device that tries to find other nearby devices is known as a discovering device and listens for devices advertising scannable advertising events. Bluetooth devices that are available to be found are known as discoverable devices and actively broadcast scannable advertising events over the advertising broadcast physical channel.

Both discovering and discoverable devices may already be connected to other Bluetooth devices in a piconet. Any time spent inquiring or occupying the advertising broadcast physical channel needs to be balanced with the connection requirements needed to maintain the already established connection with the other LE devices in the piconet.

Using device filtering by the scanning device will prevent the scanning device from discovering all the devices in a given area.

4.2.2.5 Connecting Procedure

The procedure for forming connections is asymmetrical and requires that one Bluetooth device carries out the advertising procedure while the other Bluetooth device carries out the scanning procedure. The advertising procedure can be targeted, so that only one device will respond to the advertising. The scanning device can also target an advertising device by first discovering that the advertising device is present in a connectable manner, and in the given area, and then scanning only connectable advertising events from that device using the device filter. After receiving connectable advertising events from the targeted advertising device, it can initiate a connection by sending the connec-

tion request to the targeted advertising device over the advertising broadcast physical channel.

Time spent scanning while connected needs to be balanced with the connection requirements needed to maintain the already established connection with the other LE devices in the piconet.

4.2.2.6 Connected Mode

After a successful connection procedure, the devices are physically connected to each other within a piconet. This means that there is a piconet physical channel to which they are both connected, there is a physical link between the devices, and there are default LE-C and LE-U logical links. When in the connected mode it is possible to change the properties of the physical and logical links while remaining connected to the piconet physical channel. It is also possible for the device to carry out advertising, scanning or discovery procedures without needing to disconnect from the original piconet physical channel.

Additional logical links are created using the Link Manager that exchanges LL Protocol messages with the remote Bluetooth device to negotiate the creation and settings for these links. One of these links (LE-C) transports the LL control protocol and is invisible to the layers above the Link Manager. The other link (LE-U) transports the L2CAP signaling protocol and any multiplexed L2CAP best-effort channels. It is common to refer to a default LE ACL logical transport, which can be resolved by context, but typically refers to the default LE-U logical link. Also note that these two logical links share a logical transport.

During the time that a slave device is actively connected to a piconet there is always a default LE ACL logical transport between the slave and the master device. The method of deleting the default LE ACL logical transport is to detach the device from the piconet physical channel, at which time the entire hierarchy of L2CAP channels, logical links, and logical transports between the devices is deleted.

4.2.3 AMP Procedures

4.2.3.1 AMP Discovery Procedures

The AMP Manager is responsible for discovering local AMP Controller(s) and maintaining that list over time as AMPs may be added or removed dynamically from the system. The local AMP Manager may request a list of AMPs from the remote AMP Manager. After the list of AMPs has been requested by the remote device, when an AMP is added or removed from the system, the local AMP Manager will notify that remote AMP Manager of the change.

Each AMP Is identified with an ID and a type. Once the list of AMPs has been received, the AMP Manager may request information (AMP_Info) for each AMP.

4.2.3.2 Physical Link Creation Procedure

To set up an L2CAP channel with a remote device over an AMP physical link (AMP channel), the AMP Manager first discovers the remote AMP(s), collects the necessary information about the remote AMP(s) to set up a physical link in an optimal way and then initiates the physical link creation. This is done over a dedicated L2CAP channel.

The AMP Manager provides the remote AMP information that it collected about the remote AMP to the local AMP PAL. The local AMP PAL then uses this information to create the AMP physical link.

4.2.3.3 Logical Link Creation Procedure

Once a physical link exists, L2CAP creates an AMP logical link with the desired QoS. An L2CAP channel is then created over the logical link, at which point the channel is ready for data communication.

5 SECURITY OVERVIEW

5.1 BR/EDR SECURE SIMPLE PAIRING

The primary goal of Secure Simple Pairing is to simplify the pairing procedure for the user. Secondary goals are to maintain or improve the security in Bluetooth wireless technology. Since high levels of security and ease-of-use are often at opposite ends of the spectrum in many technologies and products, much care has been taken to maximize security while minimizing complexity from the end user's point of view.

5.1.1 Security Goals

Secure Simple Pairing has two security goals: protection against passive eavesdropping and protection against man-in-the-middle (MITM) attacks (active eavesdropping). It is a goal of Secure Simple Pairing to exceed the maximum security level provided by the use of a 16 alphanumeric PIN with the pairing algorithm used in Bluetooth Core Specification version 2.0 + EDR and earlier versions. Note that many Bluetooth devices compliant with Bluetooth Core Specification 2.0 + EDR and earlier versions use a 4-digit PIN or a fixed PIN of commonly known values significantly limiting the security on the link.

5.1.2 Passive Eavesdropping Protection

A strong link key coupled with a strong encryption algorithm is necessary to give the user protection against passive eavesdropping. The strength of the link key is based on the amount of entropy (or randomness) in its generation process which would not be known by an attacker. Using legacy pairing, the only source of entropy is the PIN which, in many use cases, is typically four digits either selected by the user or fixed for a given product. Therefore, if the pairing procedure and one authentication exchange is recorded one can run an exhaustive search to find the PIN in a very short amount of time on commonly available computing hardware. With Secure Simple Pairing, the recording attack becomes much harder as the attacker must have solved a hard problem in public key cryptography in order to derive the link key from the recorded information. This protection is independent of the length of the passkey or other numeric values that the user must handle. Secure Simple Pairing gives the same resistance against the recording and passive eavesdropping attacks even when the user is not required to do anything.

Secure Simple Pairing uses Elliptic Curve Diffie Hellman (ECDH) public key cryptography as a means to thwart passive eavesdropping attacks. ECDH provides a very high degree of strength against passive eavesdropping attacks but it may be subject to MITM attacks, which however, are much harder to perform in practice than the passive eavesdropping attack (see Section 5.1.3, “Man-In-The-Middle Protection,” on page 86).

Using the security protocols in the Bluetooth Core Specification version 2.0 + EDR and earlier with a 16 numeric digit PIN achieves about 53 bits of entropy whereas a 16 character alphanumeric, case sensitive PIN yields about 95 bits of entropy when the entire 62 character set is used ([0, … 9, 'A', … 'Z', 'a', … 'z']). Secure Simple Pairing has approximately 95 bits of entropy using the FIPS approved P192 elliptic curve which is at least as good as the entropy in Bluetooth Core Specification 2.0 + EDR and earlier using a 16 character, alphanumeric, case sensitive PIN. Secure Simple Pairing, therefore, exceeds the security requirements of the Bluetooth SIM Access Profile (SAP) which is the profile with the most stringent security requirements. ECDH cryptography was selected over standard Diffie Hellman (often referred to as DH76) since it is computationally less complex and less likely to exceed the low computational capacity in common Bluetooth Controllers.

5.1.3 Man-In-The-Middle Protection

A man-in-the-middle (MITM) attack occurs when a user wants to connect two devices but instead of connecting directly with each other they unknowingly connect to a third (attacking) device that plays the role of the device they are attempting to pair with. The third device then relays information between the two devices giving the illusion that they are directly connected. The attacking device may even eavesdrop on communication between the two devices (known as active eavesdropping) and is able to insert and modify information on the connection. In this type of attack, all of the information exchanged between the two devices are compromised and the attacker may inject commands and information into each of the devices thus potentially damaging the function of the devices. Devices falling victim to the attack are capable of communicating only when the attacker is present. If the attacker is not active or out range, the two victim devices will not be able to communicate directly with each other and the user will notice it.

To prevent MITM attacks, Secure Simple Pairing offers two user assisted numeric methods: numerical comparison or passkey entry. If Secure Simple Pairing would use 16 decimal digit numbers, then the usability would be the same as using legacy pairing with 16 decimal digit PIN. The chance for a MITM to succeed inserting its own link keys in this case is a 1 in $10^{16} = 2^{53}$ pairing instances, which is an unnecessarily low probability.

Secure Simple Pairing protects the user from MITM attacks with a goal of offering a 1 in 1,000,000 chance that a MITM could mount a successful attack. The strength of the MITM protections was selected to minimize the user impact by using a six digit number for numerical comparison and Passkey entry. This level of MITM protection was selected since, in most cases, users can be alerted to the potential presence of a MITM attacker when the connection process fails as a result of a failed MITM attack. While most users feel that provided that they have not compromised their passkey, a 4-digit key is sufficient for authentication (i.e. bank card PIN codes), the use of six digits allows Secure

Simple Pairing to be FIPS compliant and this was deemed to have little perceivable usability impact.

5.1.4 Association Models

Secure Simple Pairing uses four association models referred to as Numeric Comparison, Just Works, Out Of Band, and Passkey Entry. Each of these association models are described in more detail in the following sections.

The association model used is deterministic based on the I/O capabilities of the two devices.

5.1.4.1 Numeric Comparison

The Numeric Comparison association model is designed for scenarios where both devices are capable of displaying a six digit number and both are capable of having the user enter "yes" or "no". A good example of this model is the cell phone / PC scenario.

The user is shown a six digit number (from "000000" to "999999") on both displays and then asked whether the numbers are the same on both devices. If "yes" is entered on both devices, the pairing is successful.

The numeric comparison serves two purposes. First, since many devices do not have unique names, it provides confirmation to the user that the correct devices are connected with each other. Second, the numeric comparison provides protection against MITM attacks (see Section 5.1.3 on page 86).

Note that there is a significant difference from a cryptographic point of view between Numeric Comparison and the PIN entry model used by Bluetooth Core Specification and earlier versions. In the Numeric Comparison association model, the six digit number is an artifact of the security algorithm and not an input to it, as is the case in the Bluetooth security model. Knowing the displayed number is of no benefit in decrypting the encoded data exchanged between the two devices.

5.1.4.2 Just Works

The Just Works association model is primarily designed for scenarios where at least one of the devices does not have a display capable of displaying a six digit number nor does it have a keyboard capable of entering six decimal digits. A good example of this model is the cell phone/mono headset scenario where most headsets do not have a display.

The Just Works association model uses the Numeric Comparison protocol but the user is never shown a number and the application may simply ask the user to accept the connection (exact implementation is up to the end product manufacturer).

The Just Works association model provides the same protection as the Numeric Comparison association model against passive eavesdropping but offers no protection against the MITM attack.

When compared against today's experience of a headset with a fixed PIN, the security level of the Just Works association model is considerably higher since a high degree of protection against passive eavesdropping is realized.

5.1.4.3 Out of Band

The Out of Band (OOB) association model is primarily designed for scenarios where an Out of Band mechanism is used to both discover the devices as well as to exchange or transfer cryptographic numbers used in the pairing process. In order to be effective from a security point of view, the Out of Band channel should provide different properties in terms of security compared to the Bluetooth radio channel. The Out of Band channel should be resistant to MITM attacks. If it is not, security may be compromised during authentication.

The user's experience differs a bit depending on the Out of Band mechanism. As an example, with a Near Field Communication (NFC) solution, the user(s) will initially touch the two devices together, and is given the option to pair the first device with the other device. If "yes" is entered, the pairing is successful. This is a single touch experience where the exchanged information is used in both devices. The information exchanged includes discovery information (such as the Bluetooth Device Address) as well as cryptographic information. One of the devices will use a Bluetooth Device Address to establish a connection with the other device. The rest of the exchanged information is used during authentication.

The OOB mechanism may be implemented as either read only or read/write. If one side is read only, a one-way authentication is performed. If both sides are read/write, a two-way authentication is performed.

The OOB protocol is selected only when the pairing process has been activated by previous OOB exchange of information and one (or both) of the device(s) gives OOB as the IO capabilities. The protocol uses the information which has been exchanged and simply asks the user to confirm connection.

The OOB association model supports any OOB mechanism where cryptographic information and the Bluetooth Device Address can be exchanged. The OOB association model does not support a solution where the user has activated a Bluetooth connection and would like to use OOB for authentication only.

5.1.4.4 Passkey Entry

The Passkey Entry association model is primarily designed for the scenario where one device has input capability but does not have the capability to dis-

play six digits and the other device has output capabilities. A good example of this model is the PC and keyboard scenario.

The user is shown a six digit number (from "000000" to "999999") on the device with a display, and is then asked to enter the number on the other device. If the value entered on the second device is correct, the pairing is successful. Note that there is a significant difference from a cryptographic point of view between Passkey Entry and the PIN entry model used by Bluetooth Core Specification 2.0 + EDR and earlier versions. In the Passkey Entry association model, the six digit number is independent of the security algorithm and not an input to it, as is the case in the 2.0 + EDR security model. Knowing the entered number is of no benefit in decrypting the encoded data exchanged between the two devices.

5.1.4.5 Association Model Overview

The following diagram shows Secure Simple Pairing from the point of view of the technology used for discovery and then the different association possibilities.

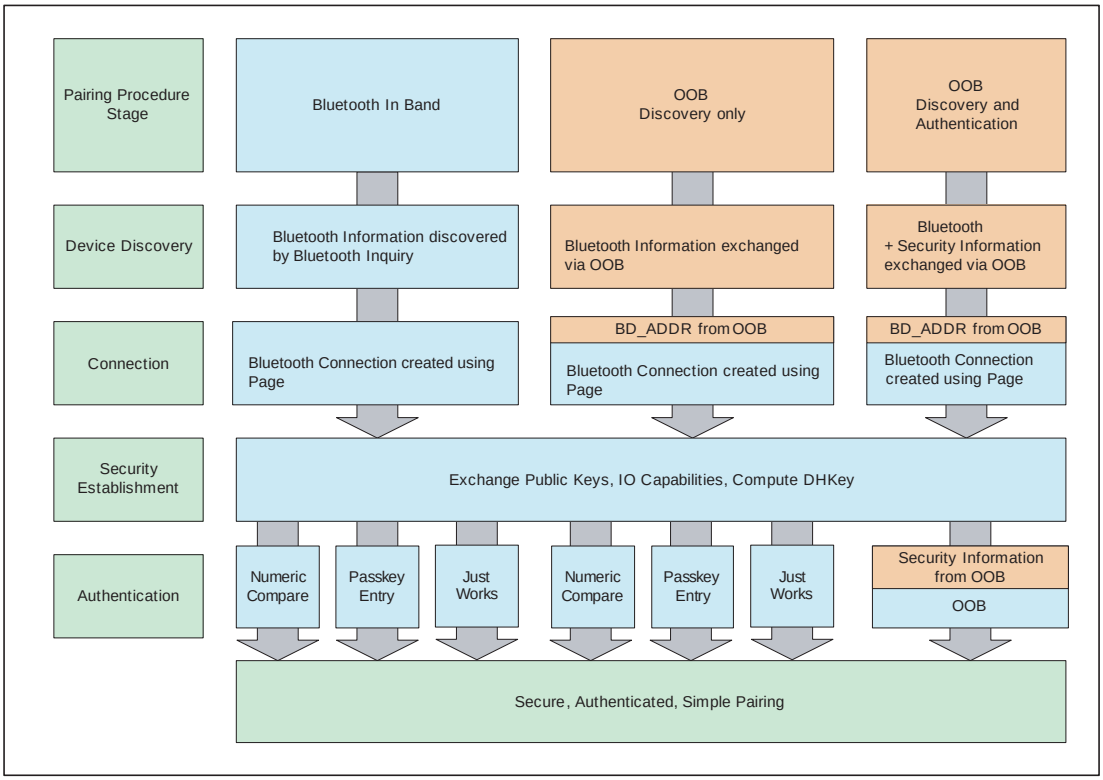


Figure 5.1: Secure Simple Pairing Association Models

5.2 LE SECURITY

Bluetooth LE has some differences in security aspects with respect to BR/EDR security features such as Secure Simple Pairing. The association models are

similar to Secure Simple Pairing from the user perspective and have the same names but have differences in the quality of the protection provided.

The overall goal of keeping the cost of the Controller and the complexity of a slave device to a minimum was used in making compromises on security capabilities in LE.

5.2.1 Association Models

Bluetooth LE uses three association models referred to as Just Works, Out of Band and Passkey Entry. Bluetooth LE does not have an equivalent of Numeric Comparison. Each of these association models is similar to Secure Simple Pairing with the following exceptions.

- Just Works and Passkey Entry do not provide any passive eavesdropping protection. This is because Secure Simple Pairing uses Elliptic Curve Diffie-Hellman and LE does not.

The use of each association model is based on the I/O capabilities of the devices in a similar manner as Secure Simple Pairing.

5.2.2 Key Generation

Key generation in Bluetooth LE is performed by the Host on each LE device independent of any other LE device. Note: Key generation in BR/EDR is performed in the Controller. By performing key generation in the Host, the key generation algorithms can be upgraded without the need to change the Controller.

Bluetooth LE uses multiple keys, each for a specific purpose, as follows:

- Confidentiality of data and device authentication
- Authentication of unencrypted data
- Device Identity

In LE a single link key is generated by combining contributions from each device into a link key used during pairing. In BR/EDR key generation is performed in the Controller.

5.2.3 Encryption

Encryption in Bluetooth LE uses AES-CCM cryptography. Like BR/EDR, in LE encryption is performed in the Controller.

5.2.4 Signed Data

Bluetooth LE supports the ability to send authenticated data over an unencrypted ATT bearer between two devices with a trusted relationship. This is accomplished by signing the data with a Connection Signature Resolving Key

(CSRK). The sending devices places a signature after the Data PDU. The receiving verifies the signature and if the signature is verified the Data PDU is assumed to come from the trusted source. The signature is composed of a Message Authentication Code generated by the signing algorithm and a counter. The counter is used to protect against a replay attack and is incremented on each signed Data PDU sent.

5.2.5 Privacy Feature

Bluetooth LE supports a feature that reduces the ability to track a LE device over a period of time by changing the Bluetooth device address on a frequent basis. The privacy feature is not used in the GAP discovery mode and procedures but it is used when supported during connection mode and connection procedures.

In order for devices using the privacy feature to reconnect to known devices, the device addresses used when the privacy feature is enabled, private address, must be resolvable to the other devices' identity. The private address is generated using the device's identity key exchanged during the bonding procedure. Use of the private address for reconnection is limited to use cases where filtering of devices disabled. Disabling device filtering may increase the power consumption of the device since the host will need to process all device requests. Situations where the device filtering is disabled should only be used for limited periods of time to minimize the power consumption in this mode.

The privacy feature also defines a reconnection address which allows for bonded device to reconnect while also filtering devices to known devices. The reconnect address is exchanged between the two devices at each connection. Since reconnect addresses only change between connections, device filtering can be used to minimize processing of excessive requests.

5.3 AMP SECURITY

AMP security does not change the user experience because it utilizes the same Secure Simple Pairing association models that were introduced in the Bluetooth 2.1 + EDR Core Specification. From the user's point of view, all radios are "paired" in one process.

AMP security starts during the Secure Simple Pairing process. The link key for BR/EDR is generated during Phase 4 of Secure Simple Pairing. A 256-bit Generic AMP Link Key (GAMP_LK) is generated from the BR/EDR Link Key. The Generic AMP Link Key is stored in the security database alongside the BR/EDR Link Key once pairing has completed.

AMP security does not affect the BR/EDR Link Key so backwards compatibility is maintained for devices that support the Generic AMP feature with devices that do not.

When an AMP is used for the first time, a dedicated AMP key is created by the AMP Manager for that AMP type using the new Secure Simple Pairing function h2 and a KeyID for the AMP type. The length of the Dedicated AMP Link Key depends on the AMP Type. If the Pair-wise Master key already exists due to a previous connection the AMP link key is not created and the stored key is reused.

The Dedicated AMP Link Key is sent to the PAL during the Physical Link creation process. Each PAL is responsible for using the Dedicated AMP Link Key during the security phase of the physical link establishment process. Note that a Dedicated AMP link key is used for multiple sessions over the same AMP.

Each time a dedicated AMP key is successfully created, the Generic AMP Link Key is updated. This is performed using the h2 function with KeyID "gamp."

6 BLUETOOTH APPLICATION ARCHITECTURE

6.1 BLUETOOTH PROFILES

Application interoperability in the Bluetooth system is accomplished by Bluetooth profiles. Bluetooth profiles define the required functions and features of each layer in the Bluetooth system from the PHY to L2CAP and any other protocols outside of the Core specification. The profile defines the vertical interactions between the layers as well as the peer-to-peer interactions of specific layers between devices.

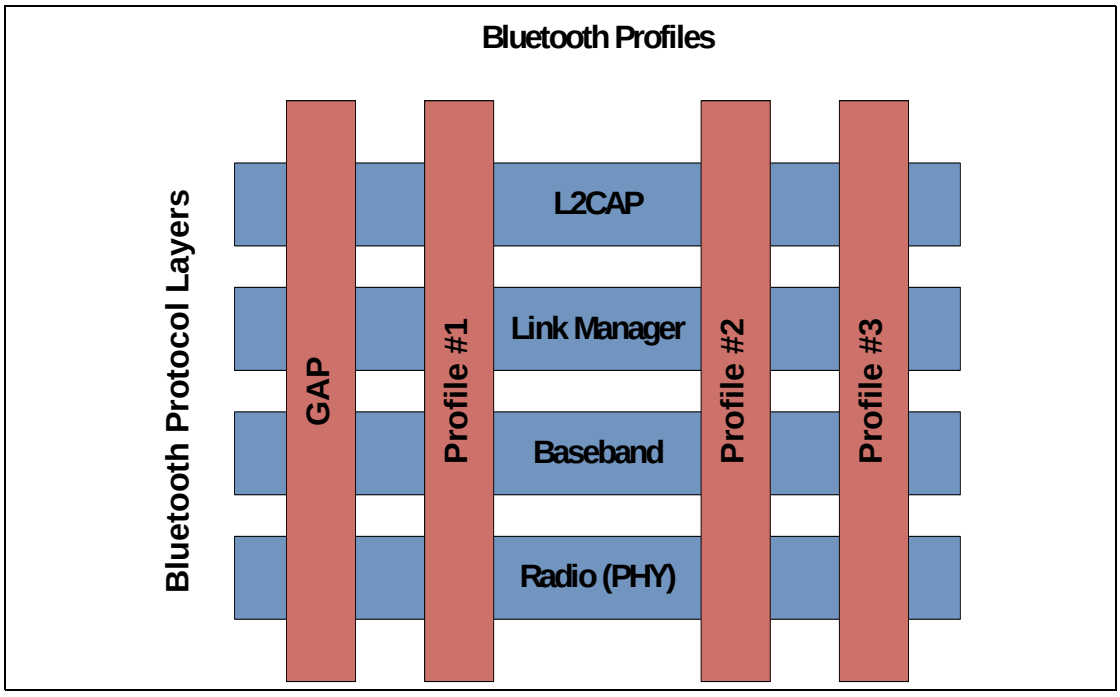


Figure 6.1: Bluetooth Profiles

In addition, application behaviors and data formats are also defined by the profile. When two devices comply with all the requirements of a Bluetooth profile, interpretability of an application is enabled.

All profiles describe service discovery requirements necessary for devices to connect, find available application services and connection information necessary for making application level connections.

6.2 GENERIC ACCESS PROFILE

The Bluetooth system defines a base profile which all Bluetooth devices implement. This profile is the Generic Access Profile (GAP), which defines the basic requirements of a Bluetooth device. For instance, for BR/EDR, it defines a Bluetooth device to include the Radio, Baseband, Link Manager, L2CAP, and the service discovery protocol functionality; for LE, it defines the Physical

Layer, Link Layer, L2CAP, Security Manager, Attribute Protocol and Generic Attribute Profile. This ties all the various layers together to form the basic requirements for a Bluetooth device. It also describes the behaviors and methods for device discovery, connection establishment, security, authentication, association models and service discovery.

In BR/EDR, GAP defines a single role with functionality that may be present in each device. This functionality includes how devices discovery each other, establish connections and describes security association models used for authentication. In BR/EDR this functionality may be present in both devices. It may be necessary for a device to implement both the initiating and accepting functionality if the device wants to discover or establish connections with all devices. A device may only include either the initiating or the accepting functionality but it requires the remote device to support the complimentary functionality to discovery or establish connections with the device. For BR/EDR, the Controller is required to support all the functionality, however the Host may limit this functionality based on the other profiles or use cases supported by the device.

In LE, GAP defines four specific roles: Broadcaster, Observer, Peripheral, and Central. A device may support multiple LE GAP roles provided that the underlying Controller supports those roles or role combinations. However, only one LE GAP role may be supported at a given time. Each role specifies the requirements for the underlying Controller. This allows for Controllers to be optimized for specific use cases.

The **Broadcaster** role is optimized for transmitter only applications. Devices supporting the broadcaster role use advertising to broadcast data. The broadcaster role does not support connections. The **Observer** role is optimized for receiver only applications. Devices supporting the observer role are the complementary device for a broadcaster and receives broadcast data contained in advertisements. The observer role does not support connections. The **Peripheral** role is optimized for devices that support a single connection and are less complex than central devices. Devices supporting the peripheral role only require Controllers that support the Controller's slave role. The **Central** role supports multiple connections and is the initiator for all connections with devices in the peripheral role. Devices supporting the central role require a Controller that support the Controller's master role and generally supports more complex functions compared to the other LE GAP roles.

6.3 PROFILE HIERARCHY

Since all Bluetooth devices are required to implement GAP, any additional profiles implemented by a Bluetooth device become supersets of GAP. Depending on the complexity of an application or the ability to reuse common requirements of functionality of the Bluetooth system between many applications, additional generic profiles can be created that are both a superset of GAP as well as being a superset of another profile. A top level profile that describes application interoperability is called an Application Profile.

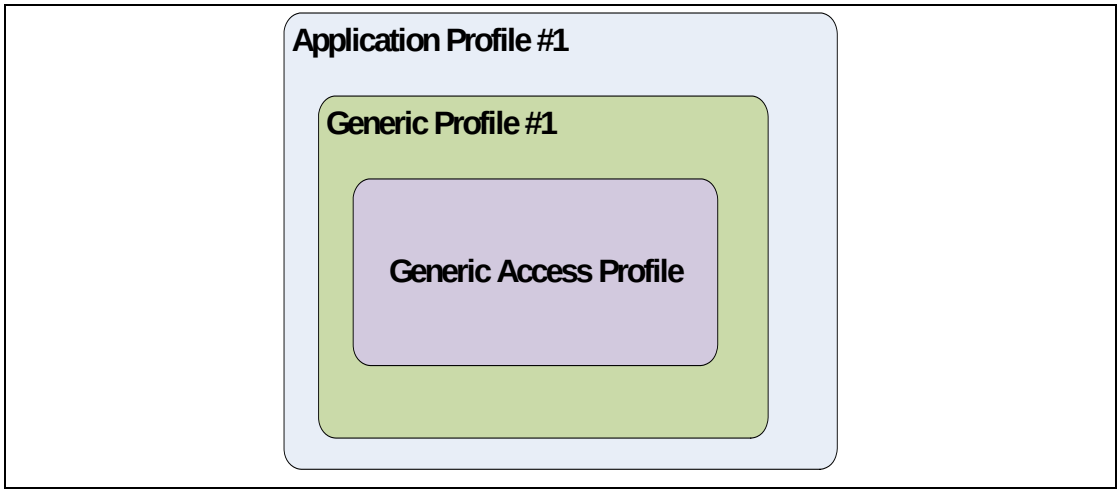


Figure 6.2: Profile Hierarchy

Application profiles contain by reference GAP and any other generic profile that describes a set of common requirements of the Bluetooth system.

6.4 GENERIC ATTRIBUTE PROFILE

Generic Attribute Profile (GATT) is built on top of the Attribute Protocol (ATT) and establishes common operations and a framework for the data transported and stored by the Attribute Protocol. GATT defines two roles: Server and Client. The GATT roles are not necessarily tied to specific GAP roles and but may be specified by higher layer profiles. GATT and ATT are not transport specific and can be used in both BR/EDR and LE. However, GATT and ATT are mandatory to implement in LE since it is used for discovering services

The GATT server stores the data transported over the Attribute Protocol and accepts Attribute Protocol requests, commands and confirmations from the GATT client. The GATT server sends responses to requests and when configured, sends indication and notifications asynchronously to the GATT client when specified events occur on the GATT server.

GATT also specifies the format of data contained on the GATT server. Attributes, as transported by the Attribute Protocol, are formatted as **Services** and **Characteristics**. Services may contain a collection of characteristics. Characteristics contain a single value and any number of descriptors describing the characteristic value.

With the defined structure of services, characteristics and characteristic descriptors a GATT client that is not specific to a profile can still traverse the GATT server and display characteristic values to the user. The characteristic descriptors can be used to display descriptions of the characteristic values that may make the value understandable by the user.

6.5 GATT-BASED PROFILE HIERARCHY

The GATT Profile specifies the structure in which profile data is exchanged. This structure defines basic elements such as services and characteristics, used in a profile.

The top level of the hierarchy is a profile. A profile is composed of one or more services necessary to fulfill a use case. A service is composed of characteristics or references to other services. Each characteristic contains a value and may contain optional information about the value. The service and characteristic and the components of the characteristic (i.e., value and descriptors) contain the profile data and are all stored in Attributes on the server.

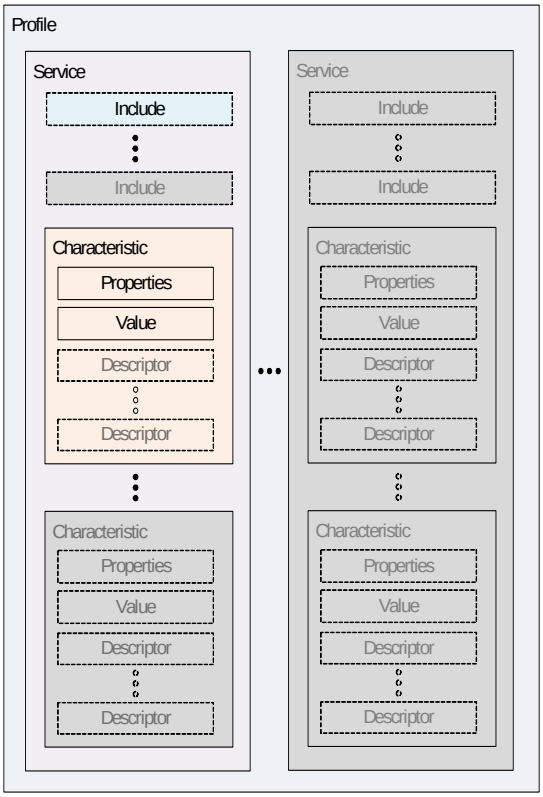


Figure 6.3: GATT-Based Profile Hierarchy

6.5.1 Service

A service is a collection of data and associated behaviors to accomplish a particular function or feature of a device or portions of a device. A service may reference other primary or secondary services and/or a set of characteristics that make up the service.

There are two types of services: primary and secondary. A primary service is a service that provides the primary functionality of a device. A secondary service is a service that provides auxiliary functionality of a device and is referenced from at least one primary service on the device.

To maintain backward compatibility with earlier clients, later revisions of a service definition can only add new referenced services or optional characteristics. Later revisions of a service definition are also forbidden from changing behaviors from previous revision of the service definition.

Services may be used in one or more profiles to fulfill a particular use case.

6.5.2 Referenced Services

A referenced service is a method to incorporate another service definition on the server as part of the service referencing it. When a service references another service, the entire referenced service becomes part of the new service including any nested referenced services and characteristics. The referenced service still exists as an independent service. There are no limits to the depth of nested references.

6.5.3 Characteristic

A characteristic is a value used in a service along with properties and configuration information about how the value is accessed and information about how the value is displayed or represented. A characteristic definition contains a characteristic declaration, characteristic properties, and a value. It may also contain descriptors that describe the value or permit configuration of the server with respect to the characteristic value.

ACRONYMS & ABBREVIATIONS

CONTENTS

1 List of Acronyms and Abbreviations 101

1 LIST OF ACRONYMS AND ABBREVIATIONS

Acronym or abbreviation	Writing out in full	Which means
8DPSK	8 phase Differential Phase Shift Keying	3 Mbps modulation type used by Enhanced Data rate
π/4 DQPSK	pi/4 Rotated Differential Quaternary Phase Shift Keying	2 Mbps modulation type used by Enhanced Data Rate
A		
A2MP	AMP Manager Protocol	
AAD	Additional Authentical Data	
AC	Access Category	
ACI	Access Category Index	
ACK	Acknowledge/ACKnowledgement	
ACL	Asynchronous Connection-oriented [logical transport]	Reliable or time-bounded, bi-directional, point-to-point.
ACL-C	ACL Control [logical link] (LMP)	
ACL-U	ACL User [logical link] (L2CAP)	
ACO	Authenticated Ciphering Offset	
AD	Advertising Data	
ADVB	LE Advertising Broadcast	
ADVB-C	LE Advertising Broadcast Control (Logical Link)	
ADVB-U	LE Advertising Broadcast User Data (Logical Link)	
Adv_idx	Advertising channel index	
AES	Advanced Encryption Standard	
AGC	Automatic Gain Control	
AFH	Adaptive Frequency Hopping	
AGC	Automatic Gain Control	
AHS	Adapted Hop Sequence	
AIFSN	Arbitration Interframe Space Number	
AMP	Alternate MAC PHY	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
AMP_ASSOC	Alternate MAC PHY association information	
AMP-U	AMP User Asynchronous/Isochronous Logical Link	
AR_ADDR	Access Request Address	
ARQ	Automatic Repeat Request	
ASB	Active Slave Broadcast [logical transport]	Unreliable, uni-directional broadcast to any devices synchronized with the physical channel.
ASB-U	ASB User [logical link] (L2CAP)	
ATT	Attribute Protocol	
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	
C		
CAC	Channel Access Code	
CL	Connectionless	
CCM	Counter with Cipher Block Chaining-Message Authentication Code	
CCMP	CTR with CBC-MAC Protocol	
CKLN	Native Clock	
CLK	Master Clock	
CLKE	Estimated Clock	
CODEC	COder DECoder	
COF	Ciphering Offset	
CRC	Cyclic Redundancy Check	
CTS	Clear to send	
CVSD	Continuous Variable Slope Delta Modulation	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
D		
DA	Destination address	
DAC	Device Access Code	
DCE	Data Communication Equipment	
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	
DEVM	Differential Error Vector Magnitude	Measure of modulation error used for Enhanced Data Rate transmitter testing
DFS	Dynamic Frequency Selection	
DH	Data-High Rate	Data packet type for high rate data
DHK	Diversifier Hiding Key	
DIAC	Dedicated Inquiry Access Code	
DIV	Diversifier	
DM	Data - Medium Rate	Data packet type for medium rate data
DPSK	Differential Phase Shift Keying	Generic description of Enhanced Data Rate modulation
DQPSK	Differential Quaternary Phase Shift Keying	Modulation type used by Enhanced Data Rate
DSAP	Destination Service Access Point	
DTE	Data Terminal Equipment	In serial communications, DTE refers to a device at the endpoint of the communications path; typically a computer or terminal.
DTM	Direct Test Mode	
DUT	Device Under Test	
DV	Data Voice	Data packet type for data and voice
E		
ECWmax	Enhanced Contention Window maximum	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
ECWmim	Enhanced Contention Window minimum	
ED	Erroneous Data	
EDCA	Enhanced Distributed Channel Access	
EDIV	Encrypted Diversifier	
EDR	Enhanced Data Rate	
EFS	Extended Flow Specification	
EIR	Extended Inquiry Response	Host supplied information transmitted in the inquiry responses substate.
EIRP	Effective Isotropic Radiated Power	Equivalent power that an isotropic antenna must transmit to provide the same field power density
EOC	Extreme Operating Conditions	
ER	Encryption Root	
EPR	Encryption Pause Resume	Mode allowing the device to initiate the pause and resume encryption sequence
ERP	Extended Rate PHY conforming to Clause 19	
eSCO	Extended Synchronous Connection Oriented [logical transport]	Bi-directional, symmetric or asymmetric, point-to-point, general regular data, limited retransmission.
eSCO-S	Stream eSCO (unframed)	used to support isochronous data delivered in a stream without framing
ESS	Extended Service Set	
ETSI	European Telecommunications Standards Institute	
F		
FCC	Federal Communications Commission	
FCS	Frame Check Sequence	
FDMA	Frequency Division Multiple Access	
FEC	Forward Error Correction code	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
FH	Frequency Hopping	
FHS	Frequency Hop Synchronization	
FHSS	Frequency Hopping Spread Spectrum	
FIFO	First In First Out	
FIPS	Federal Information Processing Standards	
FM	Frequency Modulation	Modulation Type
G		
GAP	Generic Access Profile	
GATT	Generic Attribute Protocol	
GFSK	Gaussian Frequency Shift Keying	
GIAC	General Inquiry Access Code	
GTM	Generic Test Methodology	
H		
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	
IC	Industry Canada	
IEC	International Electrotechnical Commission	
IEEE	Institute of Electronic and Electrical Engineers	
IETF	Internet Engineering Task Force	
IFS	Inter Frame Space	
IP	Internet Protocol	
IPv4	Internet Protocol version 4	
IPv6	Internet Protocol version 6	
IR	Identity Root	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
IrDA	Infra-red Data Association	
IRK	Identity Root Key	
ISM	Industrial, Scientific, Medical	
ISO	International Organization for Standardization	
IUT	Implementation Under Test	
ITU	International Telecommunication Union	
IV	Initialization Vector	
IVm	Initialization Vector (master)	
IVs	Initialization Vector (slave)	
J		
JRL	Japanese Radio Law	
K		
KCC	Korea Communications Commission	
L		
L2CAP	Logical Link Control and Adaptation Protocol	
LAP	Lower Address Part	
LC	Link Controller	Link Controller (or baseband) part of the Bluetooth protocol stack. Low level Baseband protocol handler
LC	Link Control [logical link]	The control logical links LC and ACL-C are used at the link control level and link manager level, respectively.
LCP	Link Control Protocol	
LE	Low Energy	
LE-C	Low Energy Control (link)	
LE-U	LE User [logical link]	
LFSR	Linear Feedback Shift Register	
LL	Link Layer	
LLCP	Link Layer Control Protocol	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
LLID	Logical Link Identifier	
LM	Link Manager	
LMP	Link Manager Protocol	For LM peer to peer communication
LR	Loudness Rating	
LSB	Least Significant Bit	
LSO	Least Significant Octet	
LSTO	Link Supervision Timeout Event	Controller can send LSTO event to Host
LT_ADDR	Logical Transport ADDRess	
LTK	Long-Term Key	
M		
M	Master or Mandatory	
MAC	Medium Access Control	
MAC	Message Authentication Code	Usually meaning the Medium Access Control code
MD	More Data	
Mbps	Million (Mega) bits per second	
MIC	Message Integrity Check	
MIIT	Ministry of Industry and Information Technology	
MITM	Man-in-the-middle	
MMI	Man Machine Interface	
MS	Mobile Station	
MSC	Message Sequence Chart	
MSB	Most Significant Bit	
MSC	Message Sequence Chart	
MSO	Most Significant Octet	
MTU	Maximum Transmission Unit	
N		
NAK	Negative Acknowledge	
NAP	Non-significant Address Part	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
NCC	National Communications Commission	
NESN	Next Expected Sequence Number	
NIST	National Institute of Standards and Technology	
NOC	Normal Operating Conditions	
O		
O	Optional	
OBEX	OBject EXchange protocol	
OCF	OpCode Command Field	
OGF	OpCode Group Field	
OOB	Out of Band	
P		
PBF	Packet Boundary Flag	The device supports the capability to correctly handle HCI ACL Data Packets
PCM	Pulse Coded Modulation	
PDU	Protocol Data Unit	a message
PHY	Physical Layer	
PIN	Personal Identification Number	
PM_ADDR	Parked Member Address	
PN	Pseudo-random Noise	
PPM	Part Per Million	
PPP	Point-to-Point Protocol	
PRBS	Pseudo Random Bit Sequence	
PRNG	Pseudo Random Noise Generation	
PSB	Parked Slave Broadcast [logical transport]	Unreliable, uni-directional broadcast to all piconet devices.
PSB-C	PSB Control [logical link] (LMP)	
PSB-U	PSB User [logical link] (L2CAP)	
PSK	Phase Shift Keying	Class of modulation types
PSTN	Public Switched Telephone Network	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
ptt	Packet Type Table	The ptt parameter is used to select the logical transport types via LMP.
Q		
QoS	Quality of Service	
R		
RAND	Random number	
RF	Radio Frequency	
RFC	Request For Comments	
RFCMode	Retransmission and Flow Control Mode	
RFCOMM		Serial cable emulation protocol based on ETSI TS 07.10
RFU	Reserved for Future Use	
RMS	Root Mean Square	
RSSI	Received Signal Strength Indication	
RX	Receive	
S		
S	Slave	
SAP	Service Access Points	
SAR	Segmentation and Reassembly	
SCA	Sleep Clock Accuracy	
SCO	Synchronous Connection-Oriented [logical transport]	Bi-directional, symmetric, point-to-point, AV channels.
SCO-S	Stream SCO (unframed)	
SCO-S	Synchronous logical link	used to support isochronous data delivered in a stream without framing
SD	Service Discovery	
SDP	Service Discovery Protocol	
SDU	Service Data Unit	
SEQN	Sequential Numbering scheme	
SK	Session Key	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
SKDm	Session Key Diversifier (master)	Master portion of the Session Key Diversifier
SKDs	Session Key Diversifier (slave)	Slave portion of the Session Key Diversifier
SLR	Send Loudness Ring	
SM	Security Manager	
SMP	Security Manager Protocol	
SN	Sequence Number	
SRES	Signed Response	
SRK	Signature Resolving Key	
SS	Supplementary Services	
SSI	Signal Strength Indication	
SSP	Secure Simple Pairing	Simplifies the pairing process and improves Bluetooth security.
SSR	Sniff Subrating	A mode that defines the anchor points at which the master transmits to the slave.
STK	Short Term Key	
SW	Software	
T		
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	
TDMA	Time Division Multiple Access	
T_IFS	Time Inter Frame Space	Time interval between consecutive packets on same channel index
TK	Temporary Key	
TX	Transmit	
U		
UAP	Upper Address Part	

Table 1.1: Acronyms and Abbreviations

Acronym or abbreviation	Writing out in full	Which means
UART	Universal Asynchronous receiver Transmitter	
UI	User Interface	
UI	Unnumbered Information	
ULAP	Upper and Lower Address Parts	
USB	Universal Serial Bus	
UTF-8	8-bit UCS/Unicode Transformation Format	
UUID	Universal Unique Identifier	
W		
WAP	Wireless Application Protocol	

Table 1.1: Acronyms and Abbreviations

CORE SPECIFICATION CHANGE HISTORY

CONTENTS

11 Deprecated Features 116

2 Changes from V1.1 to V1.2 117

2.1 New Features 117
2.2 Structure Changes 117
2.3 Deprecated Features list 117
2.4 Changes in Wording 118
2.5 Nomenclature Changes 118

3 Changes from V1.2 to V2.0 + EDR 119

3.1 New Features 119
3.2 Deprecated Features 119

4 Changes from V2.0 + EDR to V2.1 + EDR 120

4.1 New features 120
4.2 Deprecated Features 120

5 Changes From V2.1 + EDR To V3.0 + HS 121

5.1 New Features 121
5.2 Deprecated Features 121

6 Changes From V3.0 + HS To v4.0 122

6.1 New Features 122
6.2 Deprecated Features 122

1 DEPRECATED FEATURES

Some features have been deemed no longer useful and have been deprecated. The term *deprecation* does not mean that they are no longer allowed, but that they are no longer recommended as the best way of performing a given function.

Features that have been deprecated are not included in this specification or corresponding test documentation. Deprecated features may be implemented using the latest qualifiable version of the Bluetooth core specification that specifies the deprecated features.

2 CHANGES FROM V1.1 TO V1.2

2.1 NEW FEATURES

Several new features were introduced in the Bluetooth Core Specification 1.2. The major areas of improvement are:

- Architectural overview
- Faster connection
- Adaptive frequency hopping
- Extended SCO links
- Enhanced error detection and flow control
- Enhanced synchronization capability
- Enhanced flow specification

The feature descriptions are incorporated into the existing text in different core parts, described in Volumes 2 and 3.

2.2 STRUCTURE CHANGES

The Bluetooth Core Specification 1.2 was significantly restructured for better consistency and readability. The most important structure changes have been performed in Baseband, LMP, HCI and L2CAP. The text in these sections has been rearranged to provide:

- Presentation of the information in a more logical progression
- Removal of redundant text and requirements
- Consolidation of baseband related requirements (for example, the *Baseband Timers* and *Bluetooth Audio* sections into the Baseband Specification)
- Alignment of the specification with the new architecture and terminology presented in the Architecture Overview (see “IEEE Language” on page 131 [vol. 1]).

In addition, new text and requirements were added for the new features as well as many changes throughout the specification to update the text to use IEEE language (see “IEEE Language” on page 131 [vol. 1]).

2.3 DEPRECATED FEATURES LIST

Features deprecated in version 1.2 are:

- The use of Unit Keys for security
- Optional Paging schemes
- 23 channel hopping sequence

- Page scan period mode and associated commands

2.4 CHANGES IN WORDING

Two general classes of changes to the wording of the Bluetooth Specification have been done for version 1.2. They are a formalization of the language by using conventions established by the Institute of Electrical and Electronic Engineers (IEEE), and a regularization of Bluetooth wireless technology-specific terms. Many portions of the version 1.1 specification use imprecise or inaccurate terms to describe attributes of the protocol. A more accurate terminology described in Part E was introduced into the version 1.2 specification and have been applied in future versions.

2.5 NOMENCLATURE CHANGES

The nomenclature in Bluetooth 1.2 was also updated due to new concepts that are introduced together with the new features and the new architecture. See “Architecture” on page 11.

3 CHANGES FROM V1.2 TO V2.0 + EDR

3.1 NEW FEATURES

The Bluetooth Core Specification version 2.0 + EDR introduces Enhanced Data Rate (EDR). EDR provides a set of additional packet types that use the new 2 Mbps and 3 Mbps modes.

In addition to EDR a set of errata provided in ESR02 has been incorporated into this version and revised to include changes caused by the addition of EDR.

These additions are incorporated into the existing text in different core parts described in Volumes 2 and 3.

3.2 DEPRECATED FEATURES

The only feature deprecated in version 2.0 + EDR is the Page Scan Period Mode and associated commands (based on Erratum 694 which is also included in ESR02).

4 CHANGES FROM V2.0 + EDR TO V2.1 + EDR

4.1 NEW FEATURES

Several new features are introduced in Bluetooth Core Specification 2.1 + EDR. The major areas of improvement are:

- Erroneous Data Reporting
- Encryption Pause and Resume
- Extended Inquiry Response
- Link Supervision Timeout Changed Event
- Non-Automatically-Flushable Packet Boundary Flag
- Secure Simple Pairing
- Sniff Subrating
- Security Mode 4

4.2 DEPRECATED FEATURES

No features were deprecated in v2.1 + EDR.

5 CHANGES FROM V2.1 + EDR TO V3.0 + HS

5.1 NEW FEATURES

Several new features are introduced in Bluetooth Core Specification v3.0 + HS. The major areas of improvement are:

- AMP Manager Protocol (A2MP)
- Enhancements to L2CAP including
  - •Enhanced Retransmission Mode and Streaming Mode
  - •Improvements to the L2CAP state machine for AMP channels
  - •Fixed channel support
- Enhancements to HCI for AMP
- Enhancements to Security for AMP
- 802.11 Protocol Adaptation Layer
- Enhanced Power Control
- Unicast Connectionless Data
- HCI Read Encryption Key Size command
- Generic Test Methodology for AMP
- Enhanced USB and SDIO HCI Transports

5.2 DEPRECATED FEATURES

No features were deprecated in v3.0 + HS.

6 CHANGES FROM V3.0 + HS TO V4.0

6.1 NEW FEATURES

Several new features are introduced in Bluetooth Core Specification v4.0 Release. The major areas of improvement are:

- Bluetooth Low Energy including
  - Low Energy Physical Layer
  - Low Energy Link Layer
  - Enhancements to HCI for Low Energy
  - Low Energy Direct Test Mode
  - AES Encryption
  - Enhancements to L2CAP for Low Energy
  - Enhancements to GAP for Low Energy
  - Attribute Protocol (ATT)
  - Generic Attribute Profile (GATT)
  - Security Manager (SM)

6.2 DEPRECATED FEATURES

No features were deprecated in v4.0.

Architecture & Terminology Overview
Part D

MIXING OF SPECIFICATION VERSIONS

CONTENTS

1 Mixing of Specification Versions 126

1.1 Features and their Types 127

1.2 Core Specification Addendums 128

1 MIXING OF SPECIFICATION VERSIONS

This part describes how volumes, and parts within volumes, of different versions of the Core Specification may be mixed in Bluetooth implementations. The Core System consists of a BR/EDR Controller Package (see Volume 2), a Low Energy Controller Package (see Volume 6), a Host Package (see Volume 3) and AMP Protocol Adaptation Layers (see Volume 5).

A Core Specification Addendum contains one or more parts of a single volume or one or more parts in multiple volumes. Addendums may be used to supersede a part in a volume or may be used to add a part to a volume according to the rules in Section 1.2.

- All parts within a Primary Controller implementation shall be the same version of Volume 2 and Volume 6, and shall support at least one new Type 1, -2, or -3 feature from Table 1.2.

  All parts within a Host implementation of Volume 3 shall be the same version. An AMP Controller implementation shall contain parts of Volume 2 and Volume 5 from the same version and shall support at least one new Type 3 or Type 4 feature from Table 1.2.
- The Primary Controller, AMP Controller, and Host may be different versions within a single implementation.

In order to describe how these volumes and parts within volumes can be mixed, one needs to distinguish between four categories of features specified in the different specification versions. The four categories are:

Category	Description
Type 1	Feature that exists below HCI and cannot be configured/enabled via HCI
Type 2	Feature that exists below HCI and can be configured/enabled via HCI
Type 3	Feature that exists below and above HCI and requires HCI command/events to function
Type 4	Feature that exists only above HCI

Table 1.1: Feature type definitions

The outcome of mixing different core system packages are derived from the feature definitions in the table above:

- If an implementation contains features of type 1 or type 4, these features can function with any combination of Host Package and Controller Package or AMP Protocol Adaptation Layer (PAL) versions.
- If an implementation contains features of type 2, these features can only be used under a default condition if a Host Package of an older version is mixed with a Controller Package or AMP PAL of this version.

- In order to fully use the feature under all conditions, the Host Package, Controller Package, and AMP PAL must be of the same or later version.
- If an implementation contains features of type 3, these features can only function if the Host Package supports this version or a later version and if the Controller Package supports this version or a later version.

See the Bluetooth Brand Book for specification naming requirements.

1.1 FEATURES AND THEIR TYPES

The following table lists the features and their types.

Feature	Version	Type
Basic AFH operation	1.2	1
Enhanced inquiry	1.2	1
Configuration of AFH (setting channels and enabling/disabling channel assessment)	1.2	2
Enhanced synchronization capability	1.2	2
Interlaced inquiry scan	1.2	2
Interlaced page scan	1.2	2
Broadcast encryption	1.2	2
Enhanced flow specification and flush time-out	1.2	3
Extended SCO links	1.2	3
Inquiry Result with RSSI	1.2	3
L2CAP flow and error control	1.2	4
2 Mbps EDR	2.0 + EDR	2
3 Mbps EDR	2.0 + EDR	2
3 slot packets in EDR	2.0 + EDR	2
5 slot packets in EDR	2.0 + EDR	2
2 Mbps eSCO	2.0 + EDR	2[1]
3 Mbps eSCO	2.0 + EDR	2*
3 slot packets for EDR eSCO	2.0 + EDR	2*
Erroneous Data Reporting	2.1 + EDR	3
Extended Inquiry Response	2.1 + EDR	3
Encryption Pause and Resume	2.1 + EDR	1
Link Supervision Timeout Changed Event	2.1 + EDR	3

Table 1.2: Features and their types

Feature	Version	Type
Non-Flushable Packet Boundary Flag	2.1 + EDR	3
Sniff subrating	2.1+ EDR	3
Secure Simple Pairing	2.1.+ EDR	3
L2CAP Enhanced Retransmission Mode	Addendum 1/ 3.0 + HS	4
L2CAP Streaming Mode	Addendum 1/ 3.0 + HS	4
Enhanced Power Control	3.0 + HS	2
AMP Manager Protocol (A2MP)	3.0 + HS	4
L2CAP Enhancements for AMP	3.0 + HS	4
802.11 PAL	3.0 + HS	3
Generic Test Methodology	3.0 + HS	3
Unicast Connectionless Data	3.0 + HS	4
Low Energy (up through L2CAP)	4.0	3
Attribute Protocol	4.0	4
Generic Attribute Profile	4.0	4
Security Manager	4.0	3

Table 1.2: Features and their types

1. The EDR eSCO options are marked as 2* because eSCO requires profile support, but if a product includes the eSCO option from V1.2, then EDR eSCO will be supported without any new support above HCI.

1.2 CORE SPECIFICATION ADDENDUMS

The following table contains a list of Core Specification Addendums and Core Specification versions they are allowed to be used with.

Each part within an addendum is identified by a type indicating whether it may replace a part in an allowed core specification version or whether it may add to a package within a core specification version.

Type	Description
Replacement	The part in the addendum is used instead of the equivalent part in the allowed specification versions.
Addition	The Part in the Addendum is used in addition to the existing parts in the allowed specification versions.

The following table lists adopted addendums.

Addendum	Volume and Part	Type	Allowed Versions
1	Volume 3, Part A	Replacement	1.2, 2.0 + EDR, 2.1 + EDR

Architecture & Terminology Overview
Part E

IEEE LANGUAGE

CONTENTS

1 Use of IEEE Language 135

1.1 Shall 135

1.2 Must 136

1.3 Will 136

1.4 Should 136

1.5 May 136

1.6 Can 137

1 USE OF IEEE LANGUAGE

One of the purposes of this terminology is to make it easy for the reader to identify text that describes requirements as opposed to background information. The general term for text that describes attributes that are required for proper implementation of Bluetooth wireless technology is normative. The general term for language that provides background and context for normative text is informative. These terms are used in various sections to clarify implementation requirements.

Many portions of the Bluetooth Specification use imprecise or inaccurate terms to describe attributes of the protocol. This subsection describes the correct usage of key terms that indicate degree of requirements for processes and data structures. The information here was derived from the Institute of Electrical and Electronics Engineers (IEEE) Style Guide, see http://standards.ieee.org/guides/style/.

The following list is a summary of the terms to be discussed in more detail below:

shall	is required to – used to define requirements
must	is a natural consequence of -- used only to describe unavoidable situations
will	it is true that -- only used in statements of fact
should	is recommended that – used to indicate that among several possibilities one is recommended as particularly suitable, but not required
may	is permitted to – used to allow options
can	is able to – used to relate statements in a causal fashion
is	is defined as – used to further explain elements that are previously required or allowed
note	<informational text ONLY>

For clarity of the definition of those terms, the following sections document why and how they are used. For these sections only, the IEEE terms are italicized to indicate their use as a noun. Uses and examples of the use of the terms in this section are underlined.

1.1 SHALL

The word *shall* is used to indicate mandatory requirements that shall be followed in order to conform to the specification and from which no deviation is permitted.

There is a strong implication that the presence of the word *shall* indicates a testable requirement. All testable requirements shall be reflected in the Protocol Implementation Conformance Statement (PICS). In turn, all PICS indicators should be reflected in the Test Cases (TCs) either directly or indirectly.

A direct reference is a specific test for the attribute cited in the text. For example, a minimum value for a given parameter may be an entry in the TCs. Indirect test coverage may be appropriate if the existence of the attribute is requisite for passing a higher level test.

1.2 MUST

The word *must* shall not be used when stating mandatory requirements. *Must* is used only to describe unavoidable situations and is seldom appropriate for the text of a Specification.

An example of an appropriate use of the term *must* is: “the Bluetooth radios must be in range of each other to communicate”.

1.3 WILL

The use of the word *will* shall not be used when stating mandatory requirements. The term *will* is only used in statements of fact. As with the term *must*, *will* is not generally applicable to the description of a protocol. An example of appropriate use of will is: “when power is removed from the radio, it can be assumed that communications will fail”

1.4 SHOULD

Should equals *is recommended that*. The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable without mentioning or excluding others. Alternatively it may indicate that a certain course of action is preferred but not necessarily required. Finally, in the negative form, it indicates a certain course of action is deprecated but not prohibited.

In the Bluetooth Specification the term designates an optional attribute that may require an entry in the PICS.

Explicit specification of alternatives should be done when using *should*.

1.5 MAY

The word *may* is used to indicate a course of action permissible within the limits of the specification. The term *may* equals *is permitted*. This is generally used when there is a single, optional attribute described, but multiple alternatives may be cited.

The use of *may* implies an optional condition in the PICS and therefore may need to be reflected in the corresponding test cases.

1.6 CAN

The word *can* is used for statements of possibility and capability, whether material, physical, or causal. The term *can* equals *is able to*.

The term *can* shall be used only in informative text. It describes capabilities by virtue of the rules established by normative text.



ETN1009
140/349

Specification Volume 3

SPECIFICATION OF THE *BLUETOOTH* SYSTEM

Experience More



Core System Package

[Host volume]

Covered Core Package version:
4.0

Bluetooth®

ETN1009
141/349

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 274
1.1 Scope 274
1.2 Symbols and Conventions 275
1.2.1 Requirement Status Symbols 275
1.2.2 Signaling diagram conventions 276
1.2.3 Notation for Timers and Counters 276
2 Profile Overview 277
2.1 Profile Stack 277
2.2 Profile Roles 277
2.2.1 Roles when Operating over BR/EDR Physical Channel 277
2.2.2 Roles when Operating over an LE Physical Channel 278
2.3 User Requirements and Scenarios 280
2.4 Profile Fundamentals 281
2.5 Conformance 281
2.6 Other Requirements 281
3 User Interface Aspects 282
3.1 The User Interface Level 282
3.2 Representation of Bluetooth Parameters 282
3.2.1 Bluetooth Device Address (BD_ADDR) 282
3.2.2 Bluetooth Device Name (the user-friendly name) 283
3.2.3 Bluetooth Passkey (Bluetooth PIN) 284
3.2.4 Class of Device 286
3.3 Pairing 286
4 Modes 287
4.1 Discoverability Modes 287
4.1.1 Non-discoverable Mode 288
4.1.2 Limited Discoverable Mode 288
4.1.3 General Discoverable Mode 289
4.2 Connectability Modes 291
4.2.1 Non-connectable Mode 291
4.2.2 Connectable Mode 291
4.3 Bondable Modes 293
4.3.1 Non-bondable Mode 293
4.3.2 Bondable Mode 293
5 Security Aspects 295
5.1 Authentication 295

5.1.1 Purpose 295
5.1.2 Term on UI level 295
5.1.3 Procedure 296
5.1.4 Conditions 296
5.2 Security Modes 296
5.2.1 Legacy Security Modes 297
5.2.2 Security Mode 4 (service level enforced security) 299
6 Idle Mode Procedures 314
6.1 General Inquiry 314
6.1.1 Purpose 314
6.1.2 Term on UI level 314
6.1.3 Description 315
6.1.4 Conditions 315
6.2 Limited Inquiry 315
6.2.1 Purpose 315
6.2.2 Term on UI level 316
6.2.3 Description 316
6.2.4 Conditions 316
6.3 Name Discovery 317
6.3.1 Purpose 317
6.3.2 Term on UI level 317
6.3.3 Description 317
6.3.4 Conditions 318
6.4 Device Discovery 318
6.4.1 Purpose 318
6.4.2 Term on UI Level 318
6.4.3 Description 319
6.4.4 Conditions 319
6.5 Bonding 319
6.5.1 Purpose 319
6.5.2 Term on UI level 319
6.5.3 Description 320
6.5.4 Conditions 321
7 Establishment Procedures 323
7.1 Link Establishment 323
7.1.1 Purpose 323
7.1.2 Term on UI Level 323
7.1.3 Description 323

7.1.4 Conditions 325
7.2 Channel Establishment 326
7.2.1 Purpose 326
7.2.2 Term on UI level 326
7.2.3 Description 326
7.2.4 Conditions 327
7.3 Connection Establishment 328
7.3.1 Purpose 328
7.3.2 Term on UI level 328
7.3.3 Description 328
7.3.4 Conditions 329
7.4 Establishment of Additional Connection 329
8 Extended Inquiry Response Data Format 330
8.1 EIR Data Type Definitions 331
8.1.1 Service Class UUIDs 331
8.1.2 Local Name 332
8.1.3 Flags 332
8.1.4 Manufacturer Specific Data 332
8.1.5 TX Power Level 333
8.1.6 Secure Simple Pairing Out of Band (OOB) 333
8.2 Example Extended Inquiry Response 333
9 Operational Modes and Procedures For Use On LE Physical Channels 335
9.1 Broadcast Mode and Observation Procedure 335
9.1.1 Broadcast Mode 335
9.1.2 Observation Procedure 336
9.2 Discovery Modes and Procedures 337
9.2.1 Requirements 338
9.2.2 Non-Discoverable Mode 338
9.2.3 Limited Discoverable Mode 338
9.2.4 General Discoverable Mode 340
9.2.5 Limited Discovery Procedure 341
9.2.6 General Discovery Procedure 343
9.2.7 Name Discovery Procedure 344
9.3 Connection Modes and Procedures 344
9.3.1 Requirements 346
9.3.2 Non-Connectable Mode 346
9.3.3 Directed Connectable Mode 347
9.3.4 Undirected Connectable Mode 349

9.3.5 Auto Connection Establishment Procedure 351
9.3.6 General Connection Establishment Procedure 354
9.3.7 Selective Connection Establishment Procedure 356
9.3.8 Direct Connection Establishment Procedure 357
9.3.9 Connection Parameter Update Procedure 358
9.3.10 Terminate Connection Procedure 358
9.4 Bonding Modes and Procedures 359
9.4.1 Requirements 359
9.4.2 Non-Bondable Mode 359
9.4.3 Bondable Mode 360
9.4.4 Bonding Procedure 360
10 LE Security Aspects 362
10.1 Requirements 362
10.2 LE Security Modes 362
10.2.1 LE Security Mode 1 362
10.2.2 LE Security Mode 2 363
10.2.3 Mixed Security Modes Requirements 363
10.3 Authentication Procedure 363
10.3.1 Receiving a Service Request 364
10.3.2 Initiating a Service Request 365
10.4 Data Signing 368
10.4.1 Connection Data Signing Procedure 368
10.4.2 Authenticate Signed Data Procedure 368
10.5 Authorization Procedure 369
10.6 Encryption Procedure 369
10.7 Privacy Feature 369
10.7.1 Privacy Feature in a Peripheral 370
10.7.2 Privacy Feature in a Central 371
10.8 Random Device Address 371
10.8.1 Static Address 371
10.8.2 Private address 372
11 Advertising and Scan Response Data Format 375
11.1 AD Type Definitions 375
11.1.1 Service UUIDs 376
11.1.2 Local Name 376
11.1.3 Flags 376
11.1.4 Manufacturer Specific Data 376
11.1.5 TX Power Level 377
11.1.6 Security Manager Out of Band (OOB) 377

11.1.7 Security Manager TK Value 377
11.1.8 Slave Connection Interval Range 377
11.1.9 Service Solicitation 377
11.1.10 Service Data 377
11.2 Example Advertising Data 378

12 GAP Characteristics for Low Energy 379
12.1 Device Name Characteristic 379
12.2 Appearance Characteristic 380
12.3 Peripheral Privacy Flag Characteristic 380
12.4 Reconnection Address Characteristic 380
12.5 Peripheral Preferred Connection Parameters Characteristic 381

13 BR/EDR/LE Operation Modes and Procedure 383
13.1 Modes 383
13.1.1 Discoverability 384
13.1.2 Connectability 385
13.1.3 Bondable Modes 385
13.2 Idle Mode Procedures 385
13.2.1 General Discovery Procedure 386
13.2.2 Limited Discovery Procedure 386
13.2.3 Device Discovery Procedure 387
13.2.4 Name Discovery 387
13.3 Establishment Procedures 388
13.3.1 Link Establishment 388
13.4 SDP Interoperability Requirements 388

14 BR/EDR/LE Security Aspects 390

15 Definitions 391
15.1 General Definitions 391
15.2 Connection-related Definitions 391
15.3 Device-related Definitions 392
15.4 Procedure-related Definitions 393
15.5 Security-related Definitions 393

16 Appendix A (Normative): Timers and Constants 395

17 Appendix B (Informative): Information Flows of Related Procedures 398
17.1 LMP – Authentication 398
17.2 LMP – Pairing 399
17.3 Service Discovery 400
17.4 Generating a Resolvable Private Address 400
17.5 Resolving a Resolvable Private Address 400

18 Appendix C (Normative): EIR and AD Formats 401

18.1 Flags 401
18.2 Service 402
18.3 Local Name 402
18.4 TX Power Level 402
18.5 Simple Pairing Optional OOB Tags 403
18.6 Security Manager TK Value 403
18.7 Security Manager OOB Flags 403
18.8 Slave Connection Interval Range 404
18.9 Service Solicitation 404
18.10 Service Data 404
18.11 Manufacturer Specific Data 405

19 References 406

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 a 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.

This profile defines three device types based on the supported Core Configurations as defined in [Vol. 0], Part B Section 3.1. The device types are shown in Table 1.1:

Device Type	Description
BR/EDR	Devices that support the “Basic Rate” Core Configuration (see [Vol. 0], Part B Section 4.1)
LE only	Devices that support the “Low Energy” Core Configuration (see [Vol. 0], Part B Section 4.4)
BR/EDR/LE	Devices that support the “Basic Rate and Low Energy Combined” Core Configuration (see [Vol. 0], Part B Section 4.5)

Table 1.1: Device Types

Devices of the LE-only and BR/EDR/LE device types are capable of operating over an LE physical channel.

The terms physical links and physical channels as defined in [Vol. 1], Part A Section 3.3 and Section 4 are used in this specification.

The requirements for device types are shown in Table 1.2.

Device Types	Sections	Support
BR/EDR	1-8, 15-18	C1
LE-only	1-3, 9-12, 15-18	C1
BR/EDR/LE	1-18	C1
C1: Mandatory to support only one device type		

Table 1.2: Requirements for device types

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);

‘E’ for excluded within profile role (used for capabilities that may be supported by the unit but shall never be used in the profile role);

’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 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

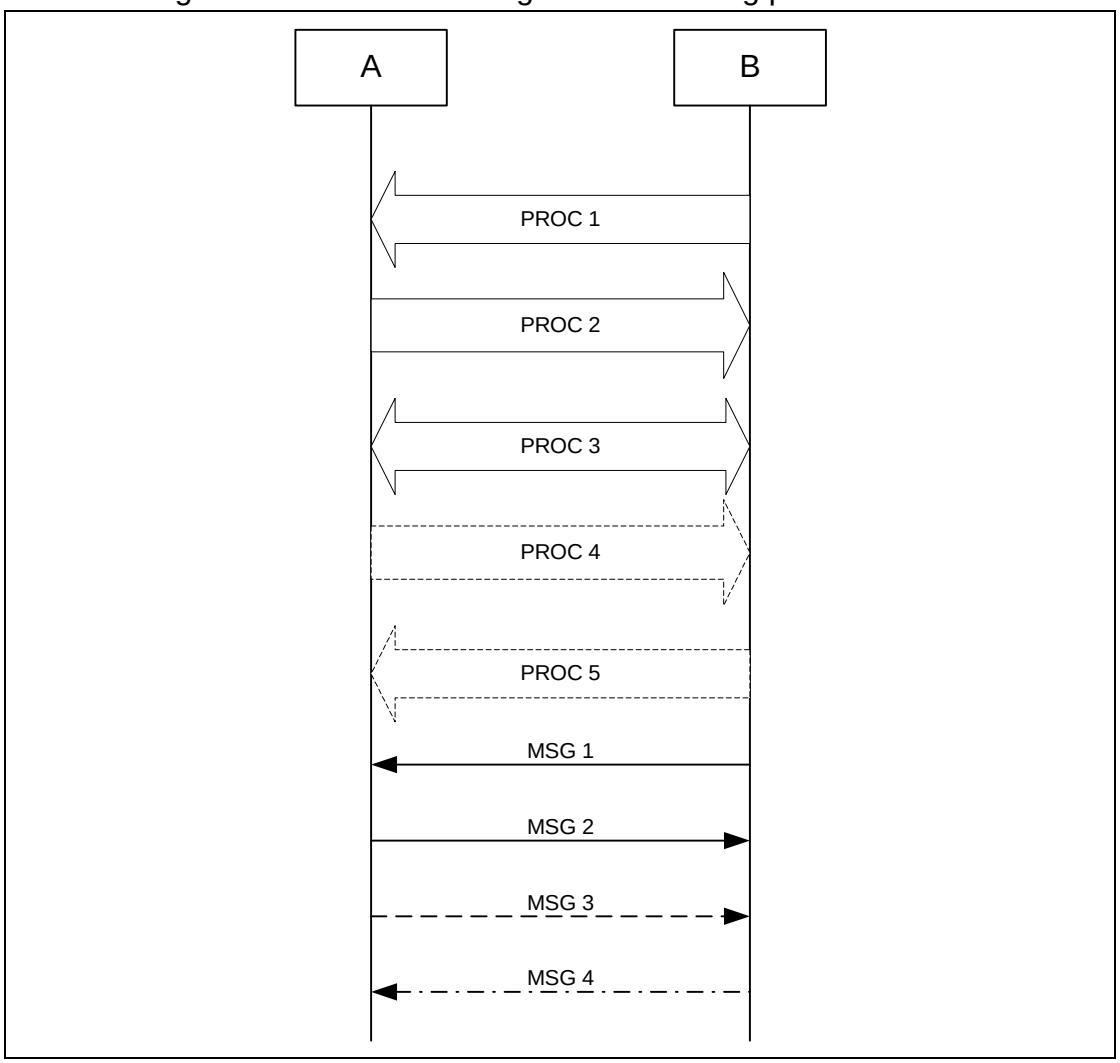


Figure 1.1: Arrows used in signaling diagrams

In Figure 1.1, 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

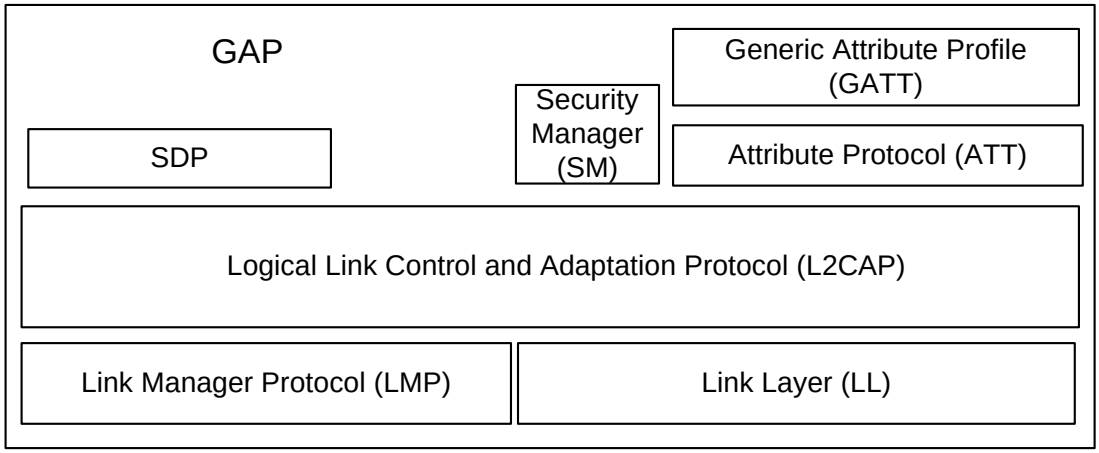


Figure 2.1: Relationship of GAP with lower layers of the Bluetooth architecture

The purpose of this profile is to describe:

- Profile roles
- Discoverability modes and procedures
- Connection modes and procedures
- Security modes and procedures

2.2 PROFILE ROLES

2.2.1 Roles when Operating over BR/EDR Physical Channel

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.

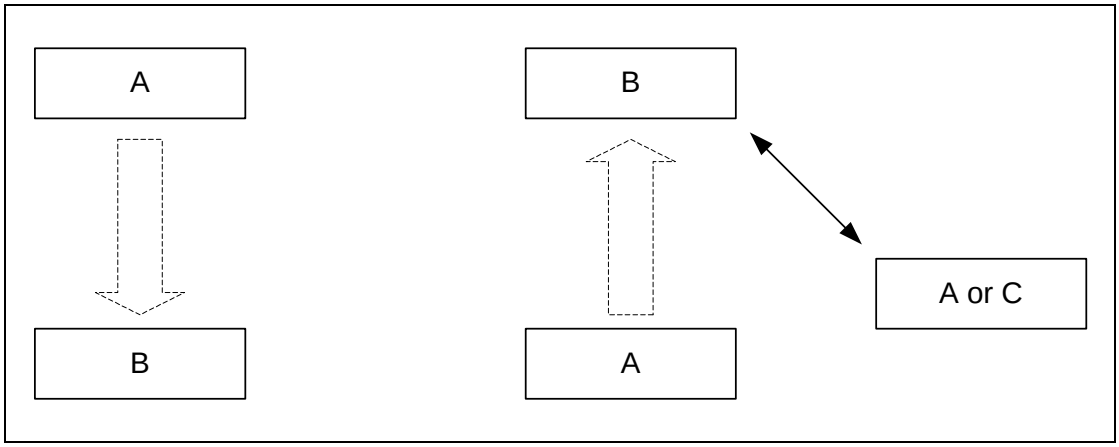


Figure 2.2: This profile covers procedures initiated by one device (A) towards another device (B) that may or may not have an existing Bluetooth link active.

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.

2.2.2 Roles when Operating over an LE Physical Channel

There are four GAP roles defined for devices operating over an LE physical channel:

- Broadcaster
- Observer
- Peripheral
- Central

2.2.2.1 Broadcaster Role

A device operating in the Broadcaster role is a device that sends advertising events as described in [Vol. 6], Part B Section 4.4.2. A device operating in the Broadcaster role is referred to as a Broadcaster and shall have a transmitter and may have a receiver.

2.2.2.2 Observer Role

A device operating in the Observer role is a device that receives advertising events as described in [Vol. 6], Part B Section 4.4.3. A device operating in the Observer role is referred to as an Observer and shall have a receiver and may have a transmitter.

2.2.2.3 Peripheral Role

Any device that accepts the establishment of an LE physical link using any of the connection establishment procedures as defined in Section 13.3 is referred to as being in the Peripheral role. A device operating in the Peripheral role will be in the Slave role in the Link Layer Connection State as described in [Vol. 6], Part B Section 4.5. A device operating in the Peripheral role is referred to as a Peripheral. A Peripheral shall have both a transmitter and a receiver.

2.2.2.4 Central Role

A device that supports the Central role initiates the establishment of a physical connection. A device operating in the Central role will be in the Master role in the Link Layer Connection State as described in [Vol. 6], Part B Section 4.5. A device operating in the Central role is referred to as a Central. A Central shall have both a transmitter and a receiver.

2.2.2.5 Concurrent Operation in Multiple GAP Roles

A device may operate in multiple GAP roles concurrently if supported by the Controller. The Host should read the supported Link Layer States and State combinations from the Controller before any procedures or modes are used.

Only supported and allowed Link Layer States and State combinations may be used as described in [Vol. 6], Part B Section 1.1.1. The Physical Layer and Link Layer functionalities of each GAP role are shown in Table 2.1.

	GAP Roles when Operating over an LE Physical Channel			
	Broadcaster	**Observer**	**Peripheral**	**Central**
Physical Layer functionality				
• Transmitter Characteristics	M	O	M	M
• Receiver Characteristics	O	M	M	M
Link Layer functionality				
States:				
• Standby State	M	M	M	M
• Advertising State	M	E	M	E
• Scanning State	E	M	E	M
• Initiating State	E	E	E	M
Connection State:				

• Slave Role	E	E	M	E
• Master Role	E	E	E	M
Advertising event types:				
• Connectable undirected event	E	E	M	E
• Connectable directed event	E	E	O	E
• Non-connectable undirected event	M	E	O	E
• Scannable undirected event	O	E	O	E
Scanning type:				
• Passive scanning	E	M	E	O
• Active scanning	E	O	E	C1
Link Layer control procedures:				
• Connection Update procedure	E	E	M	M
• Channel Map Update procedure	E	E	M	M
• Encryption procedure	E	E	O	O
• Feature Exchange procedure	E	E	M	M
• Version Exchange procedure	E	E	M	M
• Termination procedure	E	E	M	M
C1: If passive scanning is supported then active scanning is optional, otherwise active scanning is mandatory.				

Table 2.1: Physical Layer and Link Layer functionality for GAP roles when operating over an LE physical channel

2.3 USER REQUIREMENTS AND SCENARIOS

The Bluetooth user should, where expected, 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 discovered. 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 a mode that allows them to accept connections and service requests.

2.5 CONFORMANCE

Bluetooth devices shall conform to this profile to ensure basic interoperability.

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.

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.

2.6 OTHER REQUIREMENTS

Devices of the BR/EDR or BR/EDR/LE device types shall implement and use SDP for service discovery over the BR/EDR transport. Devices of the BR/EDR devices type that implement the Attribute Protocol shall also implement the Generic Attribute Profile.

Devices of the BR/EDR/LE and LE-only device types shall implement and use GATT for service discovery of the LE transport. Devices of the LE-only device type shall implementation ATT.

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.

A device that supports discovery and connections over both the BR/EDR and LE physical channels should have a unified user interface that hides the differences between the underlying technologies from the user.

3.2 REPRESENTATION OF BLUETOOTH PARAMETERS

3.2.1 Bluetooth Device Address (BD_ADDR)

3.2.1.1 Definition

A device shall be identified by a Bluetooth device address. When the device is operating over a BR/EDR physical channel the device address shall be the BD_ADDR (see [Vol. 2], Part B Section 1.2). The device address is obtained from a remote device during the device discovery procedure (see Section 6.4 and Section 13.2.3).

The BD_ADDR shall be created in accordance with Section 9.2 (“48-bit universal LAN MAC addresses”) of the IEEE 802-2001 standard (http://standards.ieee.org/getieee802/download/802-2001.pdf) and using a valid Organizationally Unique Identifier (OUI) obtained from the IEEE Registration Authority (see http://standards.ieee.org/regauth/oui/forms/ and sections 9 and 9.1 of the IEEE 802-2001 specification).

3.2.1.1.1 Bluetooth Device Address in an LE-only Device Type

An LE-only device type shall use either a public or a random device address. The public device address shall be set to the BD_ADDR. A random device address is defined in Section 10.8. An LE-only device type may support the privacy feature as defined in Section 10.7 which requires the support of private device addresses as defined in Section 10.8.2.

3.2.1.1.2 Bluetooth Device Address in a BR/EDR/LE Device Type

A BR/EDR/LE device type shall use the BD_ADDR on the BR/EDR physical channel and a public device address over the LE physical channel. The public device address shall be set to the BD_ADDR. A BR/EDR/LE device type shall be capable of connecting to an LE-only or another BR/EDR/LE device type that is identified by a random device address.

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 (see [Vol. 2], Part B Section 1.2). On the Link Layer the public and random device address are represented as 48-bit addresses.

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-exposes to remote devices. For a device supporting the BR/EDR device type, the name is a character string returned in the LMP_name_res in response to an LMP_name_req. For a device supporting the LE-only device type, the name is a character string held in the Device Name characteristic as defined in Section 12.1.

3.2.2.1.1 Bluetooth Device Name in a Device with BR/EDR/LE Device Type

A BR/EDR/LE device type shall have a single Bluetooth device name which shall be identical irrespective of the physical channel used to perform the name discovery procedure.

For the BR/EDR physical channel the name is received in the LMP_name_res. For the LE physical channel the name can be read from the Device Name characteristic as defined in Section 12.1.

Note: The Device Name Characteristic of the local device can be read by a remote device using ATT over BR/EDR if the local device supports ATT over BR/EDR.

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 (see [Vol. 2], Part C Section 4.3.5). It shall be encoded according to UTF-8 (i.e., name entered on the UI level may be down to 82 characters outside the Unicode range 0x00-0x7F are used).

A device cannot 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 passkey may be used to authenticate two Bluetooth devices with each other during the creation of a mutual link key via the pairing procedure. The passkey may be used in the pairing procedures to generate the initial link key.

The PIN may be entered on the 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

There are a number of different representations of the Bluetooth passkey. At a high level there are two distinct representations: one used with the Secure Simple Pairing and Security Manager, and another used with legacy pairing (where it is generally referred to as the Bluetooth PIN).

For Secure Simple Pairing and Security Manager, the Bluetooth passkey is a 6-digit numerical value. It is represented as integer value in the range 0x00000000 – 0x000F423F (000000 to 999999). The numeric value may be used as the input to the Authentication Stage 1 for Secure Simple Pairing Passkey Entry (see [Vol. 2], Part H Section 7.2.3), or as the TK value in the Security Manager for the process defined in [Vol. 3], Part H Section 2.3.5.

For legacy pairing (see Section 17.2 on page 399), 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 the UI level. The transformation from PINUI to PINBB shall be according to UTF-8. PINBB may be 128 bits (16 bytes).

PIN codes may be up to 16 characters. In order to take advantage of the full level of security all PINs should be 16 characters long. Variable PINs should be composed of alphanumeric characters chosen from within the Unicode range 0x00-0x7F. If the PIN contains any decimal digits these shall be encoded using the Unicode Basic Latin characters (e.g., code points 0x30 to 0x39) (Note 1).

For compatibility with devices with numeric keypads, fixed PINs shall be composed of only decimal digits, and variable PINs should be composed of only decimal digits.

If a device supports entry of characters outside the Unicode range 0x00-0x7F, other Unicode code points may be used (Note 2), except the Halfwidth and Fullwidth Forms from within the Unicode range FF00 - FFEF shall not be used (Note 3).

Examples:

User-entered code	**Corresponding PIN_{BB}[0..length-1] (value as a sequence of octets in hexadecimal notation)**
’0196554200906493’	length = 16, value = 0x30 0x31 0x39 0x36 0x35 0x35 0x34 0x32 0x30 0x30 0x39 0x30 0x36 0x34 0x39 0x33
’Børnelitteratur’	length = 16, value = 0x42 0xC3 0xB8 0x72 0x6e 0x65 0x6c 0x69 0x74 0x74 0x65 0x72 0x61 0x74 0x75 0x72

Note 1: This is to prevent interoperability problems since there are decimal digits at other code points (e.g., the Fullwidth digits at code points 0xff10 to 0xff19).

Note 2: Unicode characters outside the Basic Latin range (0x00 - 0x7F) encode to multiple bytes; therefore, when characters outside the Basic Latin range are used the maximum number of characters in the PINUI will be less than 16. The second example illustrates a case where a 16 character string encodes to 15 bytes because the character ø is outside the Basic Latin range and encodes to two bytes (0xC3 0xB8).

Note 3: This is to prevent interoperability problems since the Halfwidth and Fullwidth forms contain alternative variants of ASCII, Katakana, Hangul, punctuation and symbols. All of the characters in the Halfwidth and Fullwidth forms have other more commonly used Unicode code points.

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 [7].

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 [7]. The UI-level representation of the information in the Class of Device is implementation specific.

3.2.4.4 Usage

Some devices provide more than one service and a given service may be provided by different device types. Therefore, the device type does not have a one-to-one relationship with services supported. The major and minor device class field should not be used to determine whether a device supports any specific service(s). It may be used as an indication of devices that are most likely to support desired services before service discovery requests are made, and it may be used to guide the user when selecting among several devices that support the same service.

3.3 PAIRING

Pairing over a BR/EDR physical link is defined on LMP level (LMP pairing, see 17.2). Pairing over an LE physical link is defined by the Security Manager specification ([Vol. 3], Part H Section 2.3). 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
Discoverability modes:	4.1	
Non-discoverable mode		C1
Discoverable mode		O
Limited discoverable mode		C3
General discoverable mode		C4
Connectability modes:	4.2	
Non-connectable mode		O
Connectable mode		M
Bondable modes:	4.3	
Non-bondable mode		O
Bondable 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 bondable mode is mandatory, otherwise optional.		
C3: If Discoverable mode is supported, support for Limited Discoverable mode is mandatory, otherwise optional.		
C4: If Discoverable mode is supported, support for General Discoverable 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 [Vol. 2], Part B Section 8.4.

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 (for example, reserved synchronous slots should have priority over response packets, so that synchro-

nous links may prevent a response from being returned). A Bluetooth device that does not respond to inquiry is called a silent device.

After being made discoverable, the Bluetooth device shall be discoverable for at least $T_{GAP}(103)$.

The speed of discovery is dependent on the configuration of the inquiry scan interval and inquiry scan type of the Bluetooth device. The Host is able to configure these parameters based on trade-offs between power consumption, bandwidth and the desired speed of discovery.

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_SCAN 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.

- *Parallel scanning*

  When a Bluetooth device is in limited discoverable mode and when discovery speed is more important than power consumption or bandwidth, it is recommended that the Bluetooth device enter the INQUIRY_SCAN state at least every $T_{GAP}(105)$ and that Interlaced Inquiry scan is used.

  If, however, power consumption or bandwidth is important, but not critical, it is recommended that the Bluetooth device enter the INQUIRY_SCAN state at least every $T_{GAP}(102)$ and Interlaced Inquiry scan is used.

  When power consumption or bandwidth is critical it is recommended that the Bluetooth device enter the INQUIRY_SCAN state at least every $T_{GAP}(102)$ and Non Interlaced Inquiry scan is used.

In all cases the Bluetooth device shall enter the INQUIRY_SCAN state at least once in $T_{GAP}(102)$ and scan for the GIAC and the LIAC for at least $T_{GAP}(101)$.

When either a SCO or eSCO link is in operation, it is recommended to use interlaced scan to significantly decrease the discoverability time.

- *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 [7].

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 and when discovery speed is more important than power consumption or bandwidth, it is recommended that the Bluetooth device enter the INQUIRY_SCAN state at least every $T_{GAP}(105)$ and that Interlaced Inquiry scan is used.

If, however, power consumption or bandwidth is important, but not critical, it is recommended that the Bluetooth device enter the INQUIRY_SCAN state at least every $T_{GAP}(102)$ and Interlaced Inquiry scan is used.

When power consumption or bandwidth is critical it is recommended that the Bluetooth device enter the INQUIRY_SCAN state at least every $T_{GAP}(102)$ and Non-interlaced Inquiry scan is used.

In all cases the Bluetooth device shall enter the INQUIRY_SCAN state at least once in $T_{GAP}(102)$ and scan for the GIAC for at least $T_{GAP}(101)$.

When either a SCO or eSCO link is in operation, it is recommended to use interlaced scan to significantly decrease the discoverability time.

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 connectable mode. Paging is defined in [Vol. 2], Part B Section 8.3.

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.

The speed of connections is dependent on the configuration of the page scan interval and page scan type of the Bluetooth device. The Host is able to configure these parameters based on trade-offs between power consumption, bandwidth and the desired speed of connection.

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. The device makes page scan using the Bluetooth device address, BD_ADDR. Connection speed is a trade-off between power consumption / available bandwidth and speed. The Bluetooth Host is able to make these trade-offs using the Page Scan interval, Page Scan window, and Interlaced Scan parameters.

R0 page scanning should be used when connection speeds are critically important and when the paging device has a very good estimate of the Bluetooth clock. Under these conditions it is possible for paging to complete within two times the page scan window. Because the page scan interval is equal to the page scan window it is not possible for any other traffic to go over the Bluetooth link when using R0 page scanning. In R0 page scanning it is not possible to use interlaced scan. R0 page scanning is the highest power consumption mode of operation.

When connection times are critical but the other device either does not have an estimate of the Bluetooth clock or when the estimate is possibly out of date, it is better to use R1 page scanning with a very short page scan interval, $T_{GAP}(106)$, and Interlaced scan. This configuration is also useful to achieve nearly the same connection speeds as R0 page scanning but using less power

consumption and leaving bandwidth available for other connections. Under these circumstances it is possible for paging to complete within $T_{GAP}(106)$. In this case the Bluetooth device shall page scan for at least $T_{GAP}(101)$.

When connection times are important but not critical enough to sacrifice significant bandwidth and/or power consumption it is recommended to use either $T_{GAP}(107)$ or $T_{GAP}(108)$ for the scanning interval. Using Interlaced scan will reduce the connection time by half but may use twice the power consumption. Under these circumstances it is possible for paging to complete within one or two times the page scanning interval depending on whether Interlaced Scan is used. In this case the Bluetooth device shall page scan for at least $T_{GAP}(101)$.

In all cases the Bluetooth device shall enter the PAGE_SCAN state at least once in $T_{GAP}(102)$ and scan for at least $T_{GAP}(101)$.

The page scan interval, page scan window size, and scan type for the six scenarios are described in Table 4.2:

Scenario	Page Scan Interval	Page Scan Window	Scan Type
R0 (1.28s)	$T_{GAP}(107)$	$T_{GAP}(107)$	Normal scan
Fast R1 (100ms)	$T_{GAP}(106)$	$T_{GAP}(101)$	Interlaced scan
Medium R1 (1.28s)	$T_{GAP}(107)$	$T_{GAP}(101)$	Interlaced scan
Slow R1 (1.28s)	$T_{GAP}(107)$	$T_{GAP}(101)$	Normal scan
Fast R2 (2.56s)	$T_{GAP}(108)$	$T_{GAP}(101)$	Interlaced scan
Slow R2 (2.56s)	$T_{GAP}(108)$	$T_{GAP}(101)$	Normal scan

Table 4.2: Page scan parameters for connection speed scenarios

When either a SCO or eSCO link is in operation, it is recommended to use interlaced scan to significantly decrease the connection time.

4.2.2.2 Term on UI-level

Bluetooth device is ‘connectable’ or in ‘connectable mode’.

4.3 BONDABLE MODES

With respect to bonding, a Bluetooth device shall be either in non-bondable mode or in bondable mode. In bondable mode the Bluetooth device accepts bonding initiated by the remote device, and in non-bondable mode it does not.

4.3.1 Non-bondable Mode

4.3.1.1 Definition

When a Bluetooth device is in non-bondable mode it shall not accept a pairing request that results in bonding. Devices in non-bondable mode may accept connections that do not request or require bonding.

A device in non-bondable mode shall respond to a received LMP_in_rand with LMP_not_accepted with the reason *pairing not allowed*. An HCI-compliant host stack shall respond to an HCI_PIN_Code_Request event with the HCI_PIN_Code_Request_Negative_Reply command.

When both devices support Secure Simple Pairing and are in non-bondable mode, the local host shall respond to an IO capability request with the Authentication_Requirements parameter requesting dedicated bonding or general bonding with a negative response. An HCI-compliant host stack shall respond to an HCI_IO_Capabilities_Request event with an HCI_IO_Capabilities_Request_Negative_Reply command.

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 Bondable Mode

4.3.2.1 Definition

When a Bluetooth device is in bondable mode, and Secure Simple Pairing is not supported by either the local or remote device, the local device shall respond to a received LMP_in_rand with LMP_accepted (or with LMP_in_rand if it has a fixed PIN). An HCI-compliant host stack shall respond to an HCI_PIN_Code_Request event with the HCI_PIN_Code_Request_Reply command.

When both devices support Secure Simple Pairing, the local host shall respond to a user confirmation request with a positive response. An HCI-compliant host stack shall respond to an HCI_User_Confirmation_Request event with an HCI_User_Confirmation_Request_Reply command or an HCI_User_Passkey_Request event with an HCI_User_Passkey_Request_Reply command.

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	M
2	Security modes	5.2	
	Security mode 1		E
	Security mode 2		O.1
	Security mode 3		E
	Security mode 4		M
O.1: Security Mode 2 may only be used for backwards compatibility when the remote device does not support Secure Simple Pairing.			

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

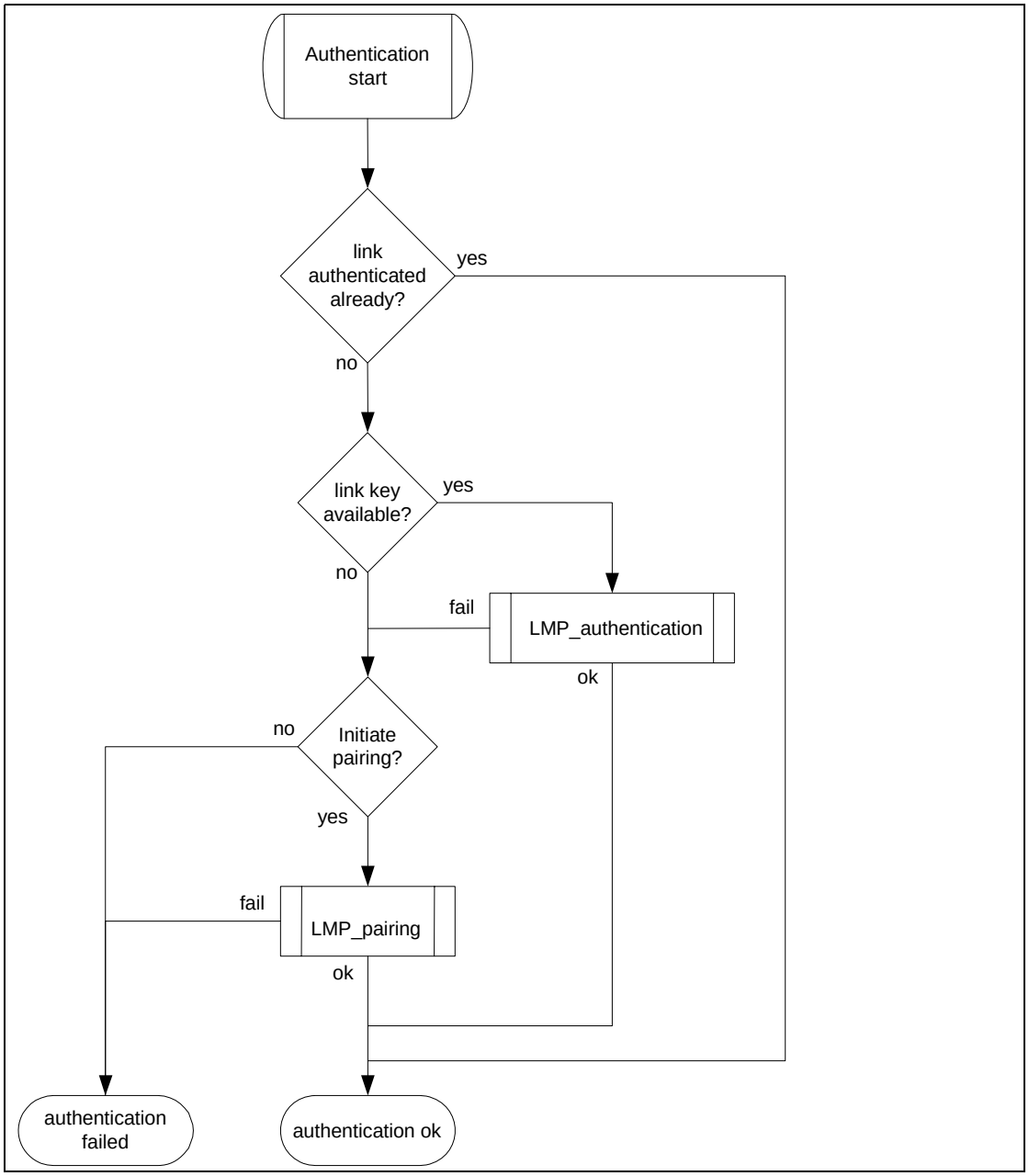


Figure 5.1: Definition of the generic authentication procedure.

5.1.4 Conditions

The local device shall initiate authentication after link establishment. The remote device may initiate security during or after link establishment.

5.2 SECURITY MODES

The following flow chart provides an overview of the channel establishment procedures.

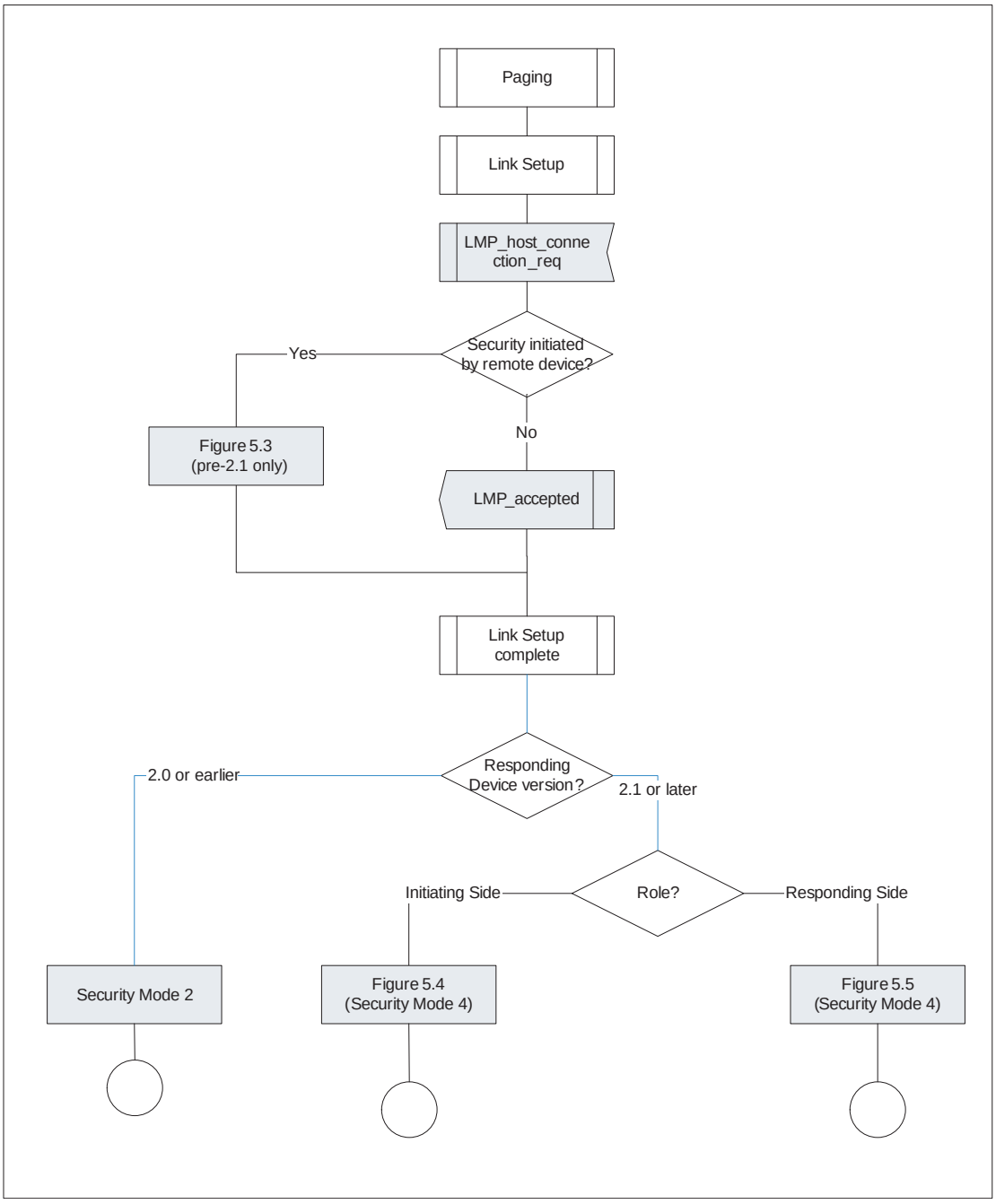


Figure 5.2: Channel establishment with security.

A device may support two security modes simultaneously: security mode 2 for backwards compatibility with remote devices that do not support Secure Simple Pairing and security mode 4 for devices that support Secure Simple Pairing.

5.2.1 Legacy Security Modes

Legacy security modes apply to those devices with a Controller or a Host that does not support SSP.

5.2.1.1 Security mode 1 (non-secure)

When a remote Bluetooth device is in security mode 1 it will never initiate any security procedure (i.e., it will never send LMP_au_rand, LMP_in_rand or LMP_encryption_mode_req).

5.2.1.2 Security mode 2 (service level enforced security)

When a remote Bluetooth device is in security mode 2 it will 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 [6].) 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.

5.2.1.3 Security mode 3 (link level enforced security)

When a remote Bluetooth device is in security mode 3 it will initiate security procedures before it sends LMP_setup_complete.

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).

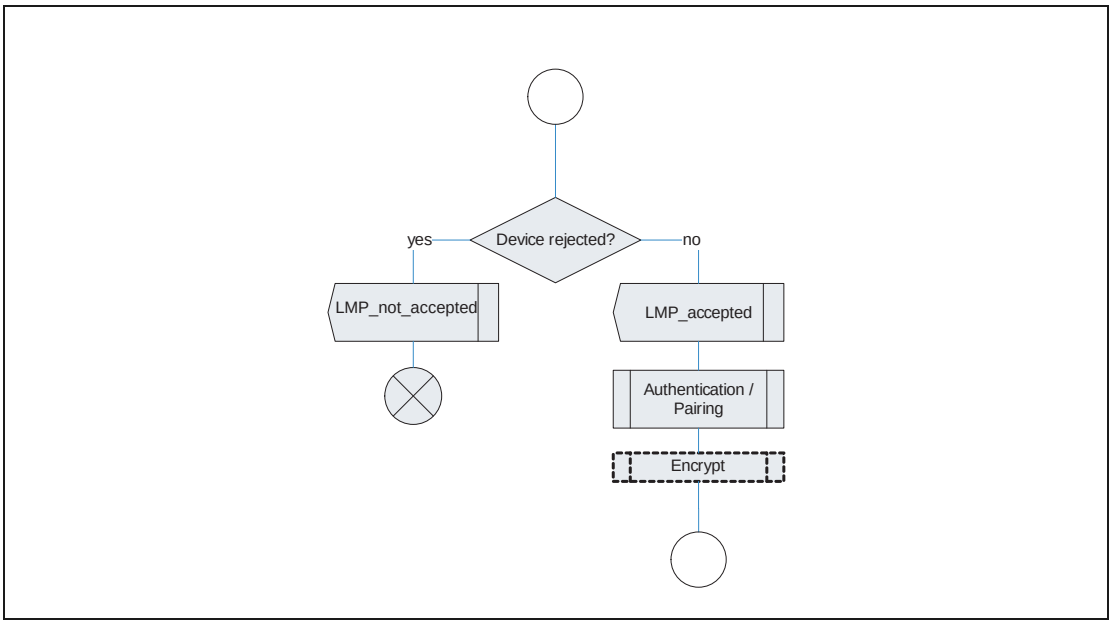


Figure 5.3: Security mode 3 with a legacy remote device

5.2.2 Security Mode 4 (service level enforced security)

A Bluetooth device in security mode 4 shall classify the security requirements of its services using at least the following attributes (in order of decreasing security):

- Authenticated link key required
- Unauthenticated link key required
- Security optional – SDP only. Limited to specific services (see Section 5.2.2.8).

An authenticated link key is a link key where either the numeric comparison, out-of-band, or passkey entry simple pairing association models were used. An authenticated link key has protection against man-in-the-middle (MITM) attacks. To ensure that an authenticated link key is created during the Simple Pairing procedure, the Authentication_Requirements parameter should be set to one of the MITM Protection Required options. An *unauthenticated link key* is a link key where the just works Secure Simple Pairing association model was used. An unauthenticated link key does not have protection against MITM attacks.

When both devices support Secure Simple Pairing, GAP shall require at least an unauthenticated link key and enabling encryption for all connections except those allowed to have security level 0 (see Section 5.2.2.8). A profile or protocol may define services that require more security (e.g., an authenticated link key) or no security (although unencrypted connections are only allowed when connecting to a service allowed to have security level 0). To allow an unauthenticated link key to be created during the Simple Pairing procedure, the

Authentication_Requirements parameter may be set to one of the MITM Protection Not Required options.

When the device is in Bondable Mode, it shall enable Secure Simple Pairing mode prior to entering Connectable Mode or establishing a link.

A Bluetooth device in security mode 4 shall respond to authentication requests during link establishment when the remote device is in security mode 3 for backwards compatibility reasons.

A Bluetooth device in security mode 4 enforces its security requirements before it attempts to access services offered by a remote device and before it grants access to services it offers to remote devices. Service access may occur via L2CAP channels or via channels established by protocols above L2CAP such as RFCOMM.

For services transmitting unicast data over the connectionless L2CAP channel, the transmitting device shall enforce its security requirements prior to sending data. There is no mechanism for a device receiving data via the L2CAP connectionless channel to prevent unencrypted data from being received. Hence, Section 5.2.2.1 addresses unicast connectionless data transmission together with devices initiating connection-oriented channels while Section 5.2.2.2 covers only devices responding to requests for connection-oriented channel establishment but does not cover unicast connectionless data reception.

5.2.2.1 Security for Access to Remote Service (Initiating Side)

When the responding device does not support Secure Simple Pairing, it may disconnect the link while the initiating device is requesting the PIN to be entered by the user. This may occur due to the lack of an L2CAP channel being present for longer than an implementation-specific amount of time (e.g., a few seconds). When this occurs, the initiating device shall allow the user to complete entering the PIN and shall then re-page the remote device and restart the pairing procedure (see [Vol. 2, Part C] Section 4.2.2 on page 249) using the PIN entered.

5.2.2.1.1 Pairing Required for Access to Remote Service

When a Bluetooth device in security mode 4 initiates access to a remote service via a connection-oriented L2CAP channel and a sufficient link-key is not available, the local device shall perform pairing procedures and enable encryption before sending a channel establishment request (L2CAP_ConnectReq or a higher-layer channel establishment request such as that of RFCOMM).

When a Bluetooth device in security mode 4 transmits data to a remote service via the unicast connectionless L2CAP channel and a sufficient link-key is not available, the local device shall perform pairing procedures and enable encryption before transmitting unicast data on the connectionless L2CAP channel.

See Section 5.2.2.8 on page 310 for details on determining whether or not a link key is sufficient.

If pairing does not succeed, the local device shall not send a channel establishment request. The local device may re-try pairing up to three (3) times. If pairing fails three consecutive times, the local device shall disconnect the ACL link with error code 0x05 - Authentication Failure.

If the link key generated is not at least as good as the expected or required type, the local device shall fail the establishment and may disconnect the ACL link with error code 0x05 - Authentication Failure.

5.2.2.1.2 Authentication Required for Access to Remote Service

When a Bluetooth device in security mode 4 initiates access to a remote service via a connection-oriented L2CAP channel and a sufficient link key is available for the remote device, it shall authenticate the remote device and enable encryption before sending a channel establishment request (L2CAP_ConnectReq or higher a layer-channel establishment request such as that of RFCOMM).

When a Bluetooth device in security mode 4 transmits unicast data to a remote service via the connectionless L2CAP channel and security is required for the application and a sufficient link-key is available then the local device shall authenticate the remote device and enable encryption before transmitting unicast data on the L2CAP connectionless channel.

See Section 5.2.2.8 on page 310 for details on determining whether or not a link key is sufficient.

If authentication is required by the service but does not succeed, or if a sufficient link-key is not available, then the local device shall not enable encryption and shall not send a channel establishment request and shall not send any unicast data via the L2CAP connectionless channel for that application. The host may then notify the user and offer to perform pairing.

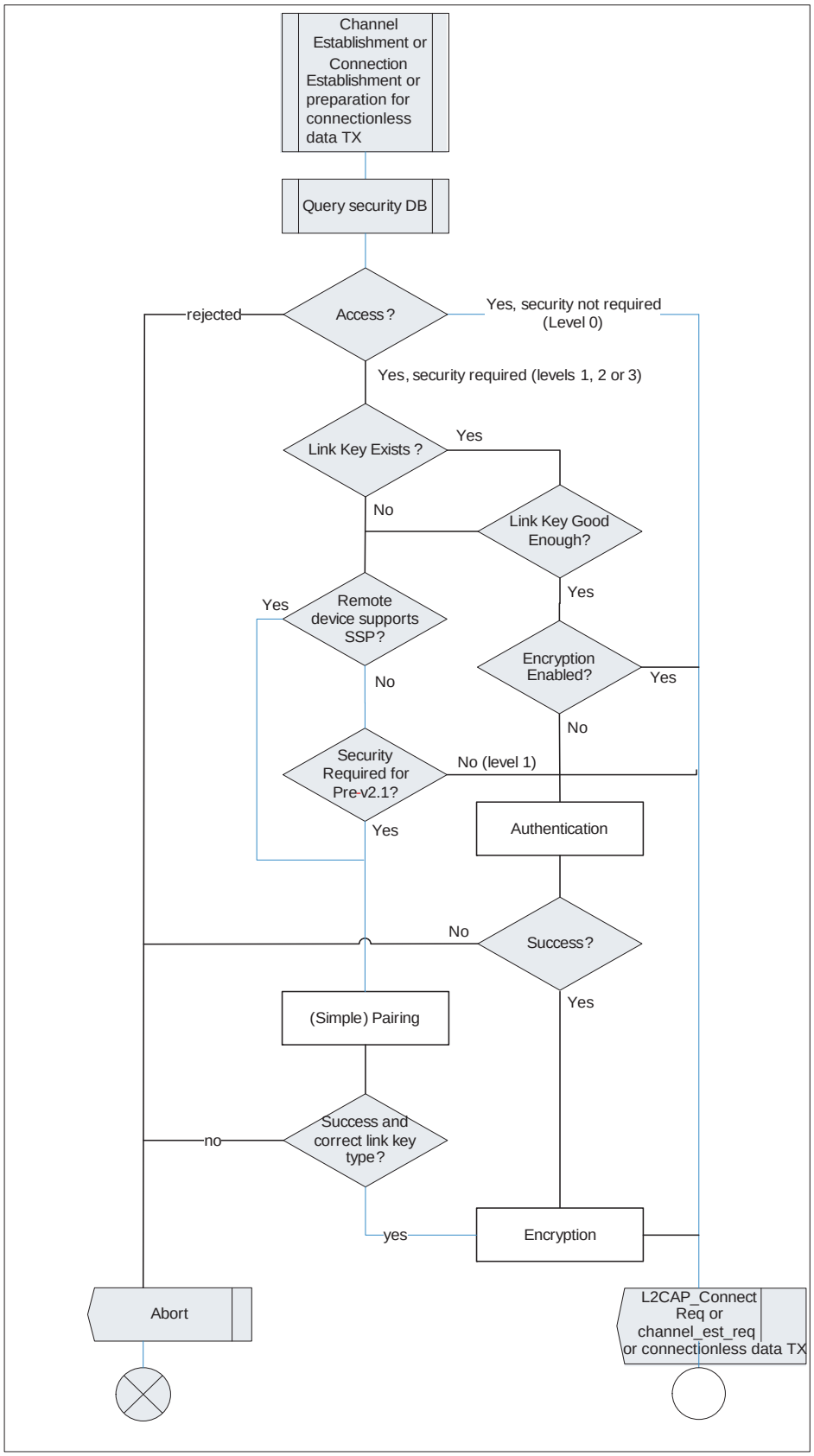


Figure 5.4: Channel establishment using security mode 4 for initiating side

5.2.2.2 Security for Access to Local Service by Remote Device (Responding Side)

When a remote device attempts to access a service offered by a Bluetooth device that is in security mode 4 and a sufficient link key exists and authentication has not been performed the local device shall authenticate the remote device and enable encryption after the channel establishment request is received but before a channel establishment confirmation (L2CAP_ConnectRsp with result code of 0x0000 or a higher-level channel establishment confirmation such as that of RFCOMM) is sent.

When L2CAP is the channel establishment protocol being used for the requested service, an L2CAP_ConnectRsp signaling packet shall be sent by the responding device containing the result code 0x0001 (connection pending) following receipt of an L2CAP_ConnectReq and prior to initiating security procedures which can result in prompting the local user for input (e.g., pairing using a PIN or Secure Simple Pairing using either the Passkey entry or Numerical Comparison association models). This will stop the L2CAP RTX timer on the remote device (which may be as short as 1 second) and will invoke the ERTX timer on the remote device, which is a minimum duration of 60 seconds.

See [Vol. 3, Part A] Section 6.1.7 on page 110, for additional information on L2CAP RTX and ERTX timers. See also [Vol. 3, Part A] Section 4.3 on page 59 for additional information on the L2CAP_ConnectRsp signalling packet, and the defined result codes.

Higher layer channel establishment protocols should be designed to restrict timeouts to be 30 seconds or longer to allow for user input, or provide mechanisms to dynamically extend timeouts when user input may be required.

If authentication or pairing fails when a remote device is attempting to access a local service, the local device shall send a negative response to the channel establishment request (L2CAP_ConnectReq or channel_est_req) indicating a security issue if possible. If the channel establishment protocol is L2CAP, then the result code 0x0003 (connection refused - security block) shall be sent in the L2CAP_ConnectRsp signal.

If the remote device has indicated support for Secure Simple Pairing, a channel establishment request is received for a service other than SDP, and encryption has not yet been enabled, then the local device shall disconnect the ACL link with error code 0x05 - Authentication Failure.

5.2.2.2.1 Pairing Required for Access to Local Service by Remote Device

When a remote device attempts to access a service offered by a Bluetooth device that is in security mode 4 and pairing is required due to the link key being absent or insufficient, the local device shall perform pairing procedures and enable encryption after the channel establishment request is received and before a channel establishment confirmation (L2CAP_ConnectRsp with result

code of 0x0000 or a higher-level channel establishment response such as that of RFCOMM) is sent.

See Section 5.2.2.6 for details on determining whether or not a link key is sufficient.

If pairing does not succeed, then the local device shall not send a channel establishment confirmation. The local device may retry pairing up to three (3) times. If pairing fails three consecutive times, then the local device shall disconnect the ACL link with error code 0x05 - Authentication Failure.

If the link-key generated is not at least as good as the expected or required type, then the local device shall fail the channel establishment and may disconnect the ACL link with error code 0x05 - Authentication Failure.

5.2.2.2.2 Authentication Required for Access to Local Service by Remote Device

See Section 5.2.2.6 for details on determining whether or not a link key is sufficient.

If authentication does not succeed, then the local device shall not send a channel establishment confirmation. The host may at this point notify the user and offer to perform pairing.

A Bluetooth device in security mode 4 shall respond to authentication and pairing requests during link establishment when the remote device is in security mode 3 for backwards compatibility reasons. However, authentication of the remote device shall be performed after the receipt of the channel establishment request is received, and before the channel establishment response is sent.

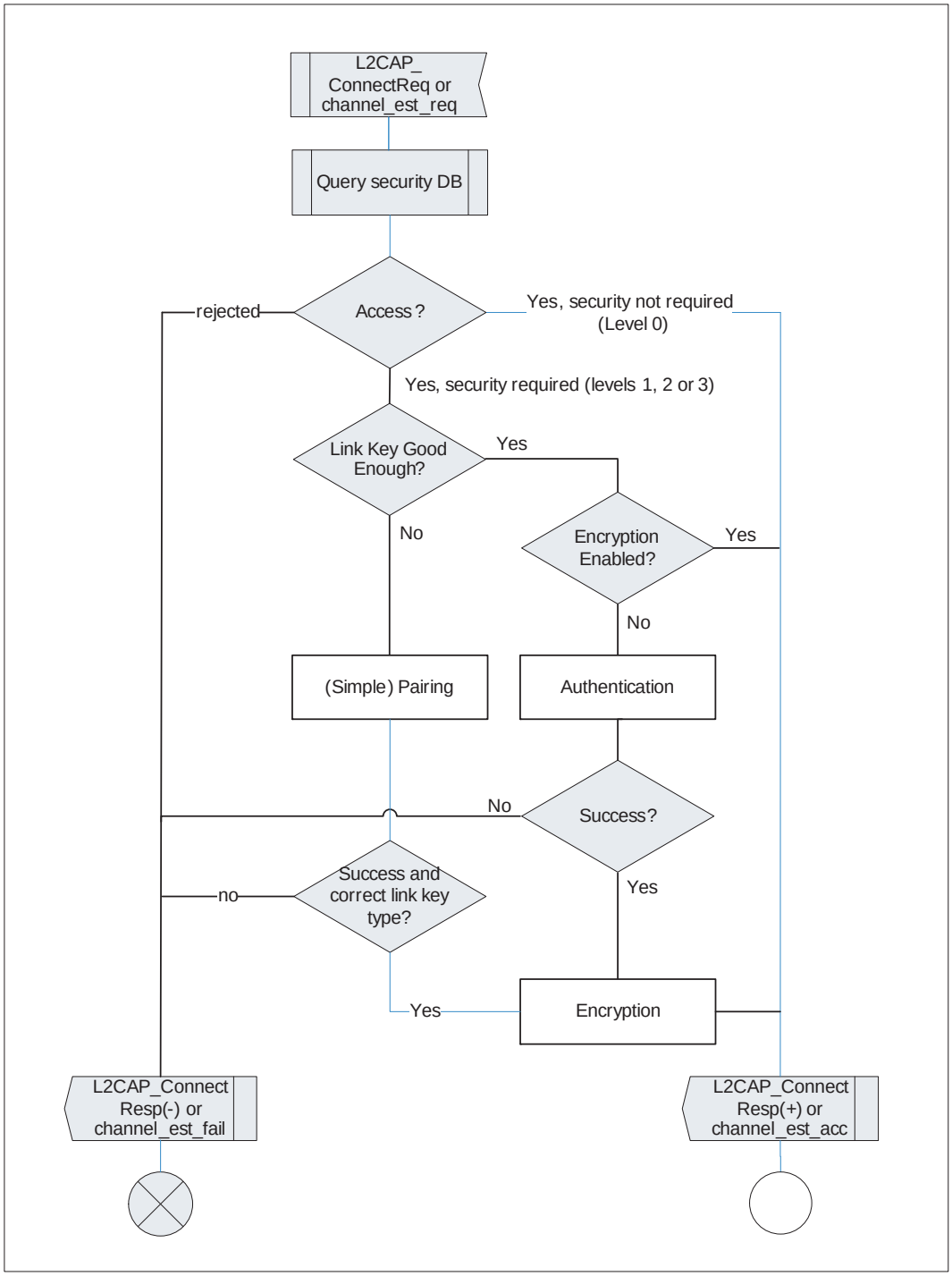


Figure 5.5: Channel establishment using security mode 4 for the responding side

5.2.2.3 Simple Pairing after Authentication Failure

When both devices support Secure Simple Pairing all non-SDP connections are encrypted regardless of whether security was required or whether the devices are bonded or not. The initial connection between the two devices will result in a link key through Secure Simple Pairing. Depending on whether or not bonding was performed and the security policy of the initiating device, the

link key may or may not be stored. When the link key is stored, subsequent connections to the same device will use authentication but this may fail if the remote device has deleted the link key. Table 5.1 defines what shall be done depending on the type of the link key and whether bonding was performed or not.

Link Key Type	Devices Bonded?	Action to take when Authentication Fails
Combination	No	Depends on security policy of the device: • Option 1: Automatically initiate pairing • Option 2: Notify user and ask if pairing is ok Option 2 is recommended.
Combination	Yes	Notify user of security failure
Unauthenticated	No	Depends on security policy of the device: • Option 1: Automatically initiate secure simple pairing • Option 2: Notify user and ask if secure simple pairing is ok. Option 1 is recommended.
Unauthenticated	Yes	Notify user of security failure
Authenticated	No	Depends on security policy of the device: • Option 1: Automatically initiate secure simple pairing • Option 2: Notify user and ask if secure simple pairing is ok Option 2 is recommended.
Authenticated	Yes	Notify user of security failure

Table 5.1: Simple Pairing after Authentication Failure

Note that non-bonded authenticated or unauthenticated link keys may be considered disposable by either device and may be deleted at any time.

5.2.2.4 IO Capabilities

Once a connection is established, if the Host determines that security is necessary and both devices support Secure Simple Pairing, the devices perform an IO capability exchange. The purpose of the IO capability exchange is to determine the authentication algorithm used in the Authentication Stage 1 phase of Simple Pairing.

The input and output capabilities are described in Table 5.2:

Capability	Description
No input	Device does not have the ability to indicate 'yes' or 'no'

Table 5.2: User Input Capabilities

Capability	Description
Yes / No	Device has at least two buttons that are mapped easily to 'yes' and 'no' or the device has a mechanism whereby the user can indicate either 'yes' or 'no' (see note below).
Keyboard	Device has a numeric keyboard that can input the numbers '0' through '9' and a confirmation. Device also has two buttons that can be easily mapped to 'yes' and 'no' or the device has a mechanism whereby the user can indicate either 'yes' or 'no' (see **Note** below).

Table 5.2: User Input Capabilities

Note: 'yes' could be indicated by pressing a button within a certain time limit otherwise 'no' would be assumed.

Capability	Description
No output	Device does not have the ability to display or communicate a 6 digit decimal number
Numeric output	Device has the ability to display or communicate a 6 digit decimal number

Table 5.3: User Output Capabilities

5.2.2.5 Mapping of Input / Output Capabilities to IO Capability

The individual input and output capabilities are mapped to a single IO capability which is later used to determine which authentication algorithm will be used.

Local Output Capacity / **Local Input Capacity**	**No Output**	**Numeric Output**
No input	NoInputNoOutput	DisplayOnly
Yes / No	NoInputNoOutput	DisplayYesNo
Keyboard	KeyboardOnly	DisplayYesNo

Table 5.4: IO Capability mapping

When a device has OOB authentication information from the remote device, it will indicate it in the LMP_IO_Capability_Res PDU. When either device has OOB information, the OOB authentication method will be used.

The Host may allow the Link Manager to ignore the IO capabilities and use the Numeric Comparison protocol with automatic accept by setting the Authentication_Requirements parameter to one of the MITM Protection *Not Required* options.

5.2.2.6 IO and OOB Capability Mapping to Authentication Stage 1 Method

Determining which association model to use in Authentication Stage 1 is performed in three steps. First, the devices look at the OOB Authentication Data Present parameter received in the remote IO capabilities. If either device has received OOB authentication data then the OOB association model is used. The event of receiving the OOB information is indicated by a device to its peer in the IO Capability Exchange step of simple pairing.

Device A / Device B	**Has not received remote OOB authentication data**	**Has received remote OOB authentication table**
Has not received remote OOB authentication data	Use the IO capability mapping table	Use OOB association with ra = 0 rb from OOB
Has received remote OOB authentication data	Use OOB association with ra from OOB rb = 0	Use OOB association with ra from OOB rb from OOB

Table 5.5: IO and OOB capability mapping

Second, if neither device has received OOB authentication data and if both devices have set the Authentication_Requirements parameter to one of the MITM Protection Not Required options, authentication stage 1 shall function as if both devices set their IO capabilities to DisplayOnly (e.g., Numeric comparison with automatic confirmation on both devices).

Finally, if neither device has received OOB authentication data and if one or both devices have set the Authentication_Requirements parameter to one of the *MITM Protection Required* options, the IO and OOB capabilities are mapped to the authentication stage 1 method as defined in the following table. A Host that has set the Authentication_Requirements parameter to one of the *MITM Protection Required* options shall verify that the resulting Link Key is an Authenticated Combination Key (see Section 7.7.24 in Host Controller Interface Functional Specification). The table also lists whether the combination key results in an authenticated or unauthenticated link key.

Note: The "DisplayOnly" IO capability only provides unidirectional authentication.

5.2.2.7 Out of Band (OOB)

An out of band mechanism may also be used to communicate discovery information as well as other information related to the pairing process.

The contents of the OOB data block are:

Mandatory contents:

Initiator A / B Responder	Display Only	DisplayYesNo	KeyboardOnly	NoInputNoOutput
DisplayOnly	Numeric Comparison with automatic confirmation on both devices. Unauthenticated	Numeric Comparison with automatic confirmation on device B only. Unauthenticated	Passkey Entry: Responder Display, Initiator Input. Authenticated	Numeric Comparison with automatic confirmation on both devices. Unauthenticated
DisplayYesNo	Numeric Comparison with automatic confirmation on device A only. Unauthenticated	Numeric Comparison: Both Display, Both Confirm. Authenticated	Passkey Entry: Responder Display, Initiator Input. Authenticated	Numeric Comparison with automatic confirmation on device A only. Unauthenticated
Keyboard Only	Passkey Entry: Initiator Display, Responder Input. Authenticated	Passkey Entry: Initiator Display, Responder Input. Authenticated	Passkey Entry: Initiator and Responder Input. Authenticated	Numeric Comparison with automatic confirmation on both devices. Unauthenticated
NoInputNoOutput	Numeric Comparison with automatic confirmation on both devices. Unauthenticated	Numeric Comparison with automatic confirmation on device B only. Unauthenticated	Numeric Comparison with automatic confirmation on both devices. Unauthenticated	Numeric Comparison with automatic confirmation on both devices. Unauthenticated

Table 5.6: IO Capability Mapping to Authentication Stage 1

- Length (2 bytes)
- BD_ADDR (6 bytes)

Optional contents:

- Class of Device (3 bytes)
- Simple Pairing Hash C (16 bytes)
- Simple Pairing Randomizer R (16 bytes)
- Local name (variable length)
- Other information

The length field includes all bytes in the OOB data block including the length field itself. The BD_ADDR will be a fixed field in the beginning of the OOB data block. Following the BD_ADDR will be zero or more EIR tag fields containing optional contents. The EIR tag format will be used for the optional contents.

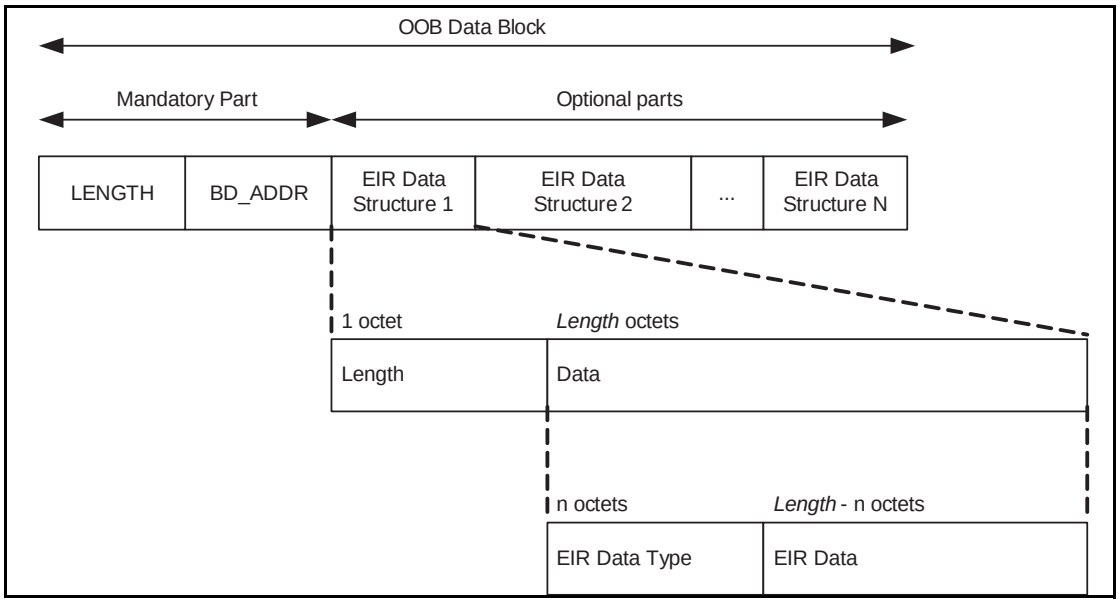


Figure 5.6: OOB Data Block Format

If Simple Pairing fails when one or both devices have OOB Authentication Data present, both devices shall discard the OOB Authentication Data and the device that originally initiated authentication shall re-initiate authentication. Note that although the user may get involved in authentication as defined by the IO capabilities of the two devices, falling back to the in-band authentication method will prevent deadlock conditions when one or both devices has stale OOB Authentication Data.

5.2.2.8 Security Database

A Bluetooth device in security mode 4 shall classify the security requirements of its services using at least the following levels attributes (in order of decreasing security) for use when pairing with remote devices supporting Secure Simple Pairing:

- Level 3, for services with the following attributes:
  MITM protection required
  Encryption required
  User interaction acceptable
- Level 2, for services with the following attributes:
  MITM protection not required
  Encryption required
- Level 1, for services with the following attributes:

MITM protection not required

Minimal user interaction desired

One additional level is permitted for L2CAP fixed signaling channels (CIDs 0x0001, 0x0003 and 0x003F),SDP and for some service traffic sent via the connectionless L2CAP channel for services that do not require security but shall not be used for any other service traffic:

- Level 0: Service requires the following:

  MITM protection not required

  No encryption required

  No user interaction required

  Level 0 may only be used for Services with the following Service Class UUIDs:

  0x1000 + BASE_ID (Service Discovery Server)

The security level required for each service offered should be stored in a security database that is accessed to determine the type of link key that is required for access to the respective service. The security level required for service data transmitted on an L2CAP connection-oriented channel may differ from the security level required for service data transmitted on another L2CAP connection-oriented channel or on the connectionless L2CAP channel. Table 5.7 on page 311 shows the type of link key required for each security level for both remote devices that support Secure Simple Pairing (v2.1 + EDR remote devices) and for those that do not (pre-v2.1 + EDR remote devices).

Security Level Required for Service	**Link Key type required for remote devices**	**Link Key type required for pre-v2.1 remote devices**	**Comments**
Level 3 • Strong MITM protection desired • Encryption desired • User interaction acceptable	Authenticated	Combination (16 digit PIN recommended)	High Security
Level 2 • MITM protection not necessary • Encryption desired	Unauthenticated	Combination	Medium Security

Table 5.7: Security Level mapping to link key requirements

Security Level Required for Service	Link Key type required for remote devices	Link Key type required for pre-v2.1 remote devices	Comments
Level 1 • MITM protection not necessary • Encryption not necessary[1] • Minimal user interaction desired	Unauthenticated	None	Low Security
Level 0 • MITM protection not necessary • Encryption not necessary • No user interaction desired	None	None	Permitted only for SDP and service data sent via either L2CAP fixed signaling channels or the L2CAP connectionless channel to PSMs that correspond to service class UUIDs which are allowed to utilize Level 0.

Table 5.7: Security Level mapping to link key requirements

1. Though encryption is not necessary for the service for Level 1, this specification mandates the use of encryption when the remote device is v2.1+EDR for all services other than SDP.

An *authenticated* link key is a link key where either the numeric comparison, out-of-band, or passkey entry simple pairing association models were used. An authenticated link key has protection against MITM attacks. To ensure that an authenticated link key is created during the Simple Pairing procedure, the Authentication_Requirements parameter should be set to one of the *MITM Protection Required* options.

An *unauthenticated* link key is a link key where the “Just Works” simple pairing association model was used (see [Vol. 1, Part A] Section 5.1.4 on page 87). An unauthenticated link key does not have protection against MITM attacks. To allow an unauthenticated link key to be created during the Simple Pairing procedure, the Authentication_Requirements parameter may be set to one of the *MITM Protection Not Required* options.

A combination link key is a link key where the v2.0 pairing mechanism was used to generate the link-key (see [Vol. 2, Part C] Section 4.2.2.4 on page 251).

When both devices support Secure Simple Pairing, GAP shall require at least an unauthenticated link key and enable encryption for service traffic sent or received via connection-oriented L2CAP channels. A profile or protocol may define services that require more security (for example, an authenticated link

key) or no security in the case of SDP or service traffic sent via the L2CAP connectionless channel for services that do not require security.

When the device is in Bondable Mode, it shall enable Secure Simple Pairing mode prior to entering Connectable Mode or establishing a link.

A Bluetooth device in security mode 4 shall respond to authentication and pairing requests during link establishment when the remote device is in security mode 3 for backwards compatibility reasons. See Section 5.2.1.3 for more information.

The remote Controller's and remote Host's support for Secure Simple Pairing shall be determined by the Link Manager Secure Simple Pairing (Host Support) feature bit.

A previously generated link key is considered “sufficient” if the link key type is of the type required for the service, or of a higher strength. Authenticated link keys are considered higher strength than Unauthenticated or Combination keys. Unauthenticated link keys are considered higher strength than Combination keys.

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 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 simple 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, page scan mode, and extended inquiry response information 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

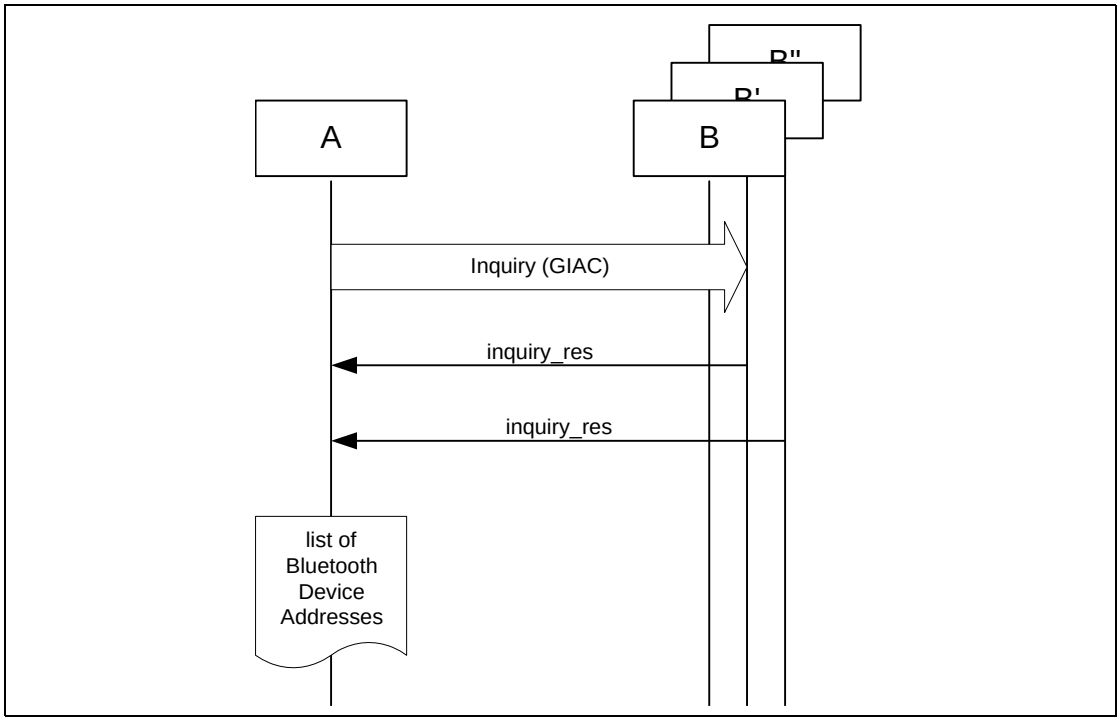


Figure 6.1: General inquiry, where B is a device in non-discoverable mode, B´ is a device in limited discoverable mode and B" is a device in general discoverable mode. (Note that all discoverable devices are discovered using general inquiry, independent of which discoverable mode they are in.)

6.1.4 Conditions

When general inquiry is initiated by a Bluetooth device, the INQUIRY state shall last $T_{GAP}(100)$ or longer, unless the inquirer collects enough responses and determines to abort the INQUIRY state earlier. The Bluetooth device shall perform inquiry using the GIAC.

In order for Device A to receive inquiry responses, 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, page scan mode, and extended inquiry response information 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

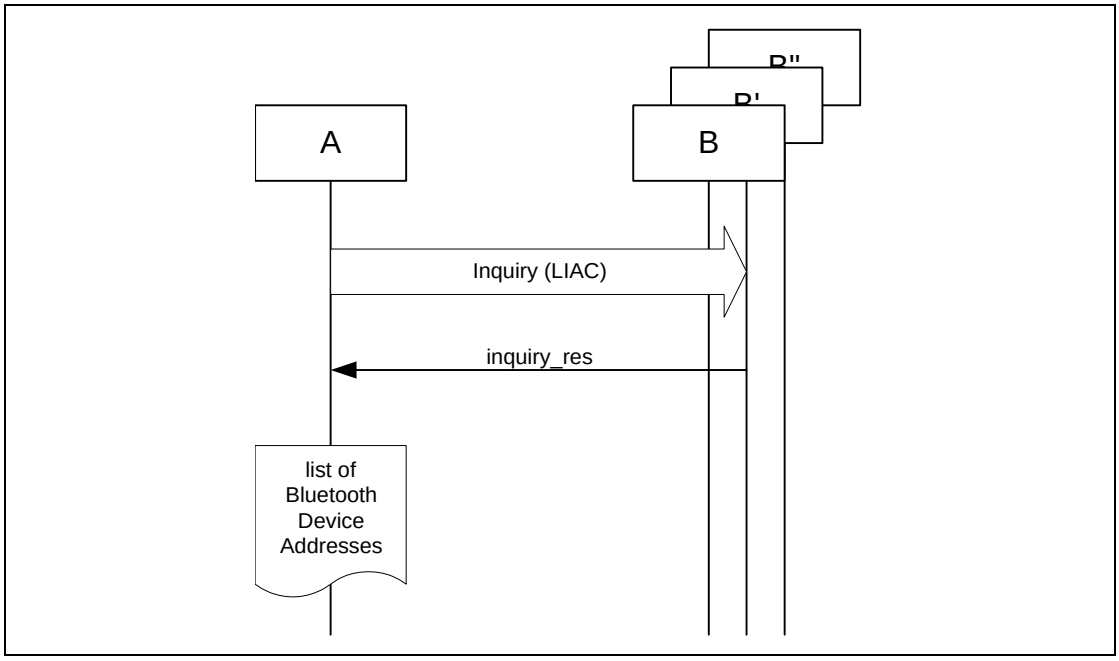


Figure 6.2: Limited inquiry where B is a device in non-discoverable mode, B' is a device in limited discoverable mode and B" is a device in general discoverable mode. (Note that only limited discoverable devices can be discovered using limited inquiry.)

6.2.4 Conditions

When limited inquiry is initiated by a Bluetooth device, the INQUIRY state shall last $T_{GAP}(100)$ or longer, unless the inquirer collects enough responses and determines to abort the INQUIRY state earlier. The Bluetooth device shall perform inquiry using the LIAC.

In order for Device A to receive inquiry responses, 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 7.1) in order to just to get the name of another device.

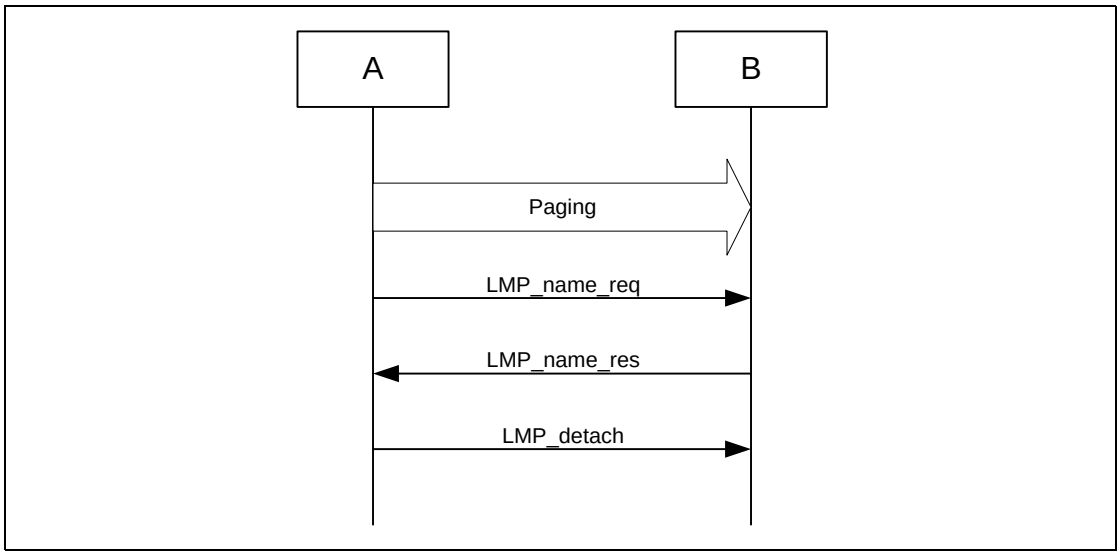


Figure 6.3: Name request procedure.

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).

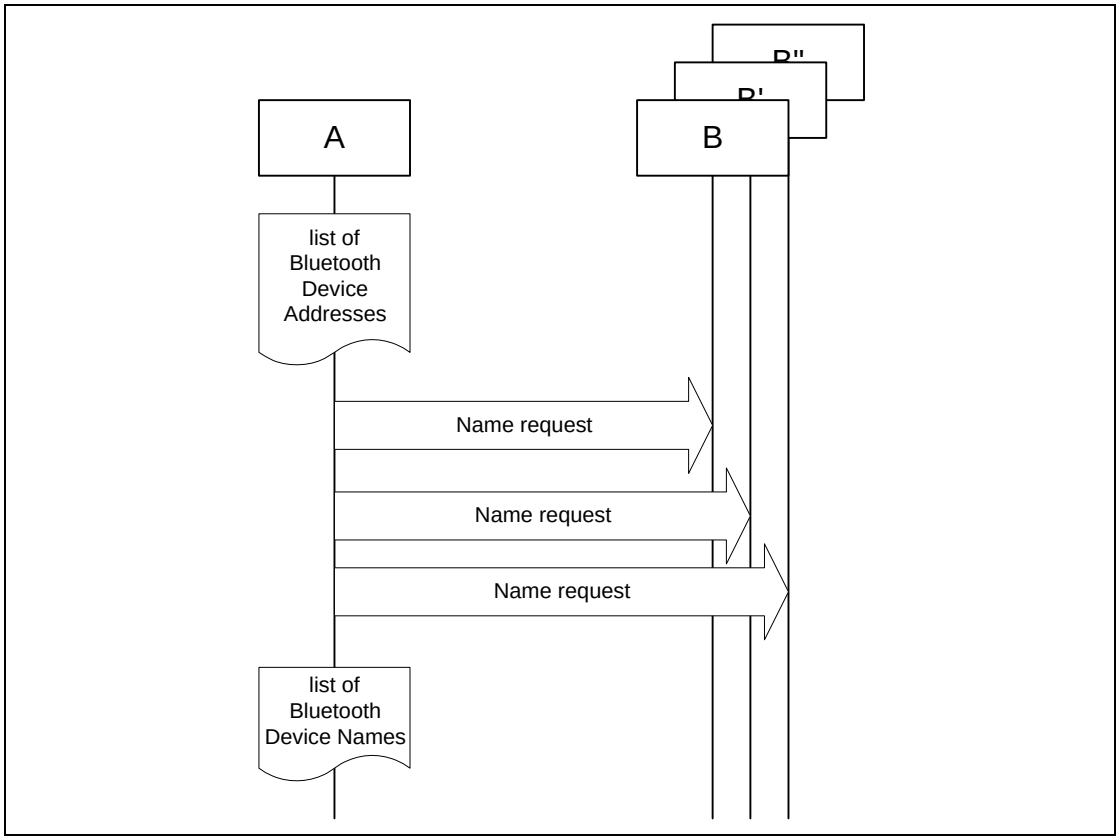


Figure 6.4: Name discovery procedure.

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

This section only applies to a device of the BR/EDR device type. Devices of the BR/EDR/LE device shall follow the device discovery defined in procedure in Section 13.2.3.

6.4.1 Purpose

The purpose of device discovery is to provide the initiator with the Bluetooth Device Address, clock, Class of Device, used page scan mode, Bluetooth Device Name, and extended inquiry response information 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. If the initiator of the device discovery receives a complete local name or a shortened local name that is considered long enough, via an extended inquiry response from a remote device, the initiator should not do a separate name discovery for that device.

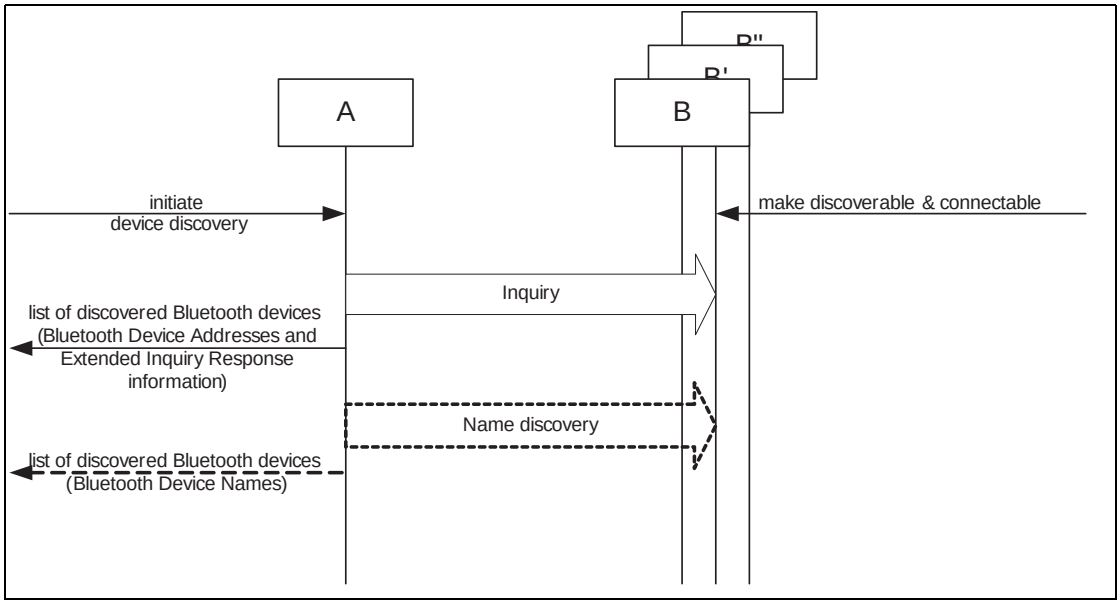


Figure 6.5: Device discovery procedure.

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 forms of bonding are described in the following sections: General Bonding and Dedicated Bonding.

6.5.3.1 General Bonding

General Bonding refers to the process of performing bonding during connection setup or channel establishment procedures as a precursor to accessing a service.

When the devices that are performing General Bonding both support Secure Simple Pairing, the Authentication_Requirements parameter should be set to MITM Protection Not Required – General Bonding unless the security policy of an available local service requires MITM Protection in which case the Authentication_Requirements parameter shall be set to MITM Protection Required – General Bonding. 'No bonding' is used when the device is performing a Secure Simple Pairing procedure, but does not intend to retain the link key after the physical link is disconnected.

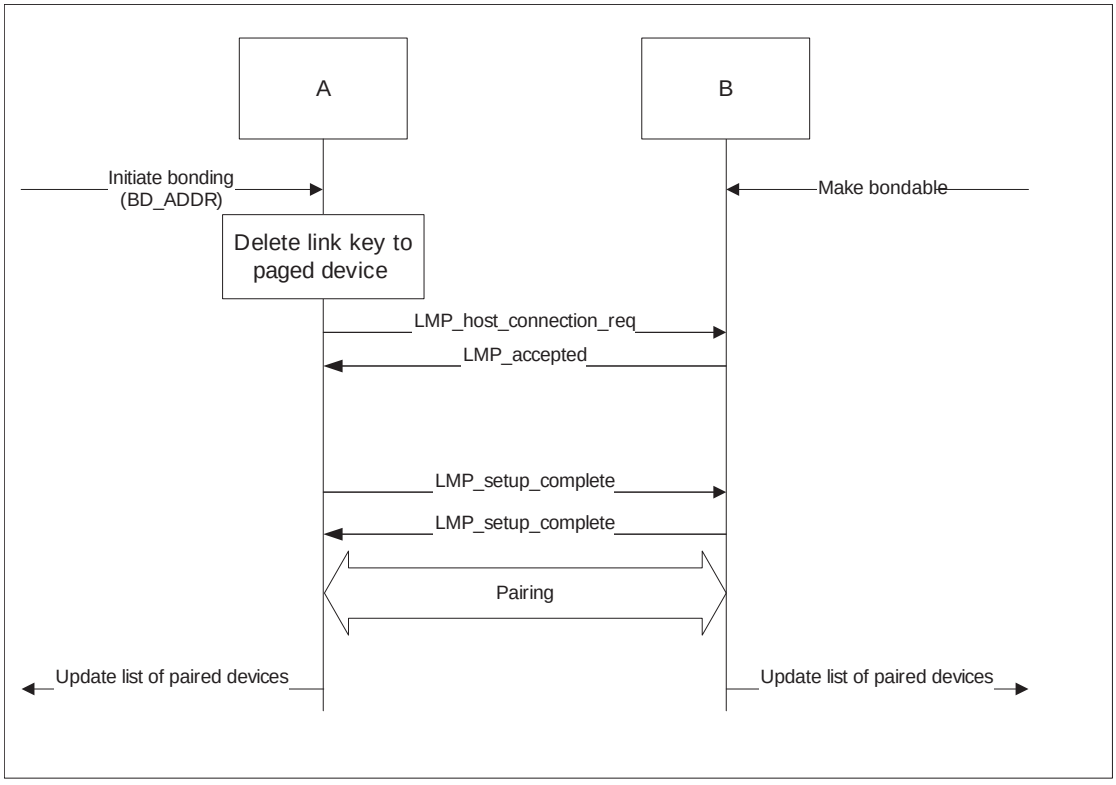


Figure 6.6: General description of general bonding as being the link establishment procedure executed under specific conditions on both devices, followed by authentication and an optional higher layer initalization process.

6.5.3.2 Dedicated Bonding

Dedicated Bonding refers to a procedure wherein one device connects to another only for the purpose of pairing without accessing a particular service.

The main difference with dedicated 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.

When the devices that are performing Dedicated Bonding both support Secure Simple Pairing, the Authentication_Requirements parameter should be set to *MITM Protection Not Required – Dedicated Bonding* unless the security policy of an available local service requires MITM Protection in which case the Authentication Required parameter shall be set to *MITM Protection Required – Dedicated Bonding*. 'No bonding' is used when the device is performing a Secure Simple Pairing procedure, but does not intend to retain the link key after the physical link is disconnected.

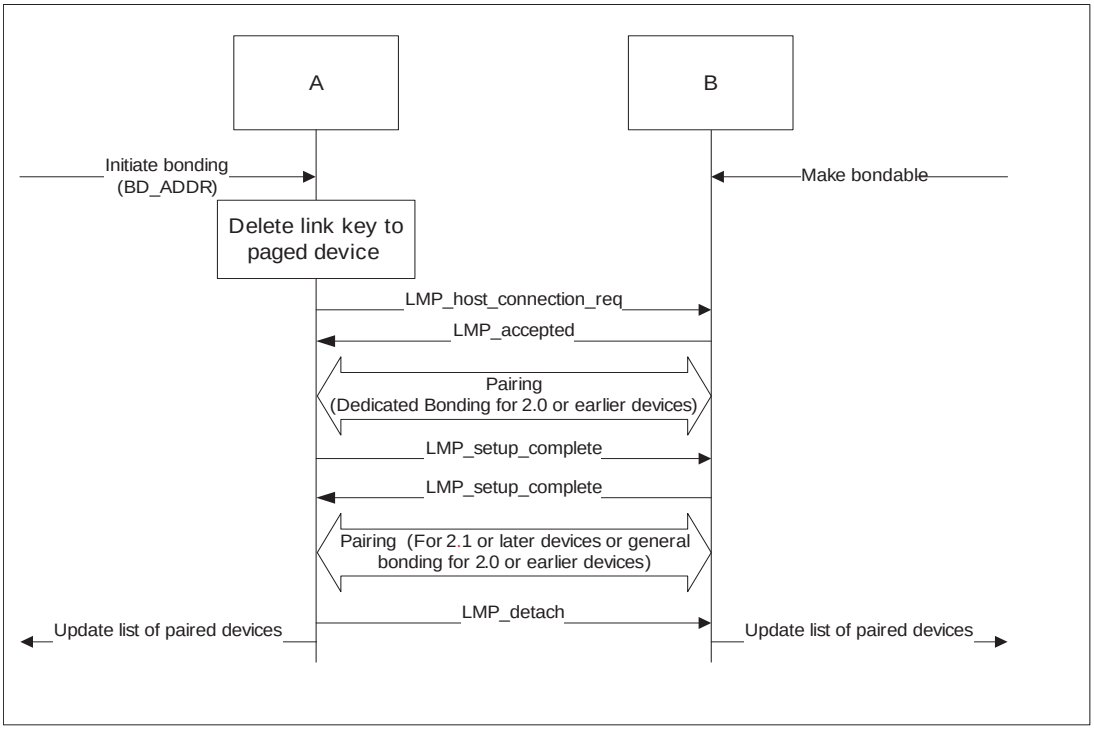


Figure 6.7: Dedicated Bonding as performed when the purpose of the procedure is only to create and exchange a link key between two Bluetooth devices.

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 bondable 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 or the extended inquiry response 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;
- The supported Service Classes.

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. During link establishment, the paging device cannot distinguish if the paged device (B) is in security mode 1, 2 or 4.

7.1.3.1 B in security mode 1, 2, or 4

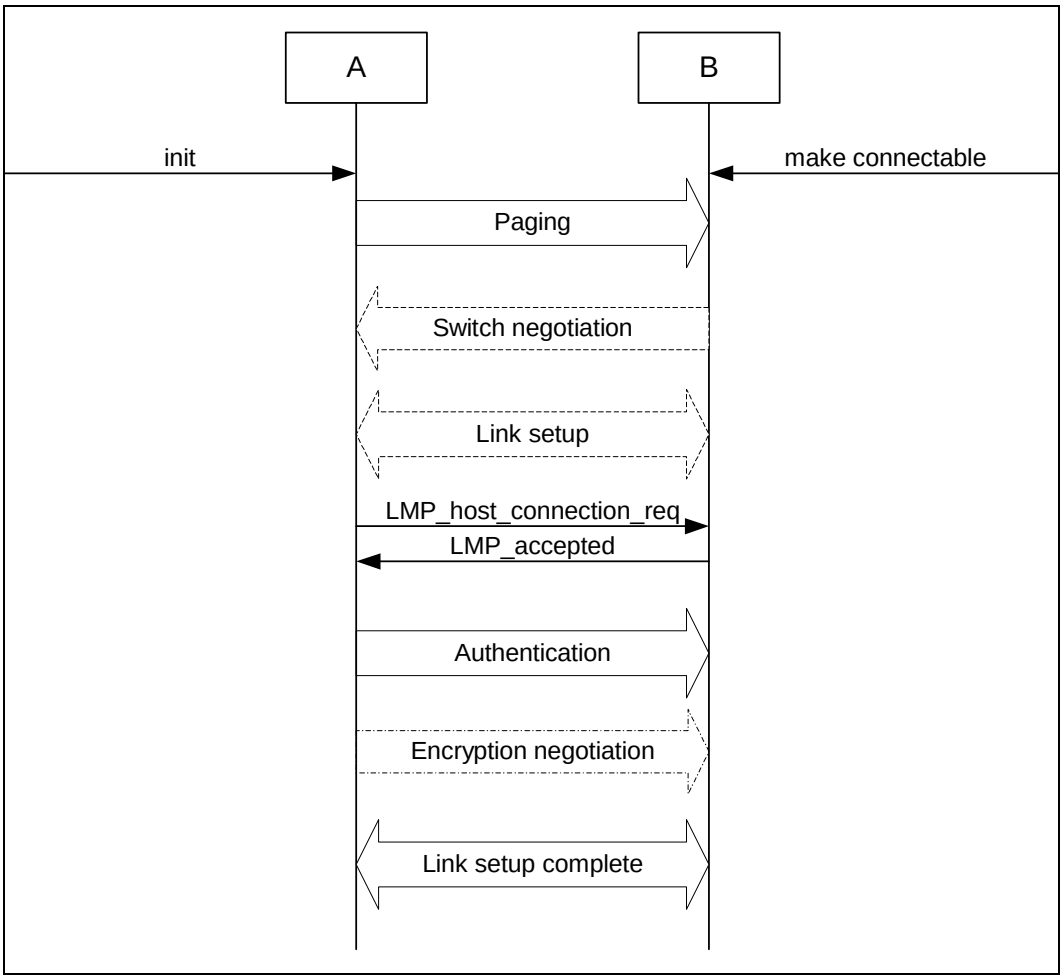


Figure 7.1: Link establishment procedure when the paging device (A) is in security mode 3 and the paged device (B) is in security mode 1, 2, or 4.

7.1.3.2 B in security mode 3

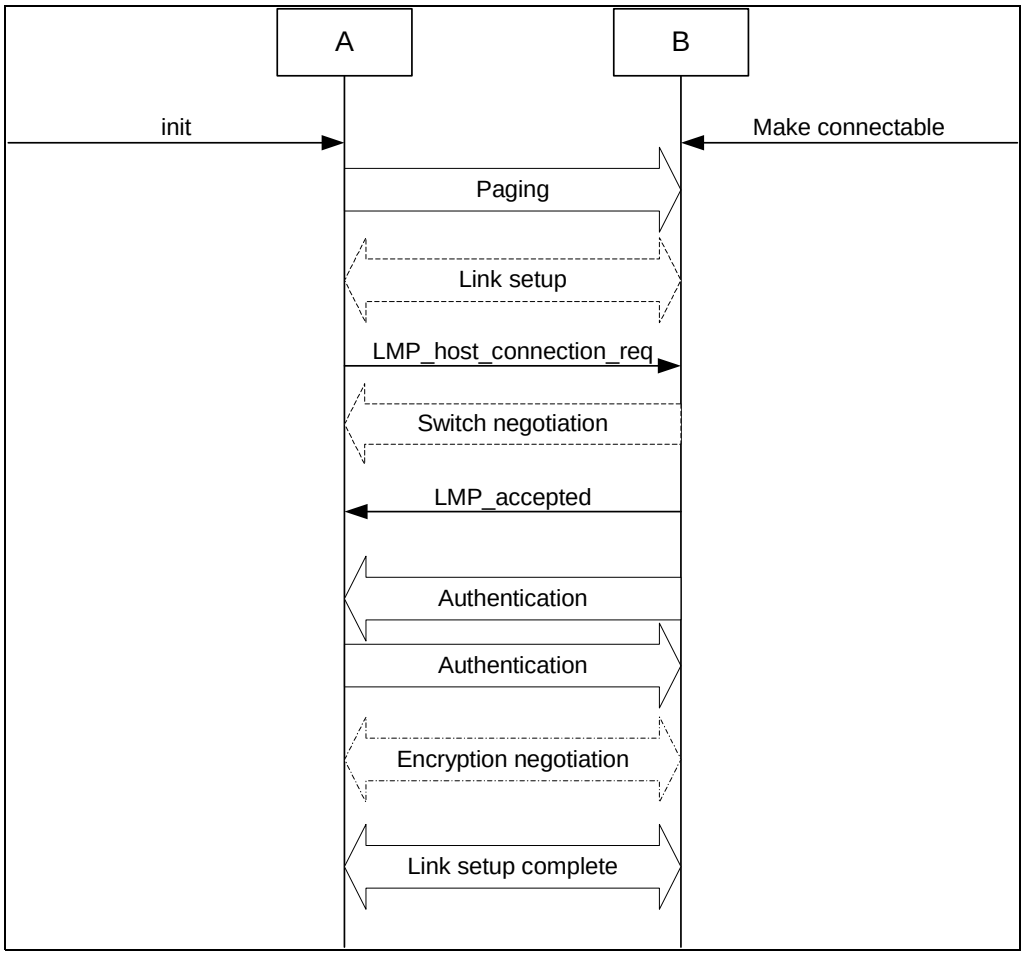


Figure 7.2: Link establishment procedure when both the paging device (A) and the paged device (B) are in security mode 3.

7.1.4 Conditions

The paging procedure shall be according to [Vol. 2], Part B Section 8.3 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_req.

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 (L2CAP channel) between two Bluetooth devices using [1].

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 or 4

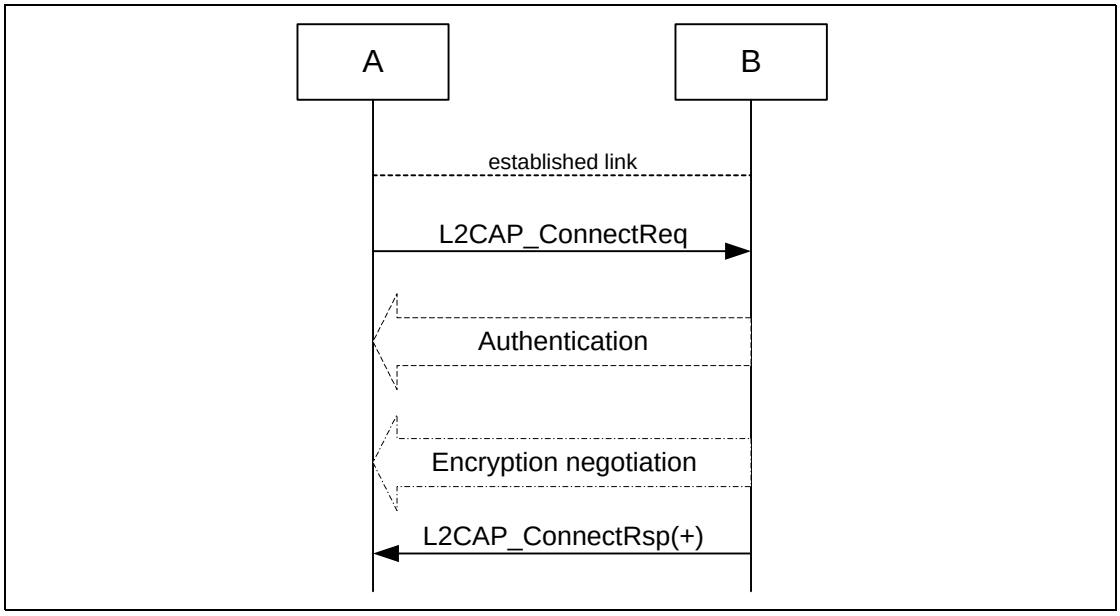


Figure 7.3: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2 or 4.

7.2.3.2 B in security mode 1 or 3

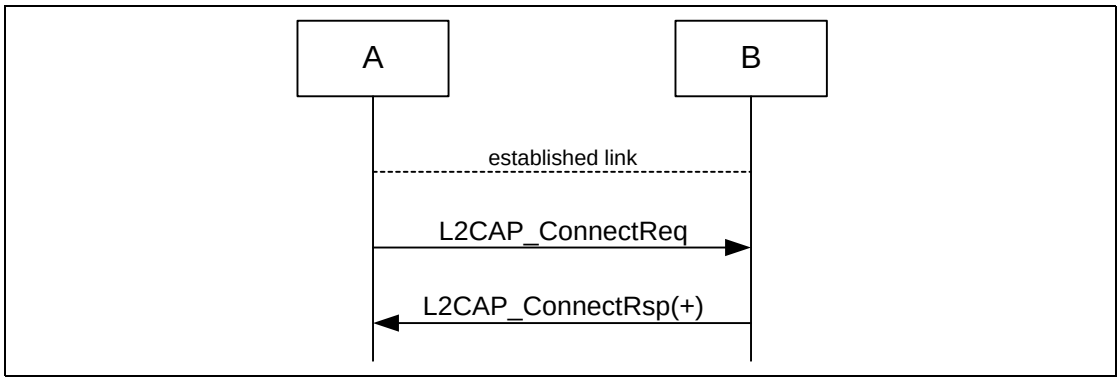


Figure 7.4: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

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 or 4

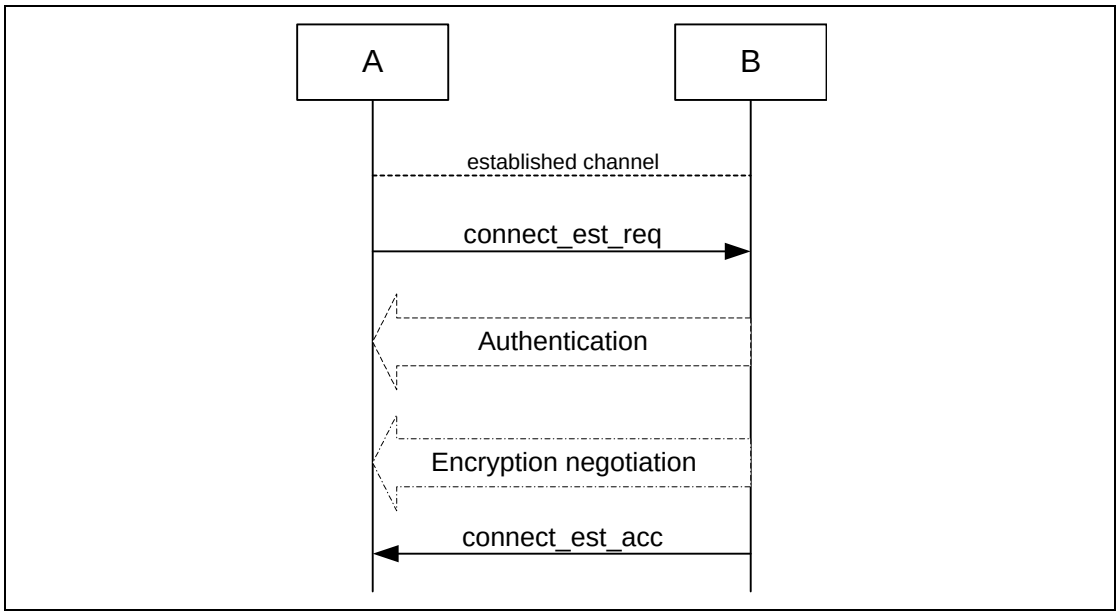


Figure 7.5: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2 or 4

7.3.3.2 B in security mode 1 or 3

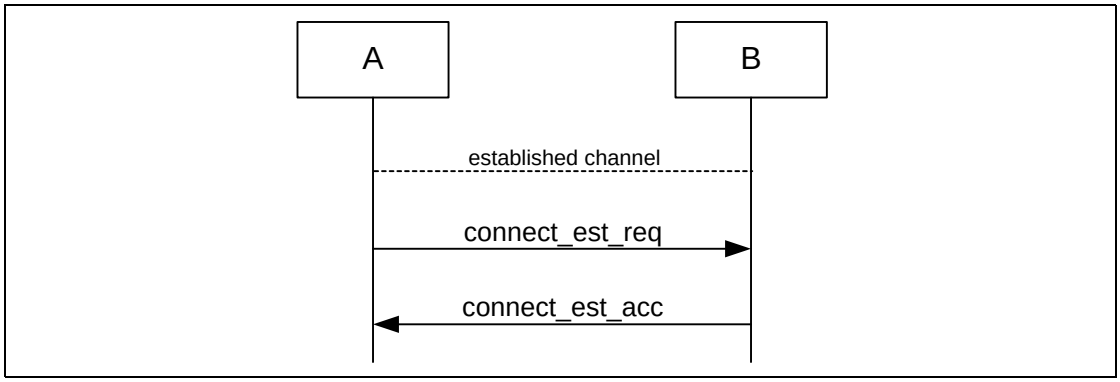


Figure 7.6: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

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 [4] in the case of a Bluetooth telephony application Cordless Telephony Profile, or initialization of RFCOMM and establishment of DLC [3] 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 logical 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 EXTENDED INQUIRY RESPONSE DATA FORMAT

The extended inquiry response data format is shown in Figure 8.1. The data is 240 octets and consists of a significant part and a non-significant part. The significant part contains a sequence of data structures. Each data structure shall have a length field of one octet, which contains the Length value, and a data field of Length octets. The first n octets of the data field contain the extended inquiry response (EIR) data type. The content of the remaining Length - n octets in the data field depends on the value of the EIR data type and is called the EIR data. The non-significant part extends the extended inquiry response to 240 octets and shall contain all-zero octets.

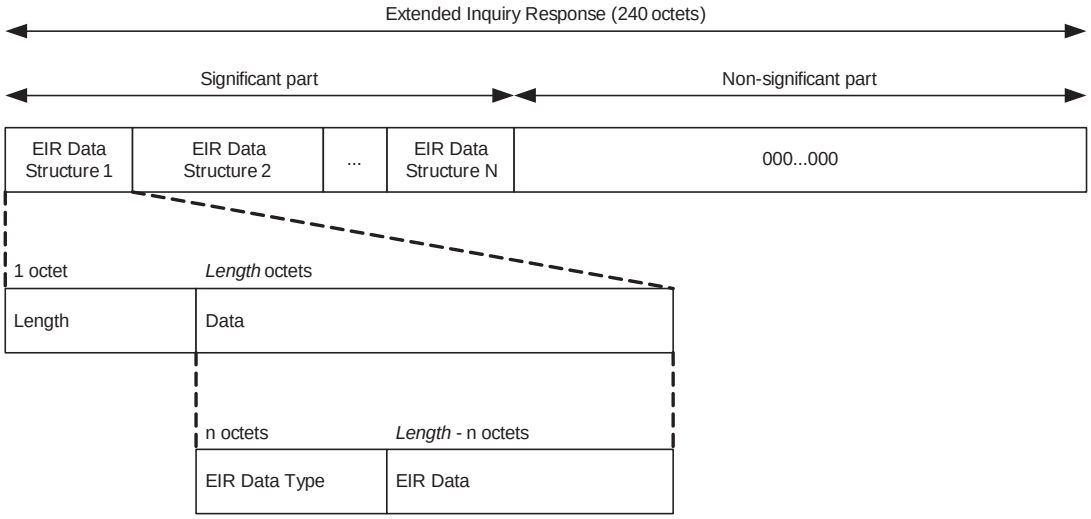


Figure 8.1: Extended Inquiry Response data format

The extended inquiry response data types are specified in the Assigned Numbers document.

If the length field is set to zero, then the data field has zero octets. This shall only occur to allow an early termination of the tagged data.

To reduce interference, the host should try to minimize the amount of EIR data such that the baseband can use a 1-slot or 3-slot EIR packet. This is advantageous because it reduces interference and maximizes the probability that the EIR packet will be received. If applications on a host provide more than 240 bytes of extended inquiry response data, it is up to the host to limit it to 240 octets.

The EIR data shall always be sent during inquiry response state. EIR data can contain device name, Tx power level, service class UUIDs, as well as manufacturers data, as defined in Section 8.1. In selecting the packet type to be used, FEC (DM1 or DM3) should be considered to maximize the range.

The Host shall include the device name in the EIR data according to the following rules:

1. If the device does not have a device name (i.e., 0 octet) and
    a) If there is no other data to be sent in the EIR packet, then host shall send a name tag with zero length and the type field set to indicate that this is the complete name (i.e., total of 2 octets with length = 1).
    b) If there is other important data to be sent in the EIR packet and a zero octet name tag will not fit, then host may avoid sending the name tag.
2. If the device has a device name (greater than 0 octet) and
    a) If it is too long be included in the EIR packet (given the choice of packet type and any other data that is being sent), then host may send a shortened version of the name (even 0 octet) and shall mark the name as 'shortened' to inform the receiver that a remote name request is required obtain the full name if the name is needed.
    b) If there are no other data to be sent in the EIR packet (given the choice of packet type selected), then host shall maximize the length of the device name to be sent, this may be complete or shortened name (e.g., if DM1 packet is chosen and device name characters equates to greater than 15 octets, then host sends first few characters that equates to 15 octets or less with shortened flag).

Note: It is not necessary to understand each and every EIR data type. If a host does not understand a given EIR data type value it should just skip over Length octets and look for the next EIR data structure.

8.1 EIR DATA TYPE DEFINITIONS

This section defines the basic EIR data types. Additional EIR data types may be defined in profile specifications.

All EIR data type values are listed in the Bluetooth Assigned Numbers document.

All numerical multi-byte entities and values associated with the following data types shall use little-endian byte order.

8.1.1 Service Class UUIDs

A list of service class UUIDs (see Section 2.4, “Service Class,” on page 215) may be included in the extended inquiry response data packet. There are three types of UUIDs that may be returned: 16-bit Bluetooth UUIDs, 32-bit Bluetooth UUIDs, and global 128-bit UUIDs. Each size is assigned two data type values.

In addition to the size, the data type values also indicate if the list of UUIDs is complete or there are more UUIDs available via SDP. An extended inquiry response shall not contain more than one list for each size of UUIDs. If a device has no service class UUIDs of a certain size the corresponding field in the extended inquiry response shall be marked as complete with no service class UUIDs. An omitted service class UUID list shall be interpreted as an empty incomplete list.

The format of the data is defined in Section 18.2.

8.1.2 Local Name

When included in the extended inquiry response, this shall be the same as, or a shortened version of, the local name assigned to the device. The data type value indicates if the name is complete or shortened. If the name is shortened, the complete name can be read using the remote name request procedure. An extended inquiry response shall not contain more than one instance of this data type.

The format of the data is defined in Section 18.3.

8.1.3 Flags

This data type is allocated for flag bits. These may for example be used for boolean information that would otherwise require a whole byte (not counting the Length and Data Type fields). The flag bits shall be included in the extended inquiry response if any of the bits are non-zero. If all flag bits are zero they may be omitted. An extended inquiry response shall not contain more than one instance of this data type.

The format of the data is defined in Section 18.1.

The LE Limited Discoverable Mode and LE General Discoverable Mode flags shall be ignored when received over the BR/EDR physical channel. The ‘BR/EDR Not Supported’ flag shall not be sent over the BR/EDR physical channel.

8.1.4 Manufacturer Specific Data

This data type is used for manufacturer specific data. The first two data octets shall contain a company identifier code from the Assigned Numbers - Company Identifiers document. The interpretation of any other octets within the data shall be defined by the manufacturer specified by the company identifier.The format of the data is defined in Section 18.11.

Value	Description
0xFF	Manufacturer Specific Data

8.1.5 TX Power Level

The TX Power Level tag indicates the transmitted power level of the EIR packet.

The format of the data is defined in Section 18.4.

The TX Power Level tag may be used to calculate path loss on a received EIR packet using the following equation:

pathloss = Tx Power Level – EIR_RSSI

For example, if Tx Power Level = +4 (dBm) and the RSSI on the EIR packet is -60 (dBm) then the total pathloss is +4 – (-60) = +64 dB. If a second EIR packet were received at -40dBm with a Tx Power Level tag = +15dBm the resulting pathloss would be +55dB. An application could use these pathloss values to choose which device it thinks might be closer (the one with the lower pathloss value).

Unfortunately, due to fading and varying antenna, circuit, and chip characteristics, these resulting pathloss values will have uncertainty. Some of the uncertainty (for example, due to fading) may be able to be removed if multiple EIR packets are received from the same device.

8.1.6 Secure Simple Pairing Out of Band (OOB)

An out of band mechanism may be used in Secure Simple Pairing to communicate discovery information as well as other information related to the pairing process. These tags use the standard EIR tag format. They shall only be used over an out-of-band mechanism.

Value	Description
NA	Length (2 bytes) 0xXXXX: 8 to 65535 bytes This field contains the length of the entire OOB data block including the length field itself.
NA	BD_ADDR (6 bytes) Format defined in [Vol. 2, Part B] Section 1.2 on page 68

Table 8.1: Mandatory OOB Tags

The format of the Optional OOB Tags is defined in Section 18.5.

8.2 EXAMPLE EXTENDED INQUIRY RESPONSE

This is an example extended inquiry response for a phone with PANU, Hands-free Audio Gateway, and Intercom Service:

Value	Notes

0x06	Length of this Data
0x09	Complete Local Name
0x50	'P'
0x68	'h'
0x6F	'o'
0x6E	'n'
0x65	'e'
0x07	Length of this Data
0x03	Complete list of 16-bit Service Class UUIDs
0x15	PANU service class UUID
0x11	
0x1F	Handsfree Audio Gateway service class UUID
0x11	
0x09	Cordless Telephony service class UUID
0x11	
0x01	Length of this data
0x05	Complete list of 32-bit Service Class UUIDs
0x01	Length of this data
0x07	Complete list of 128-bit Service Class UUIDs
0x00	End of Data (Not transmitted over the air)

9 OPERATIONAL MODES AND PROCEDURES FOR USE ON LE PHYSICAL CHANNELS

Several different modes and procedures may be performed simultaneously over an LE physical channel. The following modes and procedures are defined for use over LE physical channels:

- Broadcast mode and observation procedure
- Discovery modes and procedures
- Connection modes and procedures
- Bonding modes and procedures

Each of the above modes and procedures are independent from each other but are closely related since a combination of the modes and procedures are necessary for most devices to communicate with each other. Both the modes and procedures may be entered or executed respectively as a result of direct user action or autonomously by a device.

The Host shall configure the Controller with its local Link Layer feature information as defined in [Vol. 6], Part B Section 4.6 before performing any of the above modes and procedures.

9.1 BROADCAST MODE AND OBSERVATION PROCEDURE

The broadcast mode and observation procedure allow two devices to communicate in a unidirectional connectionless manner using the advertising events. The requirements for a device operating in a specific GAP role to support the broadcast mode and observation procedure are defined in Table 9.1.

Broadcast Mode and Observation procedure	Ref.	Broadcaster	Observer
Broadcast mode	9.1.1	M	E
Observation procedure	9.1.2	E	M

Table 9.1: Broadcast mode and Observation procedure Requirements

9.1.1 Broadcast Mode

9.1.1.1 Definition

The broadcast mode provides a method for a device to send connectionless data in advertising events.

9.1.1.2 Conditions

A device in the broadcast mode shall send data in either non-connectable undirected or scannable undirected advertising events.

The advertising data shall be formatted using the Advertising Data (AD) type format as defined in Section 11. A device in the broadcast mode shall not set the 'LE General Discoverable Mode' flag or the 'LE Limited Discoverable Mode' flag in the Flags AD Type as defined in Section 11.1.3.

When privacy is enabled, as defined in Section 10.7, and the device is operating in the broadcast mode, the Host shall generate a resolvable private address using the resolvable private address generation procedure as defined in Section 10.8.2.2. The Host shall configure the Controller to use the resolvable private address as the advertiser's device address in the advertising events. If privacy is disabled then the Host shall disable the T_{GAP}(private_addr_int) timer and configure the Controller to use the public/static address as the advertiser's device address in the advertising events. When the T_{GAP}(private_addr_int) timer expires the Host shall repeat the above procedure. The device should not send the device name in the advertising data. If the device receives a scan request, the device shall only respond if the scan request is from a known device.

Note: All data sent by a device in the broadcast mode is considered unreliable since there is no acknowledgement from any device that may have received the data.

Note: A device that only supports the broadcast mode cannot support sending encrypted or signed data because the device cannot support the distribution of keys as defined by the Security Manager (see [Vol. 3], Part H). Key distribution may be performed by a device using the Peripheral or Central profile role.

9.1.2 Observation Procedure

9.1.2.1 Definition

The observation procedure provides a method for a device to receive connectionless data from a device that is sending advertising events.

9.1.2.2 Conditions

A device performing the observation procedure may use passive scanning or active scanning to receive advertising events.

A device performing the observation procedure may use active scanning to also receive scan response data sent by any device in the broadcast mode that advertises using scannable undirected advertising events. When a device performing the observation procedure receives a resolvable private address in the

advertising event, the device may resolve the private address by using the resolvable private address resolution procedure as defined in Section 10.8.2.3.

A privacy enabled device as defined in Section 10.7 performing the observation procedure shall use a non-resolvable private address as the device address when performing active scanning. The generation of a non-resolvable address is defined in Section 10.8.2.1.

Note: In use cases where a device in the broadcast mode sends dynamic data, the receiving device should disable duplicate filtering capability in the Controller so that the Host receives all advertising packets received by the Controller.

Note: A device performing the observation procedure cannot support receiving of encrypted or signed data because the device cannot support the distribution of keys as defined by Security Manager in [Vol. 3], Part H Section 3.6.1. Key distribution may be performed by a device using the Peripheral or Central profile role.

9.2 DISCOVERY MODES AND PROCEDURES

All devices shall be in either non-discoverable mode or one of the discoverable modes. A device in the discoverable mode shall be in either the general discoverable mode or the limited discoverable mode. A device in the non-discoverable mode is not discoverable. Devices operating in either the general discoverable mode or the limited discoverable mode can be found by the discovering device. A device that is discovering other devices performs either the limited discovery procedure as defined in Section 9.2.5 or the general discovery procedure as defined in Section 9.2.6.

9.2.1 Requirements

Discovery modes and procedures	Ref.	Peripheral	Central
Non-Discoverable mode	9.2.2	M	E
Limited Discoverable mode	9.2.3	O	E
General Discoverable mode	9.2.4	C1	E
Limited Discovery procedure	9.2.5	E	O
General Discovery procedure	9.2.6	E	M
Name Discovery procedure	9.2.7	O	O
C1: if limited discoverable mode is not supported then general discoverable mode is mandatory, else optional.			

Table 9.2: Device discovery requirements

9.2.2 Non-Discoverable Mode

9.2.2.1 Description

A device configured in non-discoverable mode will not be discovered by any device that is performing either the general discovery procedure or the limited discovery procedure.

9.2.2.2 Conditions

A device in the non-discoverable mode that sends advertising events shall not set the ‘LE General Discoverable Mode’ flag or ‘LE Limited Discoverable Mode’ flag in the Flags AD type. A Peripheral device in the non-connectable mode may send non-connectable undirected advertising events or scannable undirected advertising events or may not send advertising packets.

9.2.3 Limited Discoverable Mode

9.2.3.1 Description

Devices configured in the limited discoverable mode are discoverable for a limited period of time by other devices performing the limited or general device discovery procedure. The limited discoverable mode is typically used when a user performs a specific action to make the device discoverable for a limited period of time.

9.2.3.2 Conditions

While a device is in the Peripheral role the device may support the limited discoverable mode. While a device is in the Broadcaster, Observer or Central role the device shall not support the limited discoverable mode.

A device in the limited discoverable mode sends either non-connectable advertising events, scannable undirected advertising events or connectable undirected advertising events.

While in the limited discoverable mode the device shall send advertising event types with the advertising data including the Flags AD type as defined in Section 11.1.3 with the following flags set as described:

- The LE Limited Discoverable Mode flag shall be set to one
- The LE General Discoverable Mode flag shall be set to zero.
- A device of the LE-only device type shall set the 'BR/EDR Not Supported' flag to one, the 'Simultaneous LE and BR/EDR to Same Device Capable (Controller)' flag to zero, and the 'Simultaneous LE and BR/EDR to Same Device Capable (Host)' flag to zero

The advertising data should also include the following AD types to enable a faster connectivity experience:

- TX Power Level AD type defined in Section 11.1.5.
- Local Name AD type defined in Section 11.1.2.
- Service UUIDs AD type defined in Section 11.1.1.
- Slave Connection Interval Range AD type as defined in Section 11.1.7.

Devices shall remain in the limited discoverable mode no longer than T_{GAP}(lim_adv_timeout).

While a device is in limited discoverable mode the Host configures the Controller as follows:

- The Host shall set the advertising filter policy to 'process scan and connection requests from all devices'
- The Host should set the minimum advertising interval to T_{GAP}(lim_disc_adv_int_min)
- The Host should set the maximum advertising interval to T_{GAP}(lim_disc_adv_int_max).

The device shall remain in limited discoverable mode until a connection is established or the Host terminates the mode.

If a device is in the limited discoverable mode and in the undirected connectable mode the advertising interval values should be set as defined in Section 9.3.4

A device in the limited discoverable mode shall not set both the LE Limited Discoverable Flag and the LE General Discoverable Flag to one.

Note: Data that change frequently should be placed in the advertising data and static data should be placed in the scan response data.

Note: The choice of advertising interval is a trade-off between power consumption and device discovery time.

9.2.4 General Discoverable Mode

9.2.4.1 Description

Devices configured in general discoverable mode are intended to be discoverable by devices performing the general discovery procedure. The general discoverable mode is typically used when the device is intending to be discoverable for a long period of time.

9.2.4.2 Conditions

While a device is in the Peripheral role the device may support the general discoverable mode. While a device is in the Broadcaster, Observer or Central role the device shall not support the general discoverable mode.

A device in the general discoverable mode sends either non-connectable advertising events, scannable undirected advertising events, or connectable undirected advertising events.

While in general discoverable mode the device shall send advertising events with the advertising data including the Flags AD data type as defined in Section 11.1.3 with the following flags:

- The LE Limited Discoverable Mode flag set to zero
- The LE General Discoverable Mode flag set to one
- A device of the LE-only device type shall set the ‘BR/EDR Not Supported’ flag to one, the ‘Simultaneous LE and BR/EDR to Same Device Capable (Controller)’ flag to zero, and the ‘Simultaneous LE and BR/EDR to Same Device Capable (Host)’ flag to zero

The advertising data should also include the following AD types to enable a faster connectivity experience:

- TX Power Level AD type as defined in Section 11.1.5.
- Local Name AD type as defined in Section 2.
- Service UUIDs AD type as defined in Section 11.1.1**.**
- Slave Connection Interval Range AD type as defined in Section 11.1.7.

While a device is in general discoverable mode the Host configures the Controller as follows:

- The Host shall set the advertising filter policy to ‘process scan and connection requests from all devices’
- The Host should set the minimum advertising interval to T_{GAP}(gen_disc_adv_int_min)

- The Host should set the maximum advertising interval to T_{GAP}(gen_disc_adv_int_max).

The device shall remain in general discoverable mode until a connection is established or the Host terminates the mode.

If a device is in the general discoverable mode and in the directed connectable mode or the non-connectable mode, the advertising interval values should be set as defined in Section 9.3.3.

A device in the general discoverable mode shall not set both the LE Limited Discoverable Flag and the LE General Discoverable Flag to one.

Note: Data that change frequently should be placed in the advertising data and static data should be placed in the scan response data.

Note: The choice of advertising interval is a trade-off between power consumption and device discovery time.

9.2.5 Limited Discovery Procedure

9.2.5.1 Description

A device performing the limited discovery procedure receives the device address, advertising data and scan response data from devices in the limited discoverable mode only.

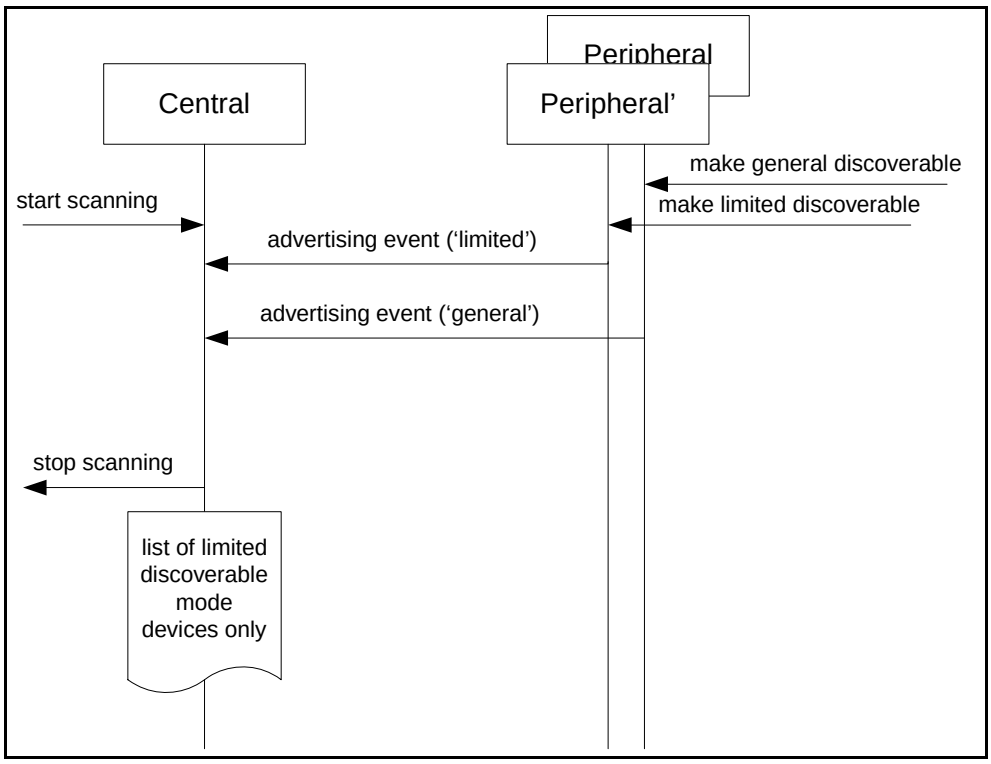


Figure 9.1: A Central performing limited discovery procedure discovering Peripherals in the limited discoverable mode

9.2.5.2 Conditions

While a device is in the Central role the device may support the limited discovery procedure. While a device is in the Broadcaster, Observer or Peripheral role the device shall not support the limited discovery procedure.

When a Host performs the limited discovery procedure, the Host configures the Controller as follows:

1. The Host shall set the scanner filter policy to 'process all advertising packets'
2. The Host should set the scan interval of T_{GAP}(lim_disc_scan_int)
3. The Host should set the scan window of T_{GAP}(lim_disc_scan_wind)
4. The Host should configure the Controller to use active scanning.

The Host shall begin scanning for advertising packets and should continue for a minimum of T_{GAP}(lim_disc_scan_min), unless the host ends the limited discovery procedure.

The Host shall check for the Flags AD type in the advertising data. If the Flags AD type is present and the LE Limited Discoverable Flag is set to one then the Host shall consider the device as a discovered device, otherwise the advertising data shall be ignored. The Flag AD type is defined in Section 11.1.3. The advertising data of the discovered device may contain data with other AD

types, e.g. Service UUIDs AD type, TX Power Level AD type, Local Name AD type, Slave Connection Interval Range AD type. The Host may use the data in performing any of the connection establishment procedures.

9.2.6 General Discovery Procedure

9.2.6.1 Description

A device performing the general discovery procedure receives the device address, advertising data and scan response data from devices in the limited discoverable mode or the general discoverable mode.

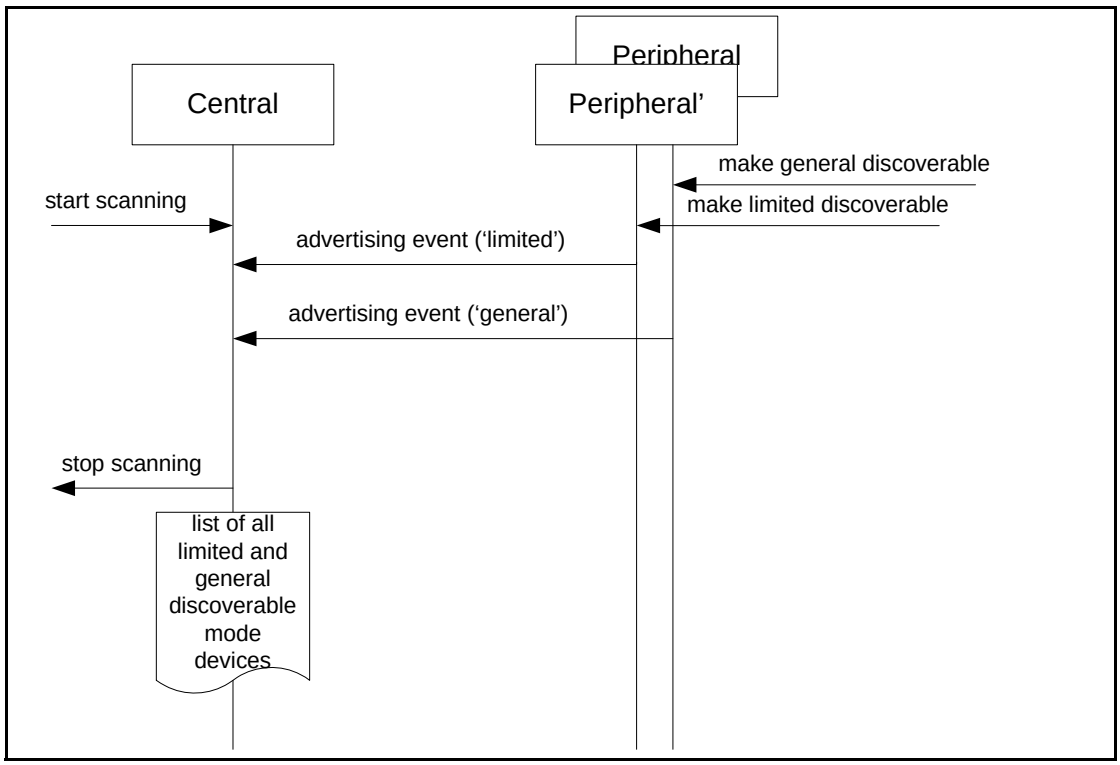


Figure 9.2: A Central performing general discovery procedure discovering Peripherals in the limited discoverable mode and general discoverable mode.

9.2.6.2 Conditions

While a device is in the Central role the device shall support the general discovery procedure. While a device is in the Broadcaster, Observer or Peripheral role the device shall not support the general discovery procedure.

When a Host performs the general discovery procedure, the Host configures the Controller as follows:

1. The Host shall set the scanner filter policy to 'process all advertising packets'
2. The Host should set the scan interval of T_{GAP}(gen_disc_scan_int)

3. The Host should set the scan window of T_{GAP}(gen_disc_scan_wind)
4. The Host should configure the Controller to use active scanning

The Host shall begin scanning for advertising packets and should continue for a minimum of T_{GAP}(gen_disc_scan_min). The procedure may be terminated early by the Host.

The Host shall check for the Flags AD type in the advertising data. If the Flags AD type is present and either the LE General Discoverable Mode flag is set to one or the LE Limited Discoverable Mode flag is set to one then the Host shall consider the device as a discovered device, otherwise the advertising data shall be ignored. The advertising data of the discovered device may contain data with other AD types, e.g. Service UUIDs AD type, TX Power Level AD type, Local Name AD type, Slave Connection Interval Range AD type. The Host may use the data in performing any of the connection establishment procedures as defined in Section 9.3.

9.2.7 Name Discovery Procedure

9.2.7.1 Description

The name discovery procedure is used to obtain the Bluetooth Device Name of a remote connectable device.

9.2.7.2 Conditions

If the complete device name is not acquired while performing either the limited discovery procedure or the general discovery procedure, then the name discovery procedure may be performed.

The name discovery procedure shall be performed as follows:

1. The Host shall establish a connection using one of the connection establishment procedures as defined in Section 9.3.
2. The Host shall read the device name characteristic using the GATT procedure Read Using Characteristic UUID [Vol. 3], Part G Section 4.8.2
3. The connection may be terminated after the GATT procedure has completed.

9.3 CONNECTION MODES AND PROCEDURES

The connection modes and procedures allow a device to establish a connection to another device.

A Peripheral device may support the privacy feature as defined in Section 10.7 in the directed and undirected modes. A Central may connect to a Peripheral that supports the privacy feature using only the general connection establishment procedure.

When devices are connected, the parameters of the connection can be updated with the Connection Parameter Update procedure. The connected device may terminate the connection using the Terminate Connection procedure. The requirements for a device to support the connection modes and procedures are defined in Table 9.3.

9.3.1 Requirements

Connection Modes and Procedures	Ref.	Peripheral	Central	Broadcaster	Observer
Non-connect-able mode	9.3.2	M	E	M	M
Directed con-nectable mode	9.3.3	O	E	E	E
Undirected con-nectable mode	9.3.4	M	E	E	E
Auto connection establishment procedure	9.3.5	E	O	E	E
General connec-tion establish-ment procedure	9.3.6	E	C1	E	E
Selective con-nection estab-lishment procedure	9.3.7	E	O	E	E
Direct connec-tion establish-ment procedure	9.3.8	E	M	E	E
Connection parameter update proce-dure	9.3.9	O	M	E	E
Terminate con-nection proce-dure	9.3.10	M	M	E	E
C1: Mandatory if privacy feature is supported, else optional					

Table 9.3: Connection modes and procedures requirements

9.3.2 Non-Connectable Mode

9.3.2.1 Description

A device in the non-connectable mode shall not allow a connection to be established.

9.3.2.2 Conditions

While a device is in the Peripheral role the device shall support the non-connectable mode. A device in the Central role cannot send advertising events, therefore the device is considered to support the non-connectable mode. A

device in the Broadcaster or Observer role cannot establish a connection, therefore the device is considered to support the non-connectable mode.

A Peripheral device in the non-connectable mode may send non-connectable undirected advertising events or scannable undirected advertising events.

9.3.3 Directed Connectable Mode

9.3.3.1 Description

A device in the directed connectable mode shall accept a connection request from a known peer device performing the auto connection establishment procedure or the general connection establishment procedure.

9.3.3.2 Conditions

While a device is in the Peripheral role the device may support the direct connectable mode. This mode shall only be used if the device has a known peer device address. While a device is in the Broadcaster, Observer, or the Central role the device shall not support the direct connectable mode.

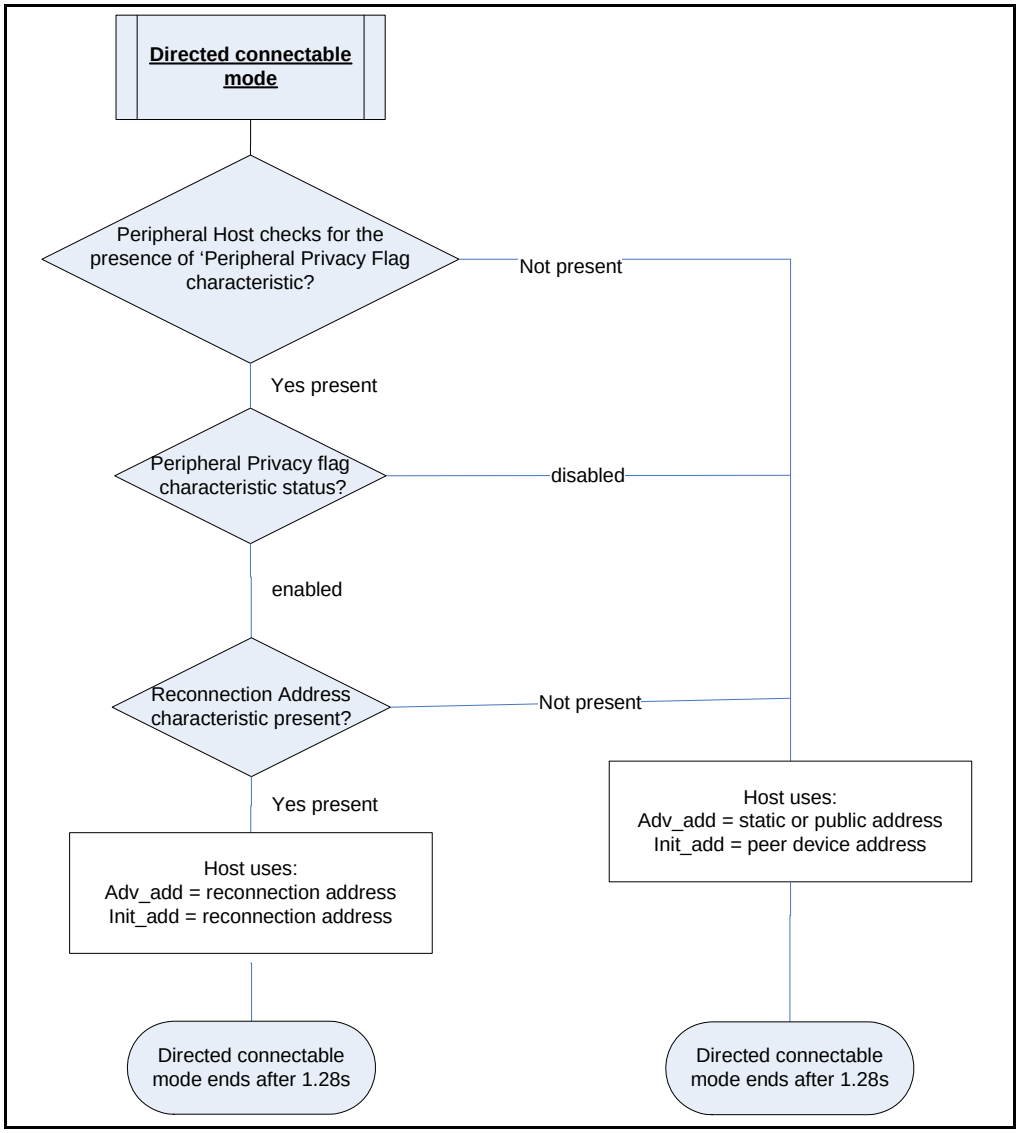


Figure 9.3: Flow chart for a device entering the directed connectable mode

A Host entering the directed connectable Mode shall perform the following steps:

1. If the Peripheral Privacy Flag characteristic exists on the local device, the Host shall proceed to step 2, otherwise the Host shall proceed to step 5.
2. If the value in the Peripheral Privacy Flag characteristic is set to 1 (privacy enabled), the Host shall proceed to step 3, otherwise the Host shall proceed to step 5.
3. If the Reconnection Address characteristic exists on the local device, the Host shall proceed to step 4, otherwise the Host shall proceed to step 5.
4. The Host uses the address in the Reconnection Address characteristic as the initiator's device address and the advertiser device address in the directed connectable advertising events. The Host shall proceed to step 6.

5. The Host uses the static or the public address of this device as the advertiser's device address. The Host uses the known peer device address as the initiator's device address. The Host shall proceed to step 6.
6. The Host should configure the Controller with the minimum advertising interval set to T_{GAP}(dir_conn_adv_int_min) and the maximum advertising interval set to T_{GAP}(dir_conn_adv_int_max). The Host shall configure the Controller to send directed connectable advertising events.

Note: When a device in the directed connectable mode establishes a connection, the device will exit this mode and enter the non-connectable mode.

9.3.4 Undirected Connectable Mode

9.3.4.1 Description

A device in the undirected connectable mode shall accept a connection request from a device performing the auto connection establishment procedure or the general connection establishment procedure.

9.3.4.2 Conditions

While a device is in the Peripheral role the device shall support the undirected connectable mode. While a device is in the Broadcaster, Observer, or the Central role the device shall not support the undirected connectable mode.

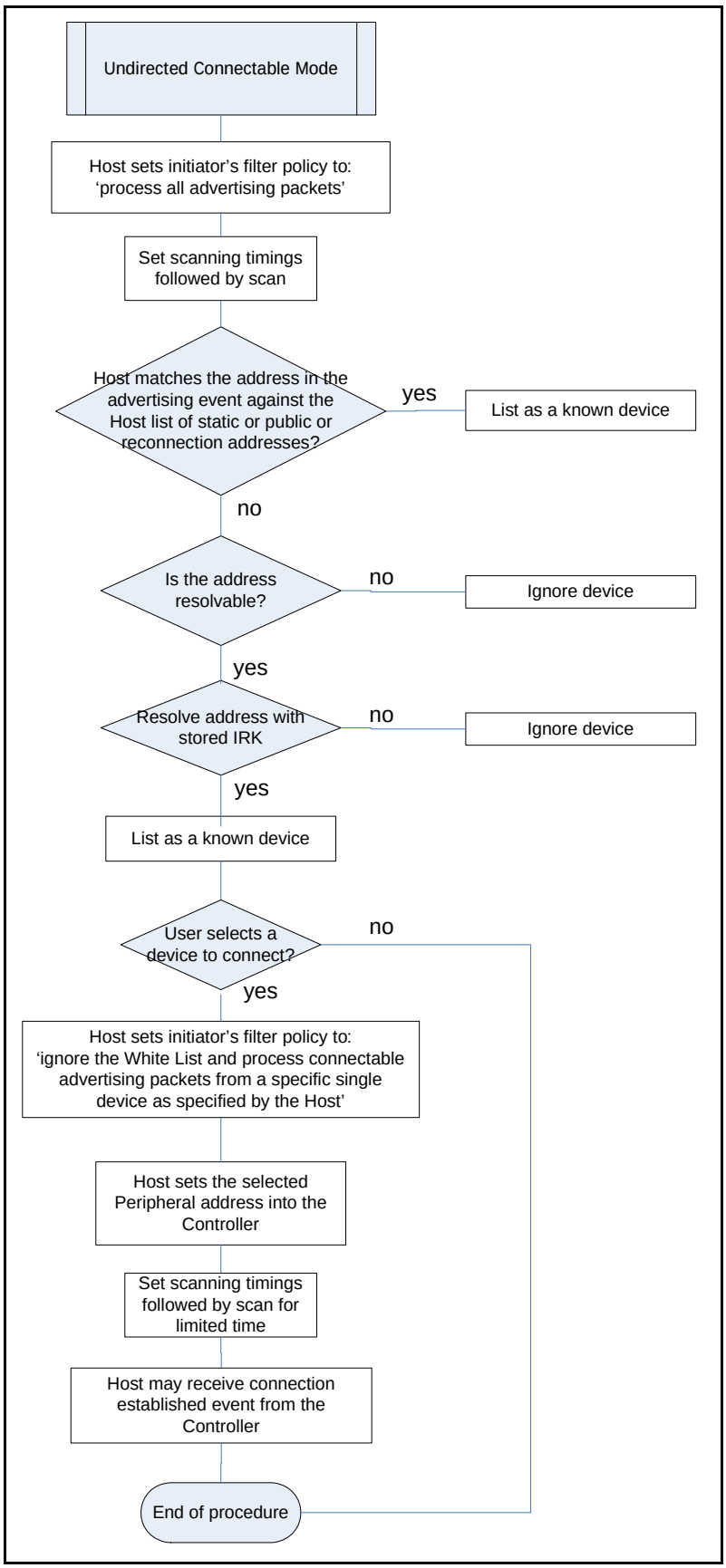


Figure 9.4: Flow chart for a device entering the undirected connectable mode

A Host entering the undirected connectable mode shall perform the following steps:

1. If the Peripheral Privacy Flag characteristic exists, the Host shall proceed to step 2, otherwise the Host shall proceed to step 4.
2. If the value in the Peripheral Privacy Flag characteristic is set to 1 (privacy enabled), the Host shall proceed to step 3, otherwise the Host shall proceed to step 4.
3. The Host shall generate a resolvable private address using the 'resolvable private address generation procedure' as defined in Section 10.8.2.2. The Host shall set a timer equal to or greater than T_{GAP}(private_addr_int). The Host shall configure the Controller to use the resolvable private address as the advertiser's device address in the connectable undirected advertising events. The Host should set the advertising filter policy to either 'process scan and connection requests only from devices in the White List' or 'process scan and connection requests from all devices'. The Host shall configure the Controller to send undirected connectable advertising events with the minimum advertising interval set to T_{GAP}(undir_conn_adv_int_min) and the maximum advertising interval set to T_{GAP}(undir_conn_adv_int_max). If privacy is disabled then the Host shall disable the T_{GAP}(private_addr_int) timer and proceed to step 4. When the T_{GAP}(private_addr_int) timer expires, the Host shall configure the Controller to stop advertising and repeat step 3.
4. The Host uses the static or the public address of the local device as the advertiser's device address. The Host should configure the Controller with the minimum advertising interval set to T_{GAP}(undir_conn_adv_int_min) and the maximum advertising interval set to T_{GAP}(undir_conn_adv_int_max). The Host shall configure the Controller to send undirected connectable advertising events.

Note: When a device in the undirected connectable mode establishes a connection, the device will exit this mode and enter the non-connectable mode.

9.3.5 Auto Connection Establishment Procedure

9.3.5.1 Description

The auto connection establishment procedure allows the Host to configure the Controller to autonomously establish a connection with one or more devices in the directed connectable mode or the undirected connectable mode. White Lists in the Controller are used to store device addresses. This procedure uses the initiator White List in the Controller. The Controller autonomously establishes a connection with a device with the device address that matches the address stored in the While List.

9.3.5.2 Conditions

While a device is in the Central role the device may support the auto connection establishment procedure. While a device is in the Broadcaster, Observer, or Peripheral role the device shall not support the auto connection establishment procedure.

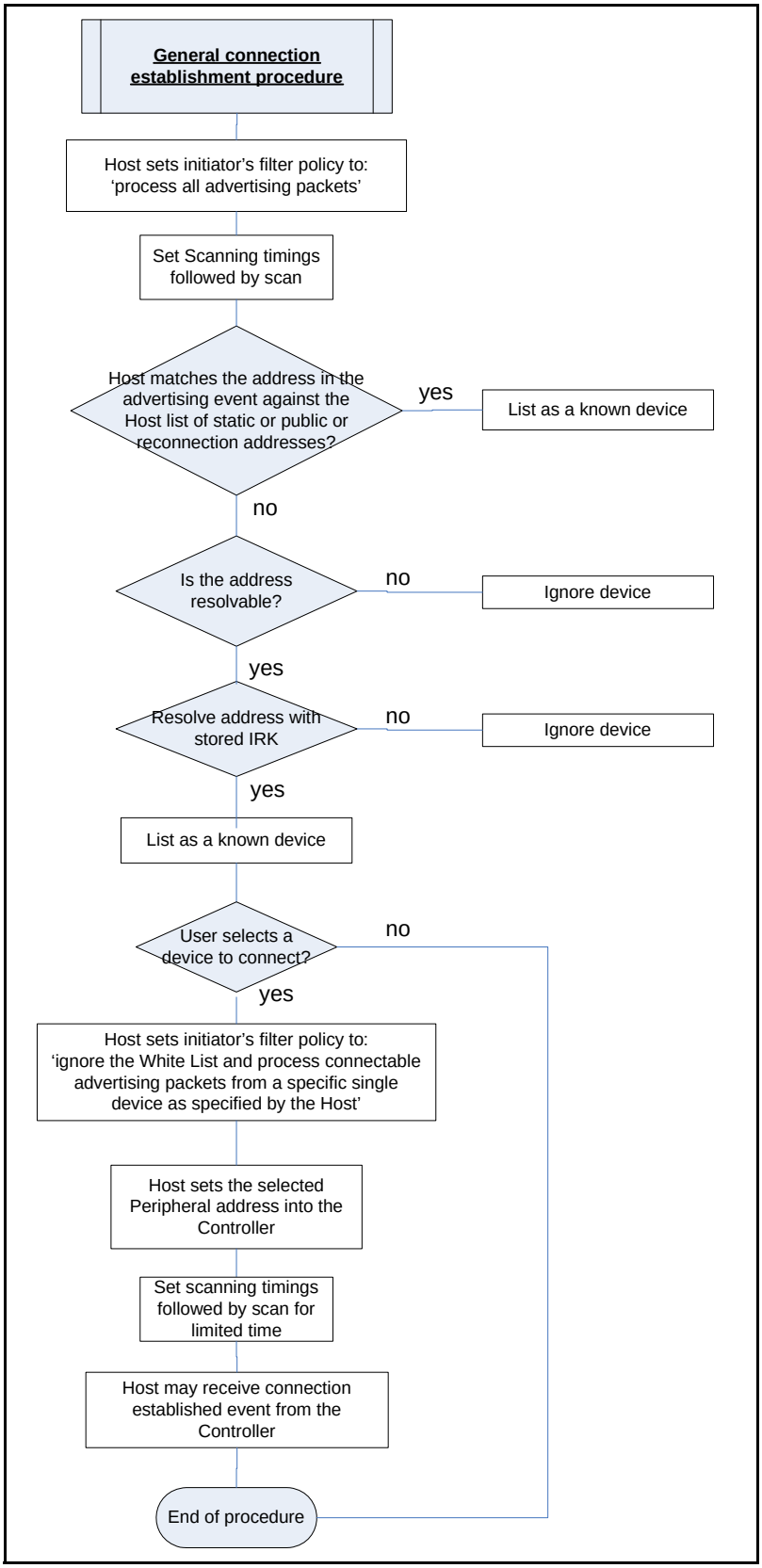


Figure 9.5: Flow chart for a device performing the auto connection establishment procedure

When a Host performs the auto connection establishment procedure, the Host configures the Controller as follows:

1. The Host shall write the list of device addresses that are to be auto connected to into the White List.
2. The Host shall set the initiator filter policy to 'process connectable advertising packets from all devices in the White List'.
3. The Host should set the minimum connection interval to be T_{GAP}(conn_est_int_min).
4. The Host should set the maximum connection interval to be T_{GAP}(conn_est_int_max)
5. The Host should set the slave latency to be T_{GAP}(conn_est_latency)
6. The Host should set the scan interval of T_{GAP}(auto_conn_est_scan_int) and the scan window of T_{GAP}(auto_conn_est_scan_wind).
7. The Host shall initiate a connection. For a privacy enabled device as defined in Section 10.7, the Host shall use a non-resolvable private address.

This procedure is terminated when a connection is established or when the Host terminates the procedure.

9.3.6 General Connection Establishment Procedure

9.3.6.1 Description

The general connection establishment procedure allows the Host to establish a connection with a set of known peer devices in the directed connectable mode or the undirected connectable mode.

9.3.6.2 Conditions

While a device is in the Central role the device shall support the general connection establishment procedure if the device supports the privacy feature, otherwise the device in the Central role may support the general connection establishment procedure. While a device is in the Broadcaster, Observer, or Peripheral role the device shall not support the general connection establishment procedure.

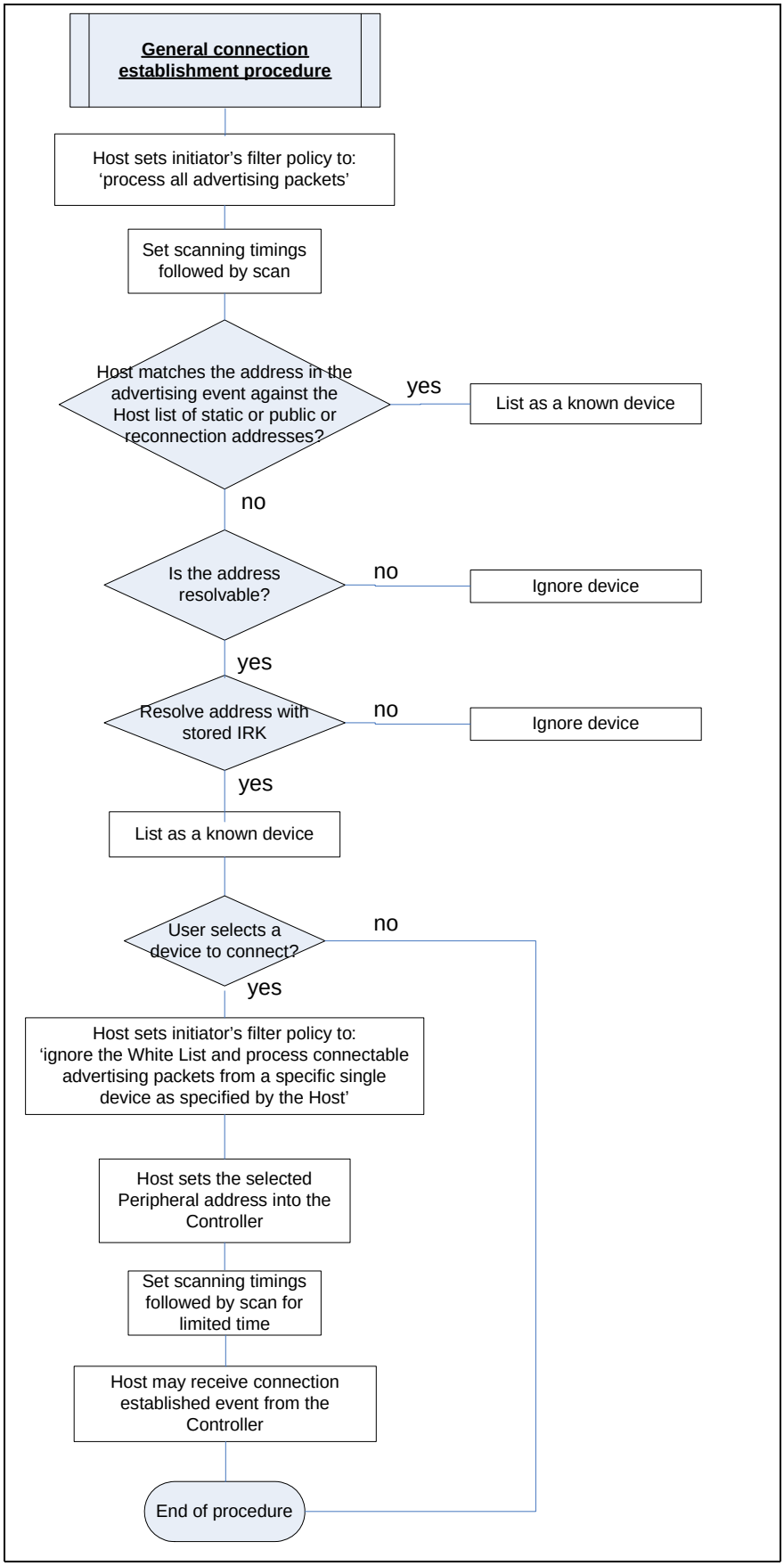


Figure 9.6: Flow chart for a device performing the general connection establishment procedure

When a Host performs the general connection establishment procedure, the Host configures the Controller as follows:

1. The Host shall set the scanner filter policy to 'process all advertising packets'.
2. The Host should set the scan interval to T_{GAP}(gen_conn_est_scan_int).
3. The Host should set the scan window to T_{GAP}(gen_conn_est_scan_wind).
4. The Host shall start scanning. For a privacy enabled device as defined in Section 10.7, the Host shall use a non-resolvable private address if the Host is in active scanning.

If the privacy feature is supported by the local device, and the Host discovers a device with a resolvable private address, the Host may resolve the resolvable private address by performing the 'resolvable private address resolution procedure' as defined in Section 10.8.2.3. If the resolution is successful, the Host considers the discovered device as a known device, otherwise the discovered device is ignored.

When the Host discovers a device to which the Host may attempt to connect, the Host shall stop the scanning, and initiate a connection using the direct connection establishment procedure.

When the Host of the Central that supports privacy establishes a connection with a Peripheral that has the 'Peripheral Privacy Flag' set to 1 (privacy enabled), the Host of the Central should generate a new reconnection address and update the reconnection address characteristic value in the Peripheral before any application data is sent by the Host of the Central. The Peripheral shall use the old reconnection address if the reconnection address has not been updated.

If the Central does not support privacy, when the Host of the Central establishes a connection with a Peripheral that has the 'Peripheral Privacy Flag' set to 1 (privacy enabled), the Host of the Central may set the Peripheral Privacy Flag to 0 (privacy disabled). If the Host of the Central is unable to disable the Peripheral Privacy Flag, the Central should not bond with the Peripheral.

This procedure is terminated when a connection is established or when the Host terminates the procedure.

9.3.7 Selective Connection Establishment Procedure

9.3.7.1 Description

The selective connection establishment procedure allows the Host to establish a connection with the Host selected connection configuration parameters with a set of devices in the White List.

9.3.7.2 Conditions

While a device is in the Central role the device may support the selective connection establishment procedure. While a device is in the Broadcaster, Observer, or the Peripheral role the device shall not support the selective connection establishment procedure.

When a Host performs the selective connection establishment procedure, the Host configures the Controller as follows:

1. The Host shall write the list of device addresses that are to be selectively connected to into the White List.
2. The Host shall set the scanner filter policy to ‘process advertising packets only from devices in the White List’.
3. The Host should set the scan interval to T_{GAP}(sel_conn_est_scan_int).
4. The Host should set the scan window to T_{GAP}(sel_conn_est_scan_wind).
5. The Host shall start active scanning or passive scanning.

When the Host discovers one of the peer devices it is connecting to, the Host shall stop scanning, and initiate a connection using the direct connection establishment procedure with the connection configuration parameters for that peer device.

This procedure is terminated when a connection is established or when the Host terminates the procedure.

9.3.8 Direct Connection Establishment Procedure

9.3.8.1 Description

The direct connection establishment procedure allows the Host to establish a connection with the Host selected connection configuration parameters with a single peer device.

9.3.8.2 Conditions

While a device is in the Central role the device shall support the direct connection establishment procedure. While a device is in the Broadcaster, Observer, or the Peripheral role the device shall not support the direct connection establishment procedure.

When a Host performs the direct connection establishment procedure, the Host configures the Controller as follows:

1. The Host shall set the initiator filter policy to ‘ignore the White List and process connectable advertising packets from a specific single device specified by the Host’.

2. The Host shall set the peer address to the device address of the specific single device.
3. The Host shall set the connection configuration parameters for the peer device.

The Host shall initiate a connection. For a privacy enabled device as defined in Section 10.7, the Host shall use a non-resolvable private address.

This procedure is terminated when a connection is established or when the Host terminates the procedure.

9.3.9 Connection Parameter Update Procedure

9.3.9.1 Description

The connection parameter update procedure allows a Peripheral or Central to update the Link Layer connection parameters of an established connection.

9.3.9.2 Conditions

While a device is in the Central role the device shall support the connection parameter update procedure. While a device is in the Peripheral role the device may support the connection parameter update procedure. While a device is in the Broadcaster or the Observer role the device shall not support the connection parameter update procedure.

A Central initiating the connection parameter update procedure shall use the Link Layer Connection Update procedure defined in [Vol. 6], Part B Section 5.1.1 with the required connection parameters.

A Peripheral initiating the connection parameter update procedure shall use the L2CAP Connection Parameter Update Request command defined in [Vol. 1], Part A Section 4.2 with the required connection parameters. The Peripheral shall not send an L2CAP Connection Parameter Update Request command within T_{GAP}(conn_param_timeout) of an L2CAP Connection Parameter Update Response being received. If the updated connection parameters are unacceptable to the Peripheral then it may disconnect the connection with the error code «Unacceptable Connection Interval». Peripherals should be tolerant of connection intervals given to it from the Central.

9.3.10 Terminate Connection Procedure

9.3.10.1 Description

The terminate connection procedure allows a Host to terminate the connection with a peer device.

9.3.10.2 Conditions

Centrals and Peripherals shall support the terminate connection procedure. Broadcasters and Observers shall not support the terminate connection procedure.

The Host initiating the terminate connection procedure shall use the Link Layer Termination Procedure defined in [Vol. 6], Part B Section 5.1.6.

9.4 BONDING MODES AND PROCEDURES

Bonding allows two connected devices to exchange and store security and identity information to create a trusted relationship. The security and identity information as defined in [Vol. 3], Part H Section 2.4.1 is also known as the bonding information. When the devices store the bonding information, it is known as the phrases ‘devices have bonded’ or ‘a bond is created’.

There are two modes for bonding, non-bondable mode and bondable mode. Bonding may only occur between two devices in bondable mode. The requirements for a device to support the bonding modes and procedure are defined in Table 9.4.

9.4.1 Requirements

Bonding	Ref.	Peripheral	Central
Non-Bondable mode	9.4.2	M	M
Bondable mode	9.4.3	O	O
Bonding procedure	9.4.4	O	O

Table 9.4: Bonding Compliance Requirements

9.4.2 Non-Bondable Mode

9.4.2.1 Description

A device in the non-bondable mode does not allow a bond to be created with a peer device.

9.4.2.2 Conditions

While a device is in the Peripheral or the Central role the device shall support the non-bondable mode. If a device does not support pairing as defined in the Security Manager section then it is considered to be in non-bondable mode.

If Security Manager pairing is supported, the Host shall set the Bonding_Flags to ‘No Bonding’ as defined in [Vol. 3], Part H Section 3.5.1 and bonding information shall not be exchanged or stored.

9.4.3 Bondable Mode

9.4.3.1 Description

A device in the bondable mode allows a bond to be created with a peer device in the bondable mode.

9.4.3.2 Conditions

The Host shall set the Bonding_Flags to ‘General Bonding’ during the pairing procedure.

9.4.4 Bonding Procedure

9.4.4.1 Description

The bonding procedure may be performed when a non-bonded device tries to access a service that requires bonding. The bonding procedure may be performed for the purpose of creating a bond between two devices.

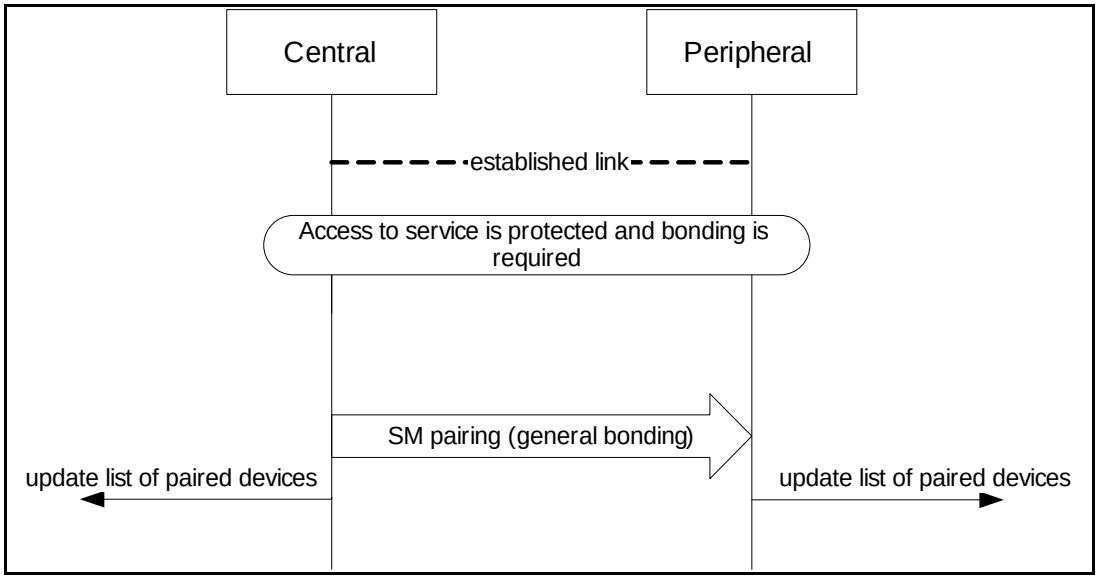


Figure 9.7: The Bonding procedure

9.4.4.2 Conditions

While a device is in the Peripheral or the Central role the device may support the Bonding procedure. While a device is in the Broadcaster or the Observer role the device shall not support the bonding procedure.

The Host of the Central initiates the pairing process as defined in [Vol. 3], Part C Section 2.1 with the Bonding_Flags set to Bonding as defined in [Vol. 3], Part H Section 3.5.1. If the peer device is in the bondable mode, the devices shall exchange and store the bonding information in the security database.

If a device has privacy enabled as defined in Table 10.7, the Host should send it's IRK to the peer device and request the IRK of the peer device during the pairing procedure. The Host can abort the pairing procedure if the authentication requirements are not sufficient to distribute the IRK. If the pairing procedure fails due to authentication requirements and IRK distribution was requested, the pairing procedure should be retried without requesting IRK distribution.

10 LE SECURITY ASPECTS

This section defines the modes and procedures that relate to the security of a connection. The following modes and procedures are defined:

- LE security mode 1
- LE security mode 2
- Authentication procedure
- Authorization procedure
- Connection data signing procedure
- Authenticate signed data procedure

Requirements for a device to support the LE security modes and procedures is shown in Table 10.1.

10.1 REQUIREMENTS

Security Modes and Procedures	Ref.	Broadcaster	Observer	Peripheral	Central
LE Security mode 1	10.2.1	E	E	O	O
LE Security mode 2	10.2.2	E	E	O	O
Authentication procedure	10.3	E	E	O	O
Authorization procedure	10.5	E	E	O	O
Connection data signing procedure	10.4.1	E	E	O	O
Authenticate signed data procedure	10.4.2	E	E	O	O

Table 10.1: Requirements related to security modes and procedures

10.2 LE SECURITY MODES

The security requirements of a device, a service or a service request are expressed in terms of a security mode and security level. Each service or service request may have its own security requirement. The device may also have a security requirement. A physical connection between two devices shall operate in only one security mode.

There are two LE security modes, LE security mode 1 and LE security mode 2.

10.2.1 LE Security Mode 1

LE security mode 1 has three security levels:

1. No security (No authentication and no encryption)
2. Unauthenticated pairing with encryption
3. Authenticated pairing with encryption

A connection operating in LE security mode 1 level 2 shall also satisfy the security requirements for LE security mode 1 level 1.

A connection operating in LE security mode 1 level 3 shall also satisfy the security requirements for LE security mode 1 level 2 or LE security mode 1 level 1.

A connection operating in LE security mode 1 level 3 shall also satisfy the security requirements for LE security mode 2.

10.2.2 LE Security Mode 2

LE security mode 2 has two security levels:

1. Unauthenticated pairing with data signing
2. Authenticated pairing with data signing

LE security mode 2 shall only be used for connection based data signing.

Data signing as defined in Section 10.4 shall not be used when a connection is operating in LE security mode 1 level 2 or LE security mode 1 level 3.

10.2.3 Mixed Security Modes Requirements

If there are requirements for both LE security mode 1 and LE security mode 2 level 2 for a given physical link then LE security mode 1 level 3 shall be used.

If there are requirements for both LE security mode 1 level 3 and LE security mode 2 for a given physical link then LE security mode 1 level 3 shall be used.

If there are requirements for both LE security mode 1 level 2 and LE security mode 2 level 1 for a given physical link then LE security mode 1 level 2 shall be used.

10.3 AUTHENTICATION PROCEDURE

The authentication procedure describes how the required security is established when a device initiates a service request to a remote device and when a device receives a service request from a remote device. The authentication procedure covers both LE security mode 1 and LE security mode 2. The authentication procedure shall only be initiated after a connection has been established.

There are two types of pairing: authenticated pairing or unauthenticated pairing. Authenticated pairing involves performing the pairing procedure defined in [Vol. 3], Part H Section 2.1 with the authentication set to ‘MITM protection’. Unauthenticated pairing involves performing the pairing procedure with authentication set to ‘No MITM protection’.

10.3.1 Receiving a Service Request

A service request is a request to perform a GATT procedure. A service request may also be a request to perform a procedure using other higher layer protocols on top of L2CAP.

When a device receives a service request from a remote device it shall behave as follows:

- The Server’s security database specifies the security settings required to accept a service request from a remote device. If no security is required, the service request shall be accepted; otherwise pairing is required before the service request can be accepted.
- If pairing has not occurred or an authenticated pairing is required but only an unauthenticated pairing has occurred, the service request shall be rejected with the error code “Insufficient Authentication“.
- If the required pairing has occurred and encryption is required (LE security mode 1) but encryption is not enabled, the service request shall be rejected with the error code “Insufficient Encryption”. If the encryption is enabled with insufficient key size then the service request shall be rejected with the error code “Insufficient Encryption Key Size.”
- If authenticated pairing has occurred and encryption is not required (LE security mode 2) then Data Signing shall be used if CSRK has been exchanged, else the Host shall re-pair and exchange CSRK before proceeding with Data Signing.

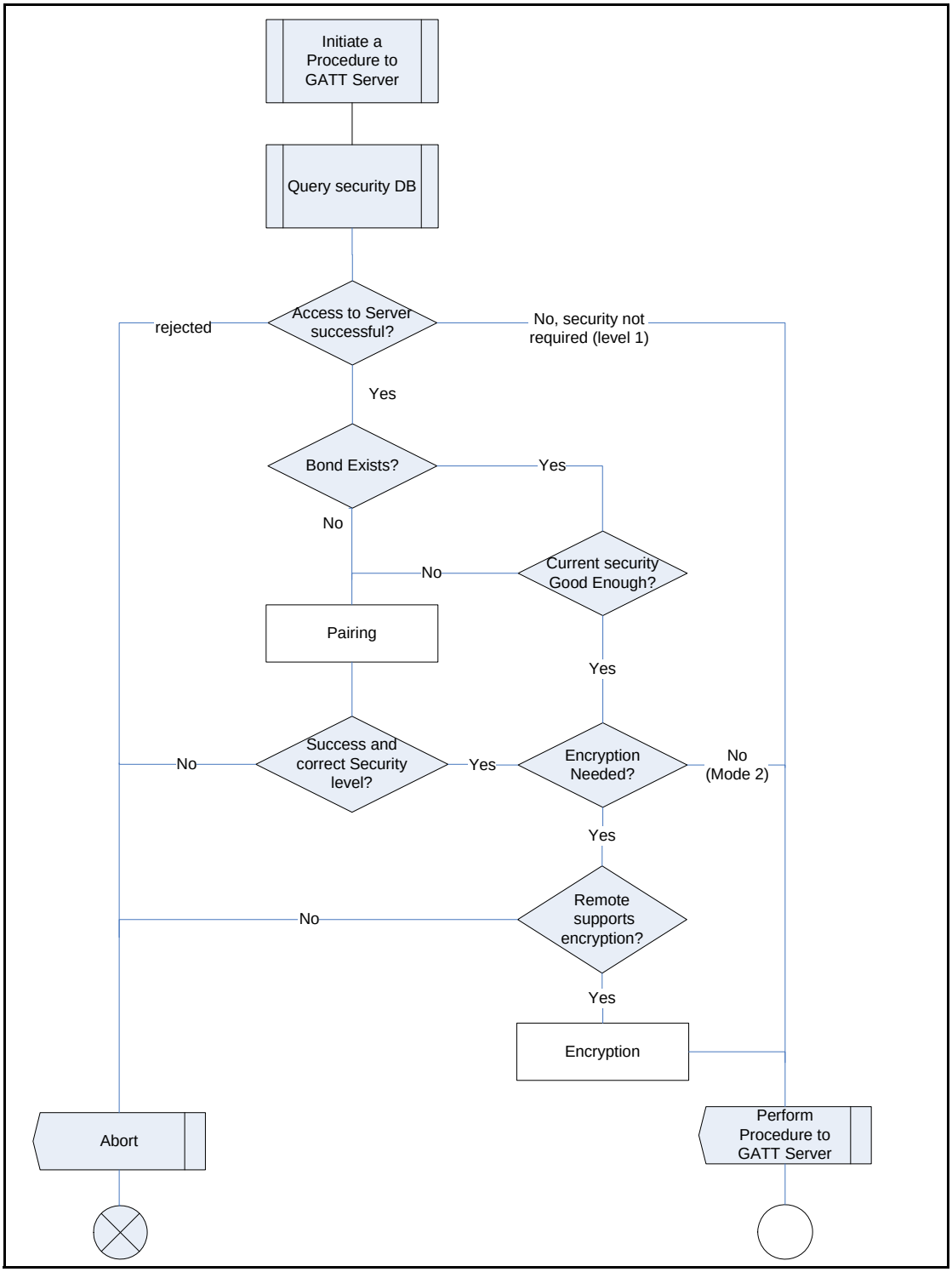


Figure 10.1: Flow chart for a server handling the service request

10.3.2 Initiating a Service Request

When a device initiates a service request to a remote device it shall behave as follows:

- The Client's security database specifies the security required to initiate a service request. If no security is required then the service request shall proceed otherwise pairing is required. If security is required but pairing has not occurred in a previous bonding procedure then pairing shall occur during the current connection.
- If pairing has not occurred or an authenticated pairing is required but only an unauthenticated pairing has occurred, the pairing procedure shall be executed with the required authentication settings. If the pairing procedure fails then the service request shall be aborted.
- If authenticated pairing has occurred but the encryption key size is insufficient the pairing procedure shall be executed with the required encryption key size. If the pairing procedure fails then the service request shall be aborted.
- If the required pairing has occurred and encryption is required (LE security mode 1) then encryption shall be enabled before the service request proceeds. If encryption is required but encryption is not enabled and the remote device supports encryption then encryption shall be enabled using the encryption procedure as defined in Section 10.6. Once encryption is enabled the service request shall proceed. If encryption is required but not supported by the remote device then the service request shall be aborted.
- If encryption is not required and authenticated pairing has occurred (LE security mode 2) then the data signing procedure shall be used when making the service request.

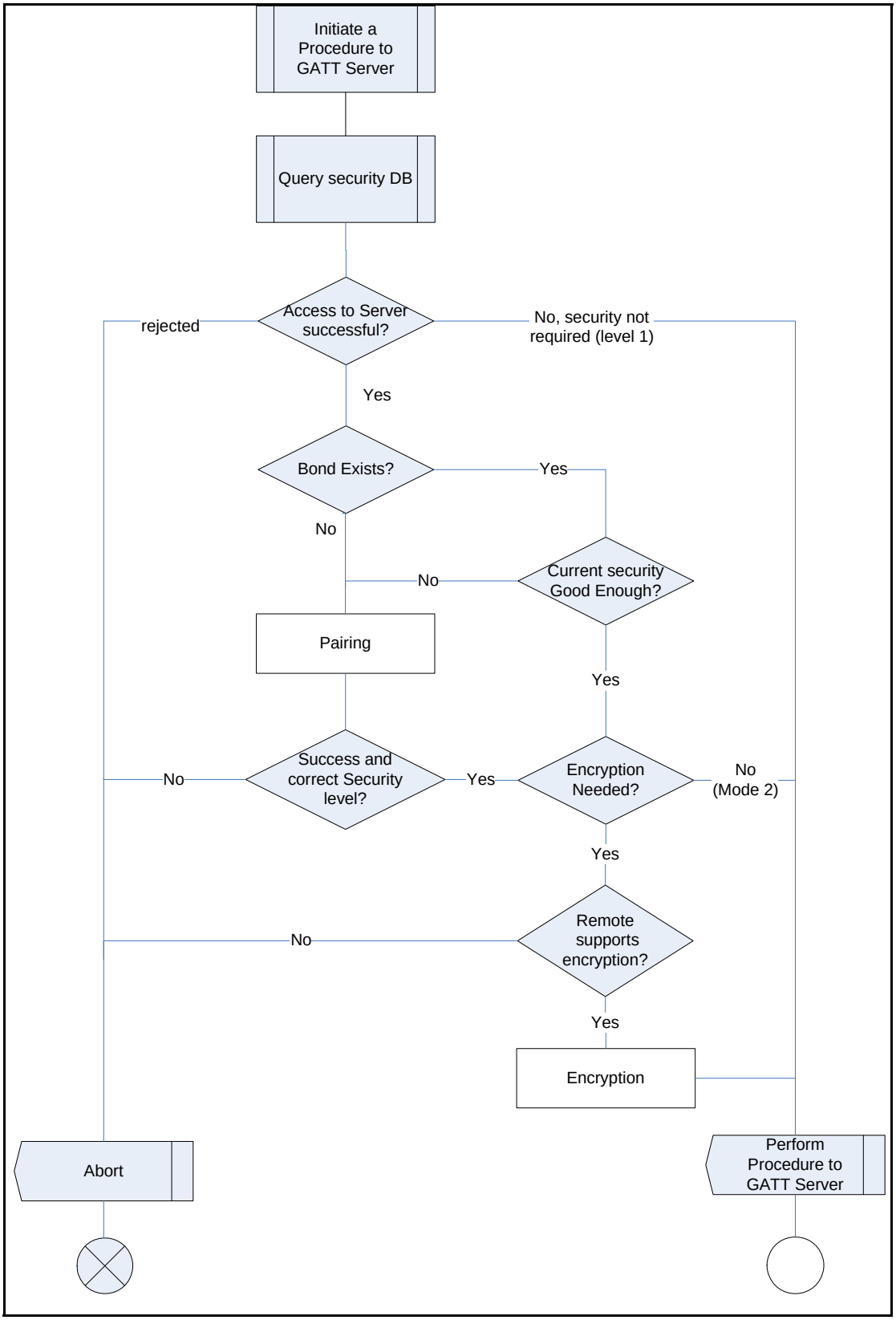


Figure 10.2: Flow chart for a client issuing a service request

10.4 DATA SIGNING

The data signing is used for transferring authenticated data between two devices in an unencrypted connection. The data signing method is used by services that require fast connection set up and fast data transfer.

If a service request specifies LE security mode 2, the connection data signing procedure shall be used.

10.4.1 Connection Data Signing Procedure

A device shall generate a new Connection Signature Resolving Key CSRK for each peer device to which it sends signed data in a connection. CSRK is defined in [Vol. 3], Part H Section 2.4.2.2.

The data shall be formatted using the Signing Algorithm as defined in [Vol. 3], Part H Section 2.4.5 where m is the Data PDU to be signed, k is the CSRK and the SignCounter is the counter value. A Signature is composed of the counter value and the Message Authentication Code (MAC) generated by the Signing Algorithm. The counter value shall be incremented by one for each new Data PDU sent.

The format of signed data is shown in Figure 10.3.

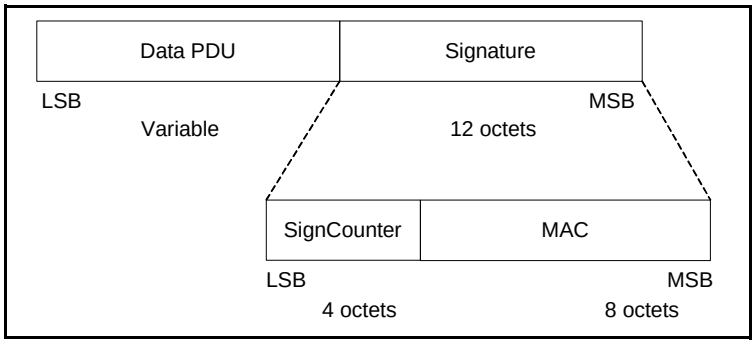


Figure 10.3: Generic format of signed data

10.4.2 Authenticate Signed Data Procedure

A device connected to a peer device sending signed data can authenticate the signed data by using the stored CSRK.

The signed data shall be authenticated by performing the Signing Algorithm where m is the Data PDU to be authenticated, k is the stored CSRK and the SignCounter is the received counter value. If the MAC computed by the Signing Algorithm does not match the received MAC, the verification fails and the Host shall ignore the received Data PDU

The receiving device shall protect against a replay attack by comparing the received SignCounter with previously received SignCounter from the same

peer device. If the SignCounter was previously used then the receiving device shall ignore the Data PDU and respond with error code “Signature Authentication Failed”, if permitted. The receiving device should protect from repeated failures by implementing a timer similar to the pairing timer as defined in [Vol. 3], Part H Section 2.3.6.

10.5 AUTHORIZATION PROCEDURE

A service may require authorization before allowing access. Authorization is a confirmation by the user to continue with the procedure. Authentication does not necessarily provide authorization. Authorization may be granted by user confirmation after successful authentication.

10.6 ENCRYPTION PROCEDURE

A Central may encrypt a connection using the Encryption Session Setup as defined in [Vol. 3], Part H Section 2.4.4 to provide integrity and confidentiality.

A Peripheral may encrypt a connection using the Slave Initiated Security Request as defined in [Vol. 3], Part H Section 2.4.6 to provide integrity and confidentiality.

10.7 PRIVACY FEATURE

The privacy feature provides a level of privacy which makes it more difficult for an attacker to track a device over a period of time. The requirements for a device to support the privacy feature are defined in Table 10.2.

Privacy Requirements	Ref.	Broadcaster	Observer	Peripheral	Central
Privacy feature	10.7	O	O	O	O
Non-resolvable private address generation procedure	10.8.2.1	E	C1	C2	C1
Resolvable private address generation procedure	10.8.2.2	C3	E	C3	E
Resolvable private address resolution procedure	10.8.2.3	E	C2	E	C3
C1: Mandatory if privacy feature and Active scanning are supported, else excluded C2: Optional if privacy is supported, else excluded C3: Mandatory if privacy feature is supported, else excluded					

Table 10.2: Requirements related to privacy feature

10.7.1 Privacy Feature in a Peripheral

Support of the privacy feature in a Peripheral depends on the presence and the value of two characteristics: Peripheral Privacy Flag characteristic as defined in Section 12.3 and the Reconnection Address characteristic as defined in Section 12.4. When the Peripheral Privacy Flag characteristic is set to 1, privacy is enabled (also known as privacy enabled Peripheral). Table 10.3 shows privacy support in the Peripheral. A privacy enabled Peripheral that does not expose the Reconnection Address characteristic may use a resolvable private address as the advertiser's device address when sending advertising events, depending on the mode that the Peripheral is operating in. A privacy enabled Peripheral that does expose the Reconnection Address characteristic may use the reconnection address as the advertiser's device address only if it has been set by the Central when sending advertising events depending on the mode that the Peripheral is operating in. When the Peripheral Privacy Flag characteristic is set to 0, privacy is disabled. When privacy is disabled the peripheral shall use a static or public address. When privacy is disabled, the permission of the Reconnection Address characteristic value shall be read-only, and shall not be considered as set by the Central.

Peripheral Privacy Flag characteristic present and set to 1 (privacy enabled)?	Reconnection Address characteristic present?	Privacy support in Peripheral
No	No	No privacy
Yes	No	Privacy in undirected connectable mode only
No	Yes	No privacy
Yes	Yes	Privacy in directed connectable mode and in undirected connectable mode

Table 10.3: privacy feature in a Peripheral

Note: A privacy enabled Peripheral may write one or more Central's static or public address in the White List of the Controller. A privacy enabled Peripheral may also write one or more reconnection addresses in the White List of the Controller. Static, public and reconnection addresses stored in the White List allow the Controller of the Peripheral to autonomously establish a connection with the Central that initiates a connection with the address that matches the address in the White List.

A Peripheral that supports the Peripheral Privacy Flag characteristic may allow a bonded Central to enable or disable the privacy feature. If the number of bonds on the Peripheral is less than or equal to one, the Peripheral Privacy Flag characteristic value may be writable. If a Peripheral has more than one bond, the permission of the Peripheral Privacy Flag characteristic value shall be read-only.

10.7.2 Privacy Feature in a Central

A device in the Central may support the privacy feature by use of the non-resolvable address or the reconnection address as the device address.

If the privacy feature is supported in a Central (also known as privacy enabled Central), the Central shall use a non-resolvable private address during active scanning.

When a Central connects to a privacy enabled Peripheral with the reconnection address characteristic, the Central shall write a new reconnection address in the reconnection address characteristic in the Peripheral. See general connection establishment procedure in Section 9.3.6. The privacy enabled Central may use the reconnection address when initiating a connection to a privacy enabled Peripheral.

When a Central connects with a Peripheral that supports the Peripheral Privacy Flag characteristic the Central may be allowed to enable or disable the privacy feature in the Peripheral.

Note: A privacy enabled Central can write the reconnection address in the White List.

10.8 RANDOM DEVICE ADDRESS

Random device addresses are defined in the Link Layer section in [Vol. 6], Part B Section 1.3.

For the purposes of this profile, the random device address may be of either of the following two sub-types:

- Static address
- Private address.

The term random device address refers to both static and private address types.

The transmission of a random device address is optional. A device shall accept the reception of a random device address from a remote device.

The private address may be of either of the following two sub-types:

- Non-resolvable private address
- Resolvable private address

10.8.1 Static Address

A static address is a 48-bit randomly generated address and shall meet the following requirements:

- The two most significant bits of the static address shall be equal to ‘1’
- All bits of the random part of the static address shall not be equal to ‘1’
- All bits of the random part of the static address shall not be equal to ‘0’

The format of a static address is shown in Figure 10.4.

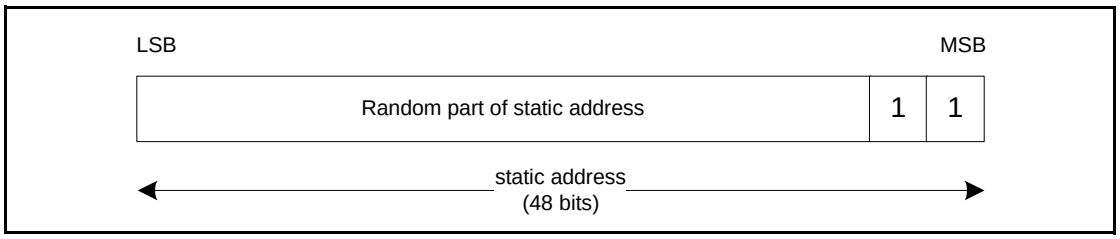


Figure 10.4: Format of static address

A device may choose to initialize its static address to a new value after each power cycle. A device shall not change its static address value once initialized until the device is power cycled.

Note: If the static address of a device is changed then the address stored in peer devices will not be valid and the ability to reconnect using the old address will be lost.

10.8.2 Private address

The private address may be of either of the following two sub-types:

- Non-resolvable private address
- Resolvable private address

The generation of a reconnection address shall use the non-resolvable private address generation procedure.

10.8.2.1 Non-Resolvable Private Address Generation Procedure

To generate a non-resolvable address or a reconnection address, the Host shall generate a 48-bit address with the following requirements:

- The two most significant bits of the address shall be equal to ‘0’
- All bits of the random part of the address shall not be equal to ‘0’
- All bits of the random part of the address shall not be equal to ‘1’
- The address shall not be equal to the static address
- The address shall not be equal to the public address

The format of a non-resolvable private address is shown in Table 10.5.

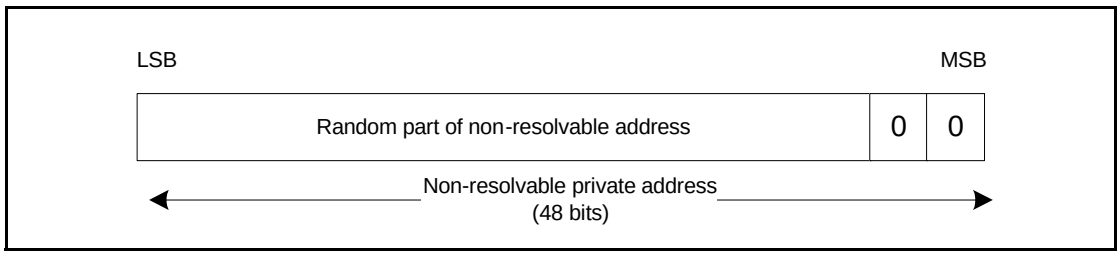


Figure 10.5: Format of non-resolvable private address

10.8.2.2 Resolvable Private Address Generation Procedure

To generate a resolvable private address the Host must have its Identity Resolving Key (IRK). The resolvable private address shall be generated with the IRK and a randomly generated 24-bit random number. The random number is known as *prand* and shall meet the following requirements:

- The two most significant bits of *prand* shall be equal to '0' and '1' as shown in Figure 10.6
- All bits of the random field of *prand* shall not be equal to '0'
- All bits of the random field of *prand* shall not be equal to '1'

The format of the resolvable private address is shown in Figure 10.6.

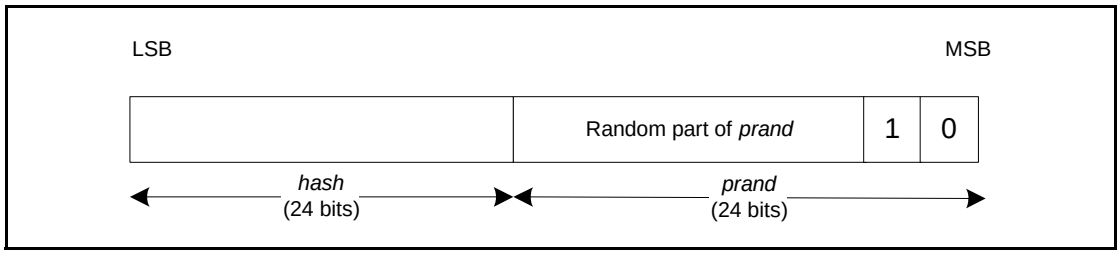


Figure 10.6: Format of resolvable private address

The *hash* is generated using the random address function *ah* defined in [Vol. 3], Part H Section 2.2.2 with the input parameter *k* set to the device's IRK and the input parameter *r* set to *prand*.

hash = *ah*(IRK, *prand*)

The *prand* and *hash* are concatenated to generate the random address (*randomAddress*) in the following manner:

randomAddress = *hash* || *prand*

The least significant octet of *hash* becomes the least significant octet of *randomAddress* and the most significant octet of *prand* becomes the most significant octet of *randomAddress*.

10.8.2.3 Resolvable Private Address Resolution Procedure

The Host can resolve a resolvable private address for all peer devices where the Host has the peer device's IRK using this procedure. If a resolvable private address is resolved, the Host can associate this address with the peer device.

The resolvable private address (*rpAddress*) is divided into a 24-bit random part (*prand*) and a 24-bit hash part (*hash*). The least significant octet of the *rpAddress* becomes the least significant octet of *hash* and the most significant octet of *rpAddress* becomes the most significant octet of *prand*.

A *localHash* value is then generated using the random address hash function *ah* defined in [Vol. 3], Part H Section 2.2.2 with the input parameter *k* set to IRK of the known device and the input parameter *r* set to the *prand* value extracted from the *rpAddress*.

localHash = *ah*(IRK, *prand*)

The *localHash* value is then compared with the *hash* value extracted from *rpAddress*. If the localHash value matches the extracted *hash* value then the identity of the peer device has been resolved.

If a Host has more than one stored IRK, the Host repeats the above procedure for each stored IRK to determine if the received resolvable private address is associated with a stored IRK, until either address resolution is successful for one of the IRKs or all have been tried.

11 ADVERTISING AND SCAN RESPONSE DATA FORMAT

The format of Advertising data and Scan Response data is shown in Figure 11.1. The data consists of a significant part and a non-significant part. The significant part contains a sequence of AD structures. Each AD structure shall have a Length field of one octet, which contains the Length value, and a Data field of Length octets. The first octet of the Data field contains the AD type field. The content of the remaining Length – 1 octet in the Data field depends on the value of the AD type field and is called the AD data. The non-significant part extends the Advertising and Scan Response data to 31 octets and shall contain all-zero octets.

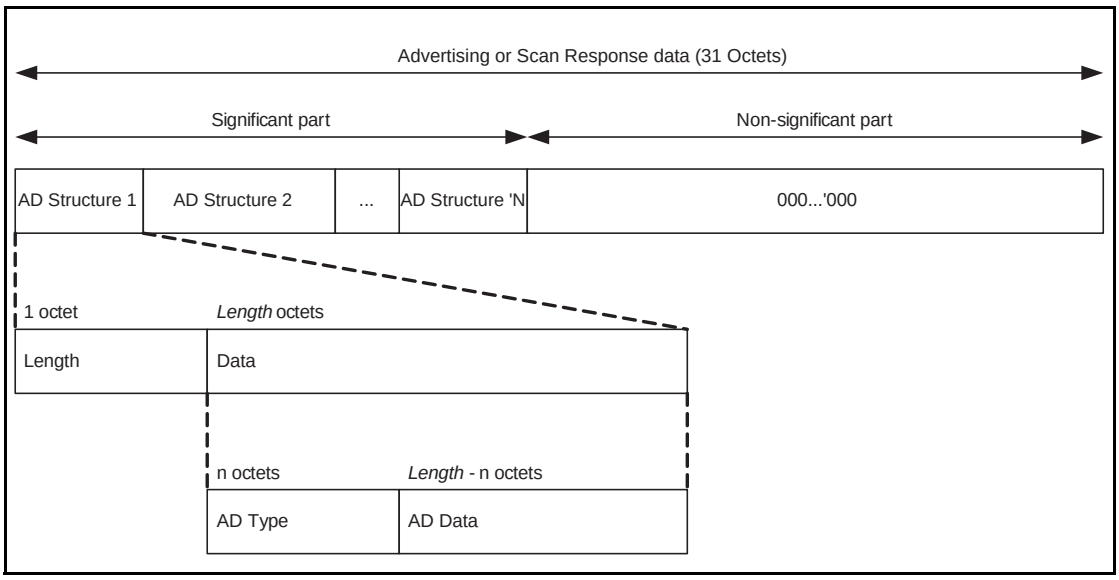


Figure 11.1: Advertising and Scan Response data format

If the Length field is set to zero, then the Data field has zero octets. This shall only occur to allow an early termination of the Advertising or Scan Response data.

Only the significant part of the Advertising or Scan Response data needs to be sent over the air.

The Advertising and Scan Response data is sent in advertising events. The Advertising Data is placed in the AdvData field of ADV_IND, ADV_NONCONN_IND, and ADV_SCAN_IND packets. The Scan Response data is sent in the ScanRspData field of SCAN_RSP packets.

11.1 AD TYPE DEFINITIONS

All AD types are listed in the Bluetooth Assigned Numbers document.

All numerical multi-byte entities and values shall use little-endian byte order.

11.1.1 Service UUIDs

A device may list the service UUIDs that it implements in its advertising data. Service UUIDs may be either 16-bit UUIDs or 128-bit UUIDs. The format of Service UUIDs for each type value is defined in Table 18.2.

11.1.2 Local Name

The Local Name AD type contains the device name, either complete or shortened as defined in Section 3.2.2. The AD type indicates if the name is complete or shortened. If the name is shortened, the complete device name can be read by reading the device name characteristic as defined in Section 12. The Advertising and Scan Response data shall not contain more than one instance of the Local Name AD type. A shortened name shall only contain contiguous characters from the beginning of the full name. For example, if the device name is ‘BT_Device_Name’ then the shortened name over BR/EDR could be ‘BT_Device’ while the shortened name on LE could be ‘BT_Dev’. The format of the data is defined in Table 18.3.

11.1.3 Flags

The Flags AD type contains several flag bits interpreted as boolean values.

Flags used over the LE physical channel are:

- Limited Discoverable Mode
- General Discoverable Mode
- BR/EDR Not Supported
- Simultaneous LE and BR/EDR to Same Device Capable (Controller)
- Simultaneous LE and BR/EDR to Same Device Capable (Host)

The Flags AD type shall not be included in the scan response data. The advertising data shall not contain more than one instance of the Flags AD type. The Flags AD type shall be included in the advertising data if any of the bits are non-zero. The Flags AD type may be omitted from the advertising data if all of the bits are zero. The format of Flags AD type is defined in Table 18.1.

11.1.4 Manufacturer Specific Data

The Manufacturer Specific AD type is used for manufacturer specific data. See Section 18.11.

11.1.5 TX Power Level

The TX Power Level AD type indicates the transmitted power level of the advertising packet. The format of TX Power Level AD is defined in Table 18.4.

11.1.6 Security Manager Out of Band (OOB)

An out of band mechanism is used by the Security Manager to communicate discovery information as well as other information related to the pairing process. The format of Security Manager Out of Band AD is defined in Section 18.7. This shall only be used over an out-of-band mechanism.

11.1.7 Security Manager TK Value

The Security Manager TK Value AD type is defined in Section 18.6. This shall only be used over an out-of-band mechanism.

11.1.8 Slave Connection Interval Range

The Slave Connection Interval Range AD type contains the Peripheral's preferred connection interval range, for all logical connections.

Note: The minimum value depends on the battery considerations of the Peripheral and the maximum connection interval depends on the buffers available on the Peripheral.

The Central should use the information from the Peripheral's Slave Connection Interval Range AD type when establishing a connection.

The Slave Connection Interval Range AD type is defined in Section 18.8**.**

11.1.9 Service Solicitation

One of the Service Solicitation AD types may be sent to invite other devices that expose one or more of the services specified in the Service Solicitation data to connect. The device should be in the undirected connectable mode and in one of the discoverable modes. This enables a Central providing one or more of these services to connect to this Peripheral, so that the Peripheral can use the services on the Central.

The Service Solicitation AD types are defined in Section 18.9.

11.1.10 Service Data

The Service Data AD type consists of a service UUID with the data associated with that service.

The Service Data AD type is defined in Section 18.10.

11.2 EXAMPLE ADVERTISING DATA

This is an example of advertising data with AD types:

Value	Notes
0x02	Length of this Data
0x01	AD type = Flags
0x01	LE Limited Discoverable Flag set
0x0A	Length of this Data
0x09	AD type = Complete local name
0x50	‘P’
0x65	‘e’
0x64	‘d’
0x6F	‘o’
0x6D	‘m’
0x65	‘e’
0x74	‘t’
0x65	‘e’
0x72	‘r’

12 GAP CHARACTERISTICS FOR LOW ENERGY

The service UUID shall be the UUID for «Generic Access Profile.»

The characteristics requirements for a device in each of the GAP roles are shown in Table 12.1. A device shall have only one instance of the GAP service. The GAP service is a GATT based service with the service UUID as defined in the Bluetooth Assigned Numbers document.

Characteristics	Ref.	Broadcaster	Observer	Peripheral	Central
Device Name	12.1	E	E	M	M
Appearance	12.2	E	E	M	M
Peripheral Privacy Flag	12.3	E	E	C1	E
Reconnection Address	12.4	E	E	C2	E
Peripheral Preferred Connection Parameters	12.5	E	E	O	E
C1: Mandatory if privacy feature is supported, else excluded C2: Optional if privacy feature and Peripheral privacy Flag characteristic are supported, else excluded					

Table 12.1: Requirements related to characteristics

A device that supports multiple GAP roles contains all the characteristics to meet the requirements for the supported roles. The device must continue to expose the characteristics when the device is operating in the role in which the characteristics are not valid.

12.1 DEVICE NAME CHARACTERISTIC

The Device Name characteristic shall contain the name of the device as an UTF-8 string as defined in Section 3.2.2. The Device Name characteristic value shall be readable without authentication or authorization. The Device Name characteristic value may be writable.

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x2A00 – UUID for «Device Name»	Device Name

Table 12.2: Device Name characteristic

The Device Name characteristic value shall be 0 to 248 octets in length. A device shall have only one instance of the Device Name characteristic.

12.2 APPEARANCE CHARACTERISTIC

The Appearance characteristic defines the representation of the external appearance of the device. This enables the discovering device to represent the device to the user using an icon, or a string, or similar. The Appearance characteristic value shall be readable without authentication or authorization. The Appearance characteristic value may be writable.

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x2A01 – UUID for «Appearance»	Appearance

Table 12.3: Appearance characteristic

The Appearance characteristic value shall be the enumerated value as defined by Bluetooth Assigned Numbers document. The Appearance characteristic value shall be 2 octets in length. A device shall have only one instance of the Appearance characteristic.

12.3 PERIPHERAL PRIVACY FLAG CHARACTERISTIC

The Peripheral Privacy Flag characteristic defines whether privacy is currently in use within this device. See Table 10.7.2.

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x2A02 – UUID for «Peripheral Privacy Flag»	Peripheral Privacy Flag

Table 12.4: Peripheral Privacy Flag characteristic

The Peripheral Privacy Flag characteristic value shall be 1 octet in length.

0 = privacy is disabled in this device

1 = privacy is enabled in this device

2 – 255 = Reserved

A device shall have only one instance of the Peripheral Privacy Flag characteristic.

12.4 RECONNECTION ADDRESS CHARACTERISTIC

The Reconnection Address characteristic defines the reconnection address. See Table 10.7.2. The reconnection address is a non-resolvable private address.

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x2A03 – UUID for «Reconnection Address»	Reconnection Address

Table 12.5: Reconnection Address characteristic

The Reconnection Address characteristic value shall be 6 octets in length. A device shall have only one instance of the Reconnection Address characteristic.

12.5 PERIPHERAL PREFERRED CONNECTION PARAMETERS CHARACTERISTIC

The Peripheral Preferred Connection Parameters (PPCP) characteristic contains the preferred connection parameters of the Peripheral.

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x0x2A04 – UUID for «Peripheral Preferred Connection Parameters»	Peripheral Preferred Connection Parameter

Table 12.6: Peripheral Preferred Connection Parameters characteristic

The Peripheral Preferred Connection Parameters characteristic value shall be 8 octets in length. A device shall have only one instance of the Peripheral Preferred Connection Parameters characteristic.

The preferred connection parameters structured data is defined as follows:

Name	Size (Octet)	Description
Minimum connection interval	2	Defines minimum value for the connection interval in the following manner: $connInterval_{min}$ = Conn_Interval_Min * 1.25 ms Conn_Interval_Min range: 0x0006 to 0x0C80 Value of 0xFFFF indicates no specific minimum. Values outside the range are reserved. (excluding 0xFFFF)

Table 12.7: Format of the preferred connection parameters structured data

Name	Size (Octet)	Description
Maximum connection interval	2	Defines maximum value for the connection interval in the following manner: $connInterval_{max}$ = Conn_Interval_Max * 1.25 ms Conn_Interval_Max range: 0x0006 to 0x0C80 Shall be equal to or greater than the Conn_Interval_Min. Value of 0xFFFF indicates no specific maximum. Values outside the range are reserved. (excluding 0xFFFF)
Slave latency	2	Defines the slave latency for the connection in number of connection events. Slave latency range: 0x0000 to 0x03E8 Values outside the range are reserved.
Connection Supervision timeout multiplier	2	Defines the connection supervisor timeout multiplier as a multiple of 10ms. Range: 0xFFFF indicates no specific value requested. Range: 0x000A to 0x0C80 Time = N * 10 msec Time Range: 100 msec to 32 seconds Values outside the range are reserved. (excluding 0xFFFF)

Table 12.7: Format of the preferred connection parameters structured data

13 BR/EDR/LE OPERATION MODES AND PROCEDURE

This section describes the requirements for devices that support the BR/EDR/LE device type. The device may support any LE GAP roles allowed by the Controller over the LE physical channel.

Feature	Ref.	Broadcaster	Observer	Peripheral	Central
BR/EDR/LE modes and procedures	13.1 and 13.2	O	O	O	O

Table 13.1: Requirements for the modes of a BR/EDR/LE device type

A BR/EDR/LE device type that supports the Broadcaster role shall meet the requirements for the Broadcaster role as defined in Section 9.

A BR/EDR/LE device type that supports the Observer role shall meet the requirements for the Observer role as defined in Section 9.

A BR/EDR/LE device type that supports the Peripheral role shall meet the requirements for the Peripheral role as defined in Section 13.1, Section 13.2 and Section 13.3.

A BR/EDR/LE device type that supports the Central role shall meet the requirements for the Central role as defined in Section 13.1, Section 13.2 and Section 13.3.

13.1 MODES

The requirements for a device supporting BR/EDR/LE device type are shown in Table 13.2.

Modes	Ref.	Peripheral	Central
Non-discoverable mode	13.1.1.1	C1	C1
Discoverable mode	13.1.1.2	C2	C2
Non-connectable mode	13.1.2.1	C3	C3
Connectable mode	13.1.2.2	M	M
Non-bondable mode	13.1.3.1	C4	C4
Bondable mode	13.1.3.2	C5	C5

Table 13.2: Requirements for the modes of a BR/EDR/LE device type

Modes	Ref.	Peripheral	Central
C1: Mandatory if the non-discoverable mode is supported over BR/EDR, else excluded			
C2: Mandatory if the discoverable mode is supported over BR/EDR, else excluded			
C3: Mandatory if the non-connectable mode is supported over BR/EDR, else excluded:			
C4: Mandatory if the non-bondable mode is supported over BR/EDR, else excluded			
C5: Mandatory if the bondable mode is supported over BR/EDR, else excluded			

Table 13.2: Requirements for the modes of a BR/EDR/LE device type

13.1.1 Discoverability

13.1.1.1 Non-Discoverable Mode

A device in non-discoverable mode shall follow the requirements for non-discoverable mode for BR/EDR as defined in Section 4.1.1 and it shall follow the requirements for non-discoverable mode for LE as defined in Section 9.2.2

13.1.1.2 Discoverable Mode

A device in general discoverable mode shall follow the requirements for general discoverable mode for BR/EDR as defined in Section 4.1.3 and it shall follow the requirements for general discoverable mode for LE as defined in Section 9.2.4, except it shall not send connectable advertising events.

A device in limited discoverable mode shall follow the requirements for limited discoverable mode for BR/EDR as defined in Section 4.1.2 and it shall follow the requirements for limited discoverable mode for LE as defined in Section 9.2.3, except it shall not send connectable advertising events.

A device that supports general discoverable and limited discoverable modes concurrently on different physical channels is not permitted in this version of the specification.

A device supporting both BR/EDR and LE physical links shall expose the capabilities of both physical links by performing the following steps:

a) The ‘LE Supported (Controller)’ and ‘LE Supported (Host)’ bits in the LMP features shall be set as defined in [Vol. 2], Part C Section 3.2.

b) The ‘BR/EDR Not Supported’ bit in the Flags AD type shall be set to ‘0’ as defined in Section 18.1.

c) The ‘Simultaneous LE and BR/EDR to Same Device Capable (Controller)’ and ‘Simultaneous LE and BR/EDR to Same Device

Capable (Host)' bits in the Flags AD type shall be set to '0' as defined in Section 18.1.

13.1.2 Connectability

13.1.2.1 Non-Connectable Mode

A device in the non-connectable mode shall follow the requirements for the non-connectable mode for BR/EDR as defined in Section 4.2.1 and it shall follow the requirements for the non-connectable mode for LE as defined in Section 9.3.2.

13.1.2.2 Connectable Mode

A device in the connectable mode shall follow the requirements for the connectable mode for BR/EDR as defined in Section 4.2.2, and it shall follow the requirements for the non-connectable mode for LE as defined in Section 9.3.2. The Host shall not set the "Simultaneous LE and BR/EDR to same device capable (Host)" LMP feature bit as defined in [Vol. 2], Part C Section 3.2.

13.1.3 Bondable Modes

13.1.3.1 Non-Bondable Mode

A device in the non-bondable mode shall follow the requirements for the non-bondable mode for BR/EDR as defined in Section 4.3.1 and it shall follow the requirements for the non-bondable mode for LE as defined in Section 9.4.2.

13.1.3.2 Bondable Mode

For a device that is operating in the LE Central role, a device in the bondable mode shall follow the requirements for the bondable mode for BR/EDR as defined inSection 4.3.2 and it shall follow the requirements for the bondable mode for LE as defined in Section 9.4.3.

For a device that is operating in the LE Peripheral role, a device in the bondable mode shall follow the requirements for the bondable mode for BR/EDR as defined in Section 4.3.2 and it shall follow the requirements for the non-bondable mode for LE as defined in Section 9.4.2.

13.2 IDLE MODE PROCEDURES

The requirements for a device supporting BR/EDR/LE device type are shown in Table 13.3.

Procedures	Ref.	Peripheral	Central
General discovery procedure	13.2.1	E	C1
Limited discovery procedure	13.2.2	E	C2
Device type discovery procedure	13.2.3.1	E	C3
Name discovery procedure	13.2.4	E	C4
Link establishment procedure	13.3	E	C5

C1: Mandatory if the general discovery procedure is supported over BR/EDR, else excluded

C2: Mandatory if the limited discovery procedure is supported over BR/EDR, else excluded

C3: Mandatory if either the general discovery or the limited discovery procedure is supported over BR/EDR, else excluded

C4: Mandatory if the name discovery procedure is supported over BR/EDR, else excluded

C5: Mandatory if the link establishment procedure is supported over BR/EDR, else excluded

Table 13.3: Requirements for the procedures of a BR/EDR/LE device type

13.2.1 General Discovery Procedure

In order to provide a good user experience for discovering remote devices, the general discovery procedure should interleave the general discovery procedure over BR/EDR as defined in Section 6.1 with the general discovery procedure over LE as defined in Section 9.2.6. A recommended method for interleaving is:

Step 1: perform the general discovery procedure over BR/EDR for (T_{GAP}(100) / 2) seconds

Step 2: perform the general discovery procedure over LE for (T_{GAP}(100) / 2) seconds

Step 3: repeat step 1 and 2 as necessary

13.2.2 Limited Discovery Procedure

In order to provide a good user experience for discovering remote devices, the limited discovery procedure should interleave the limited discovery procedure over BR/EDR as defined in Section 6.2 with the limited discovery procedure

over LE as defined in Section 9.2.5. A recommended method for interleaving is:

Step 1: perform the limited discovery procedure over BR/EDR for (T_{GAP}(100) / 2) seconds

Step 2: perform the limited discovery procedure over LE for (T_{GAP}(100) / 2) seconds

Step 3: repeat step 1 and 2 as necessary

13.2.3 Device Discovery Procedure

A discovering device shall be capable of discovering devices of each of the following device types:

- BR/EDR
- LE-only
- BR/EDR/LE

The device discovery procedure shall perform either the general discovery procedure as defined in Section 13.2.1 or the limited discovery procedure as defined in Section 13.2.2. The device type discovery procedure as defined in Section 13.2.3.1 shall be used to determine if the remote device supports both BR/EDR and LE physical links. Using the information from the device type discovery procedure a name discovery may be performed for each device discovered as defined in Section 13.2.4.

13.2.3.1 Device Type Discovery Procedure

The device type discovery procedure can be performed to discover the device type of the peer device.

If the procedure is performed on a BR/EDR physical link:

If the ‘LE Supported (Controller)’ bit in the LMP features is set and the ‘LE Supported (Host)’ bit in the LMP host features is set, then the device supports the BR/EDR/LE device type, else the device supports the BR/EDR device type.

If the procedure is performed on an LE physical channel:

If the ‘BR/EDR Not Supported’ bit in the Flags AD type is set then the device type is LE-only, else the device type is BR/EDR/LE. The device type shall not change while the device is bonded with other devices.

13.2.4 Name Discovery

When the remote device is an LE-only device type, the local device shall use the name discovery procedure as defined in Section 13.2.4. When the remote

device type is BR/EDR or BR/EDR/LE device type, the local device shall use the name discovery procedure as defined in Section 6.3.

Note: The device type of the remote device can be discovered using the device type discovery procedure as defined in Section 13.2.3.1.

13.3 ESTABLISHMENT PROCEDURES

13.3.1 Link Establishment

When establishing a link to a remote device that supports the BR/EDR/LE device type, the link establishment procedure as defined in Section 7.1 shall be used, and shall not use the connection establishment procedures as defined in Section 9.3.

When establishing a link to a remote device that supports the BR/EDR device type, the link establishment procedure as defined in Section 7.1 shall be used.

When establishing a link to a remote device of the LE-only device type, the connection establishment procedures as defined in Section 9.3 shall be used.

Note: The device type of the remote device can be discovered using the device type discovery procedure as defined in Section 13.2.3.1.

13.4 SDP INTEROPERABILITY REQUIREMENTS

A BR/EDR/LE device that supports GATT over BR/EDR shall publish the following SDP record. The GAP Start Handle shall be set to the attribute handle of the «Generic Access Profile» service declaration. The GAP End Handle shall be set to the attribute handle of the last attribute within the «Generic Access Profile» service definition group.

Item	Definition	Type	Value	Status
Service Class ID List				M
Service Class #0		UUID	Generic Access Profile	M
Protocol Descriptor List				M
Protocol #0		UUID	L2CAP	M
Parameter #0 for Protocol #0	PSM	Uint16	PSM = ATT	M
Protocol #1		UUID	ATT	M

Table 13.4: SDP Record for the Generic Access Profile

Item	Definition	Type	Value	Status
Parameter #0 for Protocol #1	GAP Start Handle	Uint16		M
Parameter #1 for Protocol #1	GAP End Handle	Uint16		M
BrowseGrou-pList			PublicBrowseGroup	M

Table 13.4: SDP Record for the Generic Access Profile

14 BR/EDR/LE SECURITY ASPECTS

The requirements for a device supporting BR/EDR/LE device type are shown in Table 14.1.

Security aspects	Ref.	Peripheral	Central
Security aspects	14	M	M

Table 14.1: Requirements for the security aspects of a BR/EDR/LE device type

If the remote device supports the BR/EDR/LE device type or the BR/EDR device type the security procedures as defined in Section 5 shall be used.

If the remote device is of the LE-only device type the security procedures as defined in Section 10 shall be used.

Note: The device type of the remote device can be discovered using the device type discovery procedure as defined in Table 13.2.3.1.

15 DEFINITIONS

In the following, terms written with capital letters refer to states.

15.1 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.

15.2 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.

15.3 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 requested.

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.

15.4 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.

15.5 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 bondable mode) after a successful pairing.

16 APPENDIX A (NORMATIVE): TIMERS AND CONSTANTS

The following timers are required by this profile.

Timer name	Value	Description	Requirement or Recommendation
$T_{GAP}(100)$	10.24 s	Time span that a Bluetooth device performs device discovery.	Recommended value
$T_{GAP}(101)$	10.625 ms	A discoverable Bluetooth device enters INQUIRY_SCAN for at least TGAP(101) every TGAP(102).	Required value
$T_{GAP}(102)$	2.56 s	Maximum time between repeated INQUIRY_SCAN enterings.	Recommended value
$T_{GAP}(103)$	30.72 s	Minimum time span that a device is in discoverable mode.	Required value
$T_{GAP}(104)$	1 min.	Maximum time span that a device is in limited discoverable mode.	Recommended value
$T_{GAP}(105)$	100ms	Maximum time between INQUIRY_SCAN enterings	Recommended value
$T_{GAP}(106)$	100ms	Maximum time between PAGE_SCAN enterings	Recommended value
$T_{GAP}(107)$	1.28s	Maximum time between PAGE_SCAN enterings (R1 page scan)	Recommended value
$T_{GAP}(108)$	2.56s	Maximum time between PAGE_SCAN enterings (R2 page scan)	Recommended value
$T_{GAP}(lim\_adv\_timeout)$	30.72 s	Maximum time to remain advertising when in the limited discoverable mode	Required value
$T_{GAP}(gen\_disc\_scan\_min)$	10.24 s	Minimum time to perform scanning when performing the general discovery procedure	Recommended value

Table 16.1: Defined GAP timers

Timer name	Value	Description	Requirement or Recommendation
T_{GAP}(gen_disc_scan_int)	11.25 ms	Scan interval used in the general discovery procedure	Recommended value
T_{GAP}(gen_disc_scan_wind)	11.25 ms	Scan window used in the general discovery procedure	Recommended value
T_{GAP}(lim_disc_scan_min)	10.24 s	Minimum time to perform scanning when performing the limited discovery procedure	Recommended value
T_{GAP}(lim_disc_scan_int)	11.25 ms	Scan interval used in the limited discovery procedure	Recommended value
T_{GAP}(lim_disc_scan_wind)	11.25 ms	Scan window used in the limited discovery procedure	Recommended value
T_{GAP}(conn_param_timeout)	30 s	Connection parameter update notification timer when performing the connection parameter update procedure	Recommended value
T_{GAP}(conn_est_int_min)	500 ms	Minimum connection interval in any connection establishment procedure	Recommended value
T_{GAP}(conn_est_int_max)	1.28 s	Maximum connection interval in any connection establishment procedure	Recommended value
T_{GAP}(conn_est_latency)	0	Connection slave latency in any connection establishment procedure	Recommended value
T_{GAP}(private_addr_int)	15 mins	Minimum time interval between private address change	Recommended value
T_{GAP}(lim_disc_adv_int_min)	250 ms	Minimum advertising interval when in the limited discoverable mode	Recommended value
T_{GAP}(lim_disc_adv_int_max)	500 ms	Maximum advertising interval when in the limited discoverable mode	Recommended value
T_{GAP}(gen_disc_adv_int_min)	1.28 s	Minimum advertising interval when in the general discoverable mode	Recommended value

Table 16.1: Defined GAP timers

Timer name	Value	Description	Requirement or Recommendation
T_{GAP}(gen_disc_adv_int_max)	2.56 s	Maximum advertising interval when in the general discoverable mode	Recommended value
T_{GAP}(dir_conn_adv_int_min)	250 ms	Minimum advertising interval when in the directed connectable mode	Recommended value
T_{GAP}(dir_conn_adv_int_max)	500 ms	Maximum advertising interval when in the directed connectable mode	Recommended value
T_{GAP}(undir_conn_adv_int_min)	1.28 s	Minimum advertising interval when in the undirected connectable mode	Recommended value
T_{GAP}(undir_conn_adv_int_max)	2.56 s	Maximum advertising interval when in the undirected connectable mode	Recommended value
T_{GAP}(auto_conn_est_scan_int)	1.28 s	Scan interval used in the auto connection establishment procedure	Recommended value
T_{GAP}(auto_conn_est_scan_wind)	11.25 ms	Scan window used in the auto connection establishment procedure	Recommended value
T_{GAP}(gen_conn_est_scan_int)	1.28 s	Scan interval used in the general connection establishment procedure	Recommended value
T_{GAP}(gen_conn_est_scan_wind)	11.25 ms	Scan window used in the general connection establishment procedure	Recommended value
T_{GAP}(sel_conn_est_scan_int)	1.28 s	Scan interval used in the selective connection establishment procedure	Recommended value
T_{GAP}(sel_conn_est_scan_wind)	11.25 ms	Scan window used in the selective connection establishment procedure	Recommended value
T_{GAP}(conn_param_timeout)	30 s	Timer used in connection parameter update procedure	Recommended value

Table 16.1: Defined GAP timers

17 APPENDIX B (INFORMATIVE): INFORMATION FLOWS OF RELATED PROCEDURES

17.1 LMP – AUTHENTICATION

The specification of authentication on link level is found in [2].

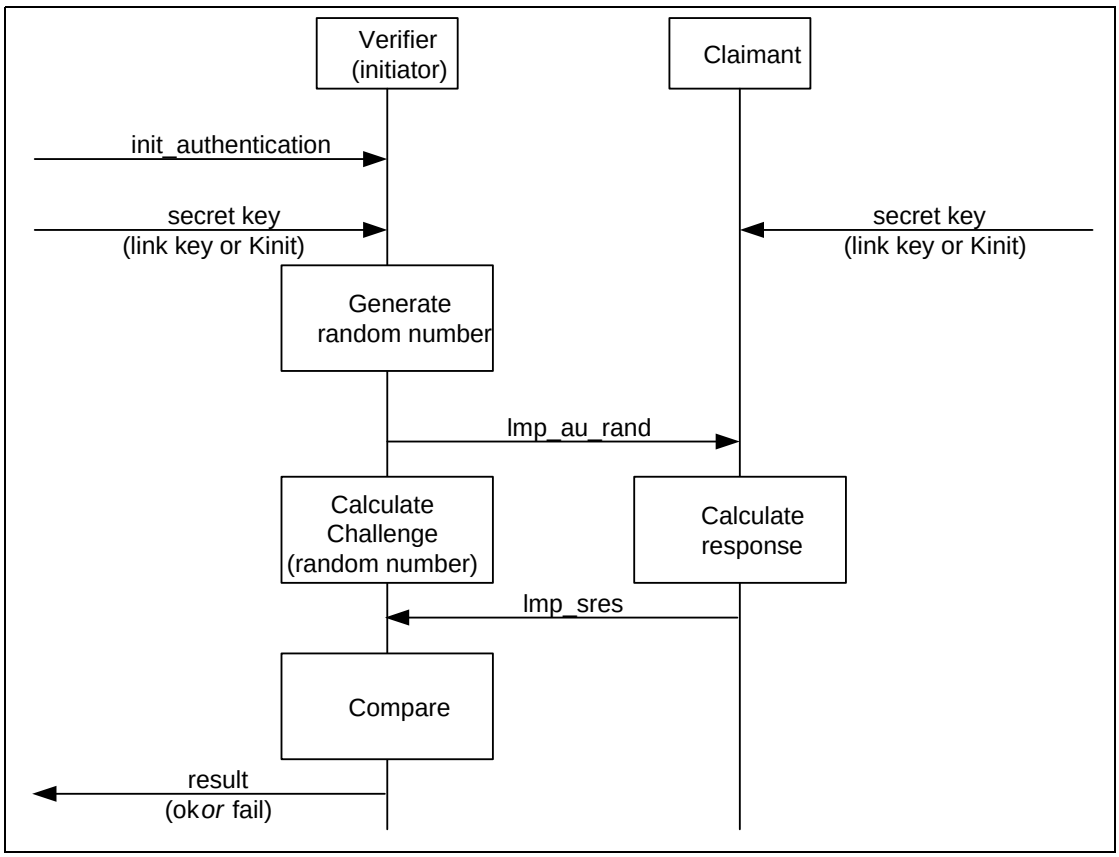


Figure 17.1: LMP-authentication as defined by [2].

The secret key used here is an already exchanged link key.

17.2 LMP – PAIRING

The specification of pairing on link level is found in [2].

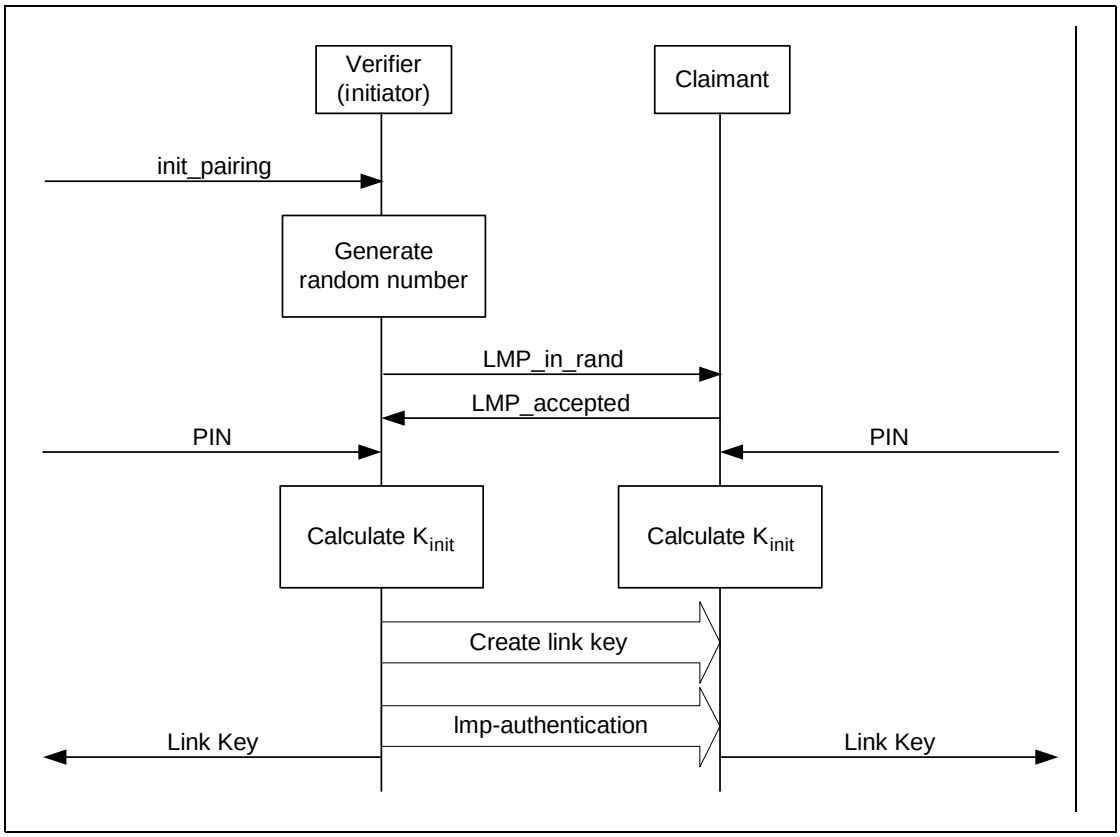


Figure 17.2: LMP-pairing as defined in [2].

The PIN used here is PINBB.

The create link key procedure is described in [Vol. 2, Part C] Section 4.2.2.4 on page 251 and [Vol. 2, Part H] Section 3.2 on page 1065. 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.

17.3 SERVICE DISCOVERY

The Service Discovery Protocol [5] 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 higher layer specifications. This is just an example.

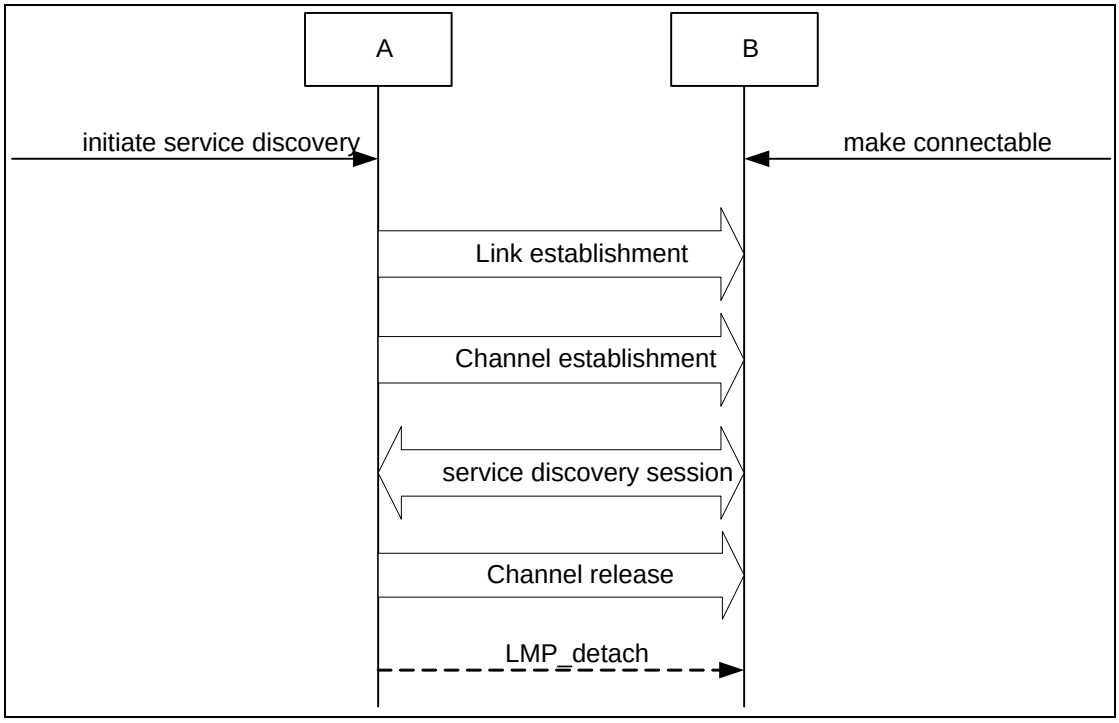


Figure 17.3: Service discovery procedure.

17.4 GENERATING A RESOLVABLE PRIVATE ADDRESS

Generating a resolvable private address is described in Section 10.8.2.2.

17.5 RESOLVING A RESOLVABLE PRIVATE ADDRESS

Resolving a resolvable private address is described in Section 10.8.2.3.

18 APPENDIX C (NORMATIVE): EIR AND AD FORMATS

This section defines the data format used in the EIR data field and in the AD format.

18.1 FLAGS

The Flags field may be zero or more octets long. This allows the Flags field to be extended while using the minimum number of octets within the data packet. All octets that are 0x00 are not transmitted as long as all other octets after that octet are also 0x00.

Value	Description	Bit	Information
0x01	Flags	0	LE Limited Discoverable Mode
		1	LE General Discoverable Mode
		2	BR/EDR Not Supported (i.e. bit 37 of LMP Extended Feature bits Page 0)
		3	Simultaneous LE and BR/EDR to Same Device Capable (Controller) (i.e. bit 49 of LMP Extended Feature bits Page 0)
		4	Simultaneous LE and BR/EDR to Same Device Capable (Host) (i.e. bit 66 of LMP Extended Feature bits Page 1)
		5..7	Reserved

Table 18.1: Flags

18.2 SERVICE

Value	Description	Information
0x02	16-bit Service UUIDs	More 16-bit UUIDs available
0x03	16-bit Service UUIDs	Complete list of 16-bit UUIDs available
0x04	32-bit Service UUIDs	More 32-bit UUIDs available
0x05	32-bit Service UUIDs	Complete list of 32-bit UUIDs available
0x06	128-bit Service UUIDs	More 128-bit UUIDs available
0x07	128-bit Service UUIDs	Complete list of 128-bit UUIDs available

Table 18.2: Service UUIDs

18.3 LOCAL NAME

Value	Description	Information
0x08	Local Name	Shortened local name
0x09	Local Name	Complete local name

Table 18.3: Local Name

18.4 TX POWER LEVEL

Value	Description	Information
0x0A	TX Power Level (1 byte)	0xXX:-127 to +127dBm Note: when the TX Power Level tag is not present, the TX power level of the packet is unknown.

Table 18.4: TX Power Level

18.5 SIMPLE PAIRING OPTIONAL OOB TAGS

Value	Description	Information
0x0D	Class of device (3 octets)	Format defined in Assigned Numbers
0x0E	Simple Pairing Hash C (16 octets)	Format defined in [Vol. 2], Part H Section 7.2.2
0x0F	Simple Pairing Randomizer R (16 octets)	Format defined in[Vol. 2], Part H Section 7.2.2

Table 18.5: Optional OOB Tags

18.6 SECURITY MANAGER TK VALUE

Value	Description	Information
0x10	TK Value	Value as used in pairing over LE Physical channel. Format defined in [Vol. 3], Part H Section 2.3

Table 18.6: Security Manager TK Value

18.7 SECURITY MANAGER OOB FLAGS

Value	Description	Bit	Information
0x11	Flag (1 octet)	0	OOB Flags Field (0 = OOB data not present, 1 = OOB data present)
		1	LE supported (Host) (i.e. bit 65 of LMP Extended Feature bits Page 1
		2	Simultaneous LE and BR/EDR to Same Device Capable (Host) (i.e. bit 66 of LMP Extended Feature bits Page 1)
		3	Address type (0 = Public Address, 1 = Random Address)
		4.7	Reserved

Table 18.7: Security Manager OOB Flags

18.8 SLAVE CONNECTION INTERVAL RANGE

Value	Description	Information
0x12	Slave Connection Interval Range	The first 2 octets defines the minimum value for the connection interval in the following manner: $connInterval_{min}$ = Conn_Interval_Min * 1.25 ms Conn_Interval_Min range: 0x0006 to 0x0C80 Value of 0xFFFF indicates no specific minimum. Values outside the range are reserved. (excluding 0xFFFF) The second 2 octets defines the maximum value for the connection interval in the following manner: $connInterval_{max}$ = Conn_Interval_Max * 1.25 ms Conn_Interval_Max range: 0x0006 to 0x0C80 Conn_Interval_Max shall be equal to or greater than the Conn_Interval_Min. Value of 0xFFFF indicates no specific maximum. Values outside the range are reserved (excluding 0xFFFF)

Table 18.8: Slave Connection Interval Range

18.9 SERVICE SOLICITATION

Value	Description	Information
0x14	Service UUIDs	List of 16 bit Service UUIDs
0x15	Service UUIDs	List of 128 bit Service UUID

Table 18.9: Service Solicitation

18.10 SERVICE DATA

Value	Description	Information
0x16	Service Data (2 or more octets)	The first 2 octets contain the 16 bit Service UUID followed by additional service data

Table 18.10: Service Data

18.11 MANUFACTURER SPECIFIC DATA

Value	Description	Information
0xFF	Manufacturer Specific Data (2 or more octets)	The first 2 octets contain the Company Identifier Code followed by additional manufacturer specific data

Table 18.11: Manufacturer Specific Data

19 REFERENCES

[1] Baseband Specification

[2] Link Manager Protocol

[3] RFCOMM

[4] Telephony Control Specification

[5] Service Discovery Protocol

[6] Security Architecture (white paper)

[7] Bluetooth Assigned Numbers

TEST SUPPORT

Specification Volume 6

SPECIFICATION OF THE *BLUETOOTH* SYSTEM

Experience More



Core System Package [Low Energy Controller volume]

Covered Core Package version:
4.0

Bluetooth®

ETN1009
285/349

LINK LAYER SPECIFICATION

This part of the specification describes the Bluetooth low energy Link Layer.

CONTENTS

1 General Description 32
1.1 Link Layer States 32
1.1.1 State and Role Combination Restrictions 33
1.2 Bit Ordering 34
1.3 Device Address 35
1.4 Physical Channel 36
1.4.1 Advertising and Data Channel Indexes 37
2 Air Interface Packets 38
2.1 Packet Format 38
2.1.1 Preamble 38
2.1.2 Access Address 38
2.1.3 PDU 39
2.1.4 CRC 39
2.2 Reserved for Future Use (RFU) 39
2.3 Advertising Channel PDU 39
2.3.1 Advertising PDUs 41
2.3.1.1 ADV_IND 41
2.3.1.2 ADV_DIRECT_IND 42
2.3.1.3 ADV_NONCONN_IND 42
2.3.1.4 ADV_SCAN_IND 43
2.3.2 Scanning PDUs 43
2.3.2.1 SCAN_REQ 43
2.3.2.2 SCAN_RSP 44
2.3.3 Initiating PDUs 44
2.3.3.1 CONNECT_REQ 44
2.4 Data Channel PDU 46
2.4.1 LL Data PDU 47
2.4.2 LL Control PDU 48
2.4.2.1 LL_CONNECTION_UPDATE_REQ 50
2.4.2.2 LL_CHANNEL_MAP_REQ 50
2.4.2.3 LL_TERMINATE_IND 51
2.4.2.4 LL_ENC_REQ 51
2.4.2.5 LL_ENC_RSP 51
2.4.2.6 LL_START_ENC_REQ 52
2.4.2.7 LL_START_ENC_RSP 52
2.4.2.8 LL_UNKNOWN_RSP 52
2.4.2.9 LL_FEATURE_REQ 52
2.4.2.10 LL_FEATURE_RSP 52
2.4.2.11 LL_PAUSE_ENC_REQ 53
2.4.2.12 LL_PAUSE_ENC_RSP 53

2.4.2.13 LL_VERSION_IND 53
2.4.2.14 LL_REJECT_IND 53

3 Bit Stream Processing 54
3.1 Error Checking 54
3.1.1 CRC Generation 54
3.2 Data Whitening 55

4 Air Interface Protocol 57
4.1 Inter Frame Space 57
4.2 Timing Requirements 57
4.2.1 Active Clock Accuracy 57
4.2.2 Sleep Clock Accuracy 57
4.3 Link Layer Device Filtering 57
4.3.1 White List 58
4.3.2 Advertising Filter Policy 58
4.3.3 Scanner Filter Policy 59
4.3.4 Initiator Filter Policy 59
4.4 Non-Connected States 59
4.4.1 Standby State 59
4.4.2 Advertising State 59
4.4.2.1 Advertising Channel Selection 61
4.4.2.2 Advertising Interval 61
4.4.2.3 Connectable Undirected Event Type 61
4.4.2.4 Connectable Directed Event Type 63
4.4.2.5 Scannable Undirected Event Type 64
4.4.2.6 Non-connectable Undirected Event Type 66
4.4.3 Scanning State 67
4.4.3.1 Passive Scanning 68
4.4.3.2 Active Scanning 68
4.4.4 Initiating State 69
4.5 Connection State 69
4.5.1 Connection Events 70
4.5.2 Supervision Timeout 71
4.5.3 Connection Event Transmit Window 72
4.5.4 Connection Setup – Master Role 72
4.5.5 Connection Setup – Slave Role 73
4.5.6 Closing Connection Events 74
4.5.7 Window Widening 75
4.5.8 Data Channel Index Selection 76
4.5.8.1 Channel Classification 76
4.5.8.2 Channel Selection 76

4.5.9 Acknowledgement and Flow Control 77
4.5.9.1 Flow Control 79
4.6 Feature Support 79
4.6.1 LE Encryption 79
5 Link Layer Control 80
5.1 Link Layer Control Procedures 80
5.1.1 Connection Update Procedure 80
5.1.2 Channel Map Update Procedure 82
5.1.3 Encryption Procedure 84
5.1.3.1 Encryption Start Procedure 84
5.1.3.2 Encryption Pause Procedure 86
5.1.4 Feature Exchange Procedure 87
5.1.5 Version Exchange 88
5.1.6 Termination Procedure 88
5.2 Procedure Response Timeout 89

1 GENERAL DESCRIPTION

1.1 LINK LAYER STATES

The operation of the Link Layer can be described in terms of a state machine with the following five states:

- Standby State
- Advertising State
- Scanning State
- Initiating State
- Connection State

The Link Layer state machine allows only one state to be active at a time. The Link Layer shall have at least one Link Layer state machine that supports one of Advertising State or Scanning State. The Link Layer may have multiple instances of the Link Layer state machine. Certain combinations of states and roles within multiple state machines in the Link Layer are prohibited (see Section 1.1.1).

The Link Layer in the Standby State does not transmit or receive any packets. The Standby State can be entered from any other state.

The Link Layer in the Advertising State will be transmitting advertising channel packets and possibly listening to and responding to responses triggered by these advertising channel packets. A device in the Advertising State is known as an advertiser. The Advertising State can be entered from the Standby State.

The Link Layer in the Scanning State will be listening for advertising channel packets from devices that are advertising. A device in the Scanning State is known as a scanner. The Scanning State can be entered from the Standby State.

The Link Layer in the Initiating State will be listening for advertising channel packets from a specific device(s) and responding to these packets to initiate a connection with another device. A device in the Initiating State is known as an initiator. The Initiating State can be entered from the Standby State.

The Connection State can be entered either from the Initiating State or the Advertising State. A device in the Connection State is known as being in a connection.

Within the Connection State, two roles are defined:

- Master Role
- Slave Role

When entered from the Initiating State, the Connection State shall be in the Master Role. When entered from the Advertising State, the Connection State shall be in the Slave Role.

The Link Layer in the Master Role will communicate with a device in the Slave Role and defines the timings of transmissions.

The Link Layer in the Slave Role will communicate with a single device in the Master Role.

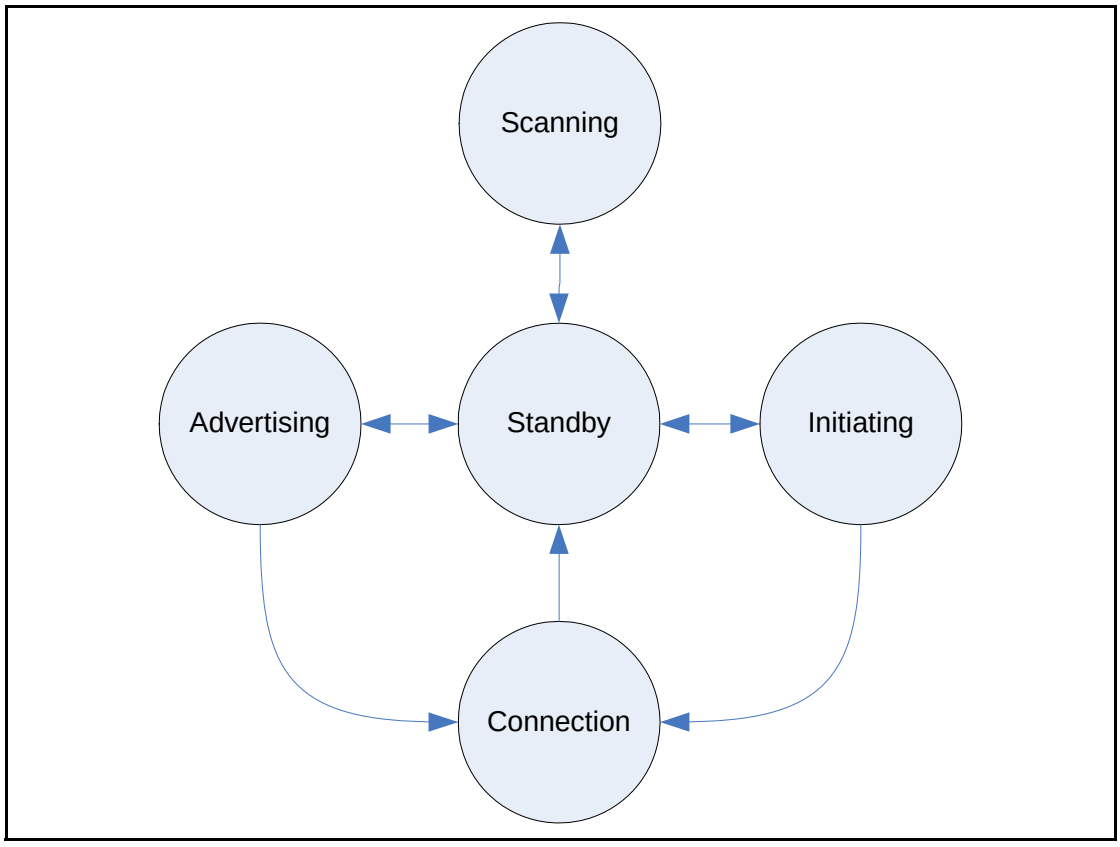


Figure 1.1: State diagram of the Link Layer state machine

1.1.1 State and Role Combination Restrictions

The Link Layer may optionally support multiple state machines. If it does support multiple state machines, the following state and role restrictions shall apply.

- The Link Layer in the Connection State shall not operate in the Master Role and Slave Role at the same time.
- The Link Layer in the Connection State operating in the Slave Role shall have only one connection.
- The Link Layer in the Connection State operating in the Master Role may have multiple connections.

- The Link Layer shall not operate in the Initiating State if the Link Layer is already operating in the Connection State in the Slave Role.
- If the Link Layer is already operating in the Connection State or Initiating State, the Link Layer shall not operate in the Advertising State with a type of advertising that could result in the Link Layer entering the Connection State in the Slave Role.

All other combinations of states may be supported. A Link Layer implementation is not required to support all the possible state combinations that are allowed by this specification.

The table below specifies the allowable and prohibited Link Layer state machine state combinations.

Multiple State Machine State and Roles		**Advertising**	**Scanning**	**Initiating**	**Connection**	
					Master Role	**Slave Role**
Advertising		Prohibited	Allowed	Allowed*	Allowed*	Allowed*
Scanning		Allowed	Prohibited	Allowed	Allowed	Allowed
Initiating		Allowed*	Allowed	Prohibited	Allowed	Prohibited
Connection	**Master Role**	Allowed*	Allowed	Allowed	Allowed	Prohibited
	Slave Role	Allowed*	Allowed	Prohibited	Prohibited	Prohibited

Table 1.1: Allowable and prohibited Link Layer state machine state combinations

* Only advertising packets that shall not result in the Link Layer entering Connection State in the Slave Role allowed.

1.2 BIT ORDERING

The bit ordering when defining fields within the packet or Protocol Data Unit (PDU) in the Link Layer specification follows the Little Endian format. 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

Furthermore, data fields defined in the Link Layer, such as the PDU header fields, shall be transmitted with the LSB first. For instance, a 3-bit parameter *X=3* is sent as:

$b_0b_1b_2 = 110$

Over the air, 1 is sent first, 1 is sent next, and 0 is sent last. This is shown as 110 in the specification.

Binary field values specified in this specification that follow the format 10101010b (e.g., pre-amble in Section 2.1.1 or advertising channel Access Address in Section 2.1.2) are written with the MSB to the left.

Multi-octet fields, with the exception of the Cyclic Redundancy Check (CRC) and the Message Integrity Check (MIC), shall be transmitted with the least significant octet first. Each octet within multi-octet fields, with the exception of the CRC (see Section 3.1.1), shall be transmitted in LSB first order. For example, the 48-bit addresses in the advertising channel PDUs shall be transmitted with the least significant octet first, followed by the remainder of the five octets in increasing order.

Multi-octet field values specified in this specification (e.g. the CRC initial value in Section 2.3.3.1) are written with the most significant octet to the left; for example in 0x112233445566, the octet 0x11 is the most significant octet.

1.3 DEVICE ADDRESS

Devices are identified using a device address. Device addresses may be either a public device address or a random device address. A public device address and a random device address are both 48 bits in length.

A device shall contain at least one type of device address and may contain both.

The public device address shall be created in accordance with section 9.2 ("48-bit universal LAN MAC addresses") of the IEEE 802-2001 standard (http://standards.ieee.org/getieee802/download/802-2001.pdf) and using a valid Organizationally Unique Identifier (OUI) obtained from the IEEE Registration Authority (see http://standards.ieee.org/regauth/oui/forms/ and sections 9 and 9.1 of the IEEE 802-2001 specification).

The public device address is divided into the following two fields:

- company_assigned field is contained in the 24 least significant bits
- company_id field is contained in the 24 most significant bits



Figure 1.2: Format of public device address

The random device address is divided into the following two fields:

- hash field is contained in the 24 least significant bits, as defined in [Vol. 3] Part C, Section 10.8.2.3.
- random field is contained in the 24 most significant bits, as defined in [Vol. 3] Part C, Section 10.8.2.2.

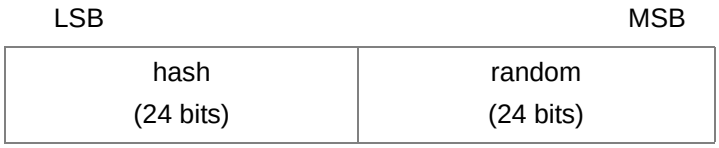


Figure 1.3: Format of random device address

1.4 PHYSICAL CHANNEL

As specified in Part A, Section 2, 40 RF Channels are defined in the 2.4GHz ISM band. These RF Channels are allocated into two LE physical channels: advertising and data. The advertising physical channel uses three RF channels for discovering devices, initiating a connection and broadcasting data. The data physical channel uses up to 37 (see Section 4.5.8) RF channels for communication between connected devices. Each of these RF Channels is allocated a unique channel index (see Section 1.4.1).

Two devices that wish to communicate use a shared physical channel. To achieve this, their transceivers must be tuned to the same RF Channel at the same time.

Given that the number of RF Channels is limited, and that many Bluetooth devices may be operating independently within the same spatial and temporal area, there is a strong likelihood of two independent Bluetooth devices having their transceivers tuned to the same RF Channel, resulting in a physical channel collision. To mitigate the unwanted effects of this collision, each transmission on a physical channel starts with an Access Address that is used as a correlation code by devices tuned to the physical channel. This Access Address is a property of the physical channel. The Access Address is present at the start of every transmitted packet.

The Link Layer uses one physical channel at a given time.

Whenever the Link Layer is synchronized to the timing, frequency and Access Address of a physical channel it is said to be 'connected' to this channel (whether or not it is actively involved in communications over the channel).

1.4.1 Advertising and Data Channel Indexes

Table 1.2 shows the mapping from RF Channel to Data Channel Index and Advertising Channel Index. It also shows the allocation of channel type to each RF Channel.

RF Channel	RF Center Frequency	Channel Type	Data Channel Index	Advertising Channel Index
0	2402 MHz	Advertising channel		37
1	2404 MHz	Data channel	0	
2	2406 MHz	Data channel	1	
...	…	Data channels	…	
11	2424 MHz	Data channel	10	
12	2426 MHz	Advertising channel		38
13	2428 MHz	Data channel	11	
14	2430 MHz	Data channel	12	
...	…	Data channels	…	
38	2478 MHz	Data channel	36	
39	2480 MHz	Advertising channel		39

Table 1.2: Mapping of RF Channel to Data Channel Index and Advertising Channel Index

2 AIR INTERFACE PACKETS

LE devices shall use the packets as defined in the following sections.

2.1 PACKET FORMAT

The Link Layer has only one packet format used for both advertising channel packets and data channel packets.

The packet format is shown in Figure 2.1. Each packet consists of four fields: the preamble, the Access Address, the PDU, and the CRC.

LSB MSB

Preamble (1 octet)	Access Address (4 octets)	PDU (2 to 39 octets)	CRC (3 octets)

Figure 2.1: Link Layer packet format

The preamble is 1 octet and the Access Address is 4 octets. The PDU range is from 2 to a maximum of 39 octets. The CRC is 3 octets.

The Preamble is transmitted first, followed by the Access Address, followed by the PDU followed by the CRC.

The shortest packet is 80 bits in length. The longest packet is 376 bits in length.

2.1.1 Preamble

All Link Layer packets have an eight bit preamble. The preamble is used in the receiver to perform frequency synchronization, symbol timing estimation, and Automatic Gain Control (AGC) training.

Advertising channel packets shall have 10101010b as the preamble.

The data channel packet preamble is either 10101010b or 01010101b, depending on the LSB of the Access Address. If the LSB of the Access Address is 1, the preamble shall be 01010101b, otherwise the preamble shall be 10101010b.

2.1.2 Access Address

The Access Address for all advertising channel packets shall be 10001110100010011011111011010110b (0x8E89BED6).

The Access Address in data channel packets shall be different for each Link Layer connection between any two devices with certain restrictions as defined below. The Access Address shall be a random 32-bit value, generated by the device in the Initiating State and sent in a connection request as defined in Section 2.3.3.1. The initiator shall ensure that the Access Address meets the following requirements:

- It shall have no more than six consecutive zeros or ones.
- It shall not be the advertising channel packets' Access Address.
- It shall not be a sequence that differs from the advertising channel packets' Access Address by only one bit.
- It shall not have all four octets equal.
- It shall have no more than 24 transitions.
- It shall have a minimum of two transitions in the most significant six bits.

2.1.3 PDU

The preamble and Access Address are followed by a PDU.

When a packet is transmitted in an advertising physical channel, the PDU shall be the Advertising Channel PDU as defined in Section 2.3. When a packet is transmitted in a data physical channel, the PDU shall be the Data Channel PDU as defined in Section 2.4.

2.1.4 CRC

At the end of every Link Layer packet there is a 24-bit CRC. It shall be calculated over the PDU. The CRC polynomial is defined in Section 3.1.1.

2.2 RESERVED FOR FUTURE USE (RFU)

Any field marked as RFU is reserved for future use. It shall be set to zero on transmission and ignored upon receipt.

2.3 ADVERTISING CHANNEL PDU

The advertising channel PDU has a 16-bit header and a variable size payload. Its format is as shown in Figure 2.2. The 16 bit Header field of the advertising channel PDU is as shown in Figure 2.3.

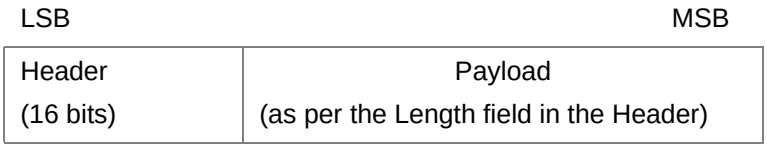


Figure 2.2: Advertising channel PDU

LSB MSB

PDU Type (4 bits)	RFU (2 bits)	TxAdd (1 bit)	RxAdd (1 bit)	Length (6 bits)	RFU (2 bits)

Figure 2.3: Advertising channel PDU Header

The PDU Type field of the advertising channel PDU that is contained in the header indicates the PDU type as defined in Table 2.1.

The TxAdd and RxAdd fields of the advertising channel PDU that are contained in the header contain information specific to the PDU type defined for each advertising channel PDU separately. If the TxAdd or RxAdd fields are not defined as used in a given PDU then they shall be considered Reserved for Future Use.

The Length field of the advertising channel PDU header indicates the payload field length in octets. The valid range of the Length field shall be 6 to 37 octets.

The Payload fields in the advertising channel PDUs are specific to the PDU Type and are defined in Section 2.3.1 through Section 2.3.3. The PDU Types marked as Reserved shall not be sent and shall be ignored upon receipt.

Within advertising channel PDUs, Advertising Data from the Host may be included in the Payload in some PDU Types. The format of this data is defined in [Vol. 3] Part C, Section 17.

PDU Type $b_3b_2b_1b_0$	Packet Name
0000	ADV_IND
0001	ADV_DIRECT_IND
0010	ADV_NONCONN_IND
0011	SCAN_REQ
0100	SCAN_RSP
0101	CONNECT_REQ
0110	ADV_SCAN_IND
0111-1111	Reserved

Table 2.1: Advertising channel PDU Header's PDU Type field encoding

2.3.1 Advertising PDUs

The following advertising channel PDU Types are called advertising PDUs and are used in the specified events:

- ADV_IND: connectable undirected advertising event
- ADV_DIRECT_IND: connectable directed advertising event
- ADV_NONCONN_IND: non-connectable undirected advertising event
- ADV_SCAN_IND: scannable undirected advertising event

These PDUs are sent by the Link Layer in the Advertising State and received by a Link Layer in the Scanning State or Initiating State.

2.3.1.1 ADV_IND

The ADV_IND PDU has the Payload as shown in Figure 2.4. The PDU shall be used in connectable undirected advertising events. The TxAdd in the Flags field indicates whether the advertiser's address in the AdvA field is public (TxAdd = 0) or random (TxAdd = 1).

Payload	
AdvA (6 octets)	AdvData (0-31 octets)

Figure 2.4: ADV_IND PDU Payload

The Payload field consists of AdvA and AdvData fields. The AdvA field shall contain the advertiser's public or random device address as indicated by

TxAdd. The AdvData field may contain Advertising Data from the advertiser's Host.

2.3.1.2 ADV_DIRECT_IND

The ADV_DIRECT_IND PDU has the Payload as shown in Figure 2.5. The PDU shall be used in connectable directed advertising events. The TxAdd in the Flags field indicates whether the advertiser's address in the AdvA field is public (TxAdd = 0) or random (TxAdd = 1). The RxAdd in the Flags field indicates whether the initiator's address in the InitA field is public (RxAdd = 0) or random (RxAdd = 1).

Payload	
AdvA (6 octets)	InitA (6 octets)

Figure 2.5: ADV_DIRECT_IND PDU Payload

The Payload field consists of AdvA and InitA fields. The AdvA field shall contain the advertiser's public or random device address as indicated by TxAdd. The InitA field is the address of the device to which this PDU is addressed. The InitA field shall contain the initiator's public or random device address as indicated by RxAdd.

Note: This packet does not contain any Host data.

2.3.1.3 ADV_NONCONN_IND

The ADV_NONCONN_IND PDU has the Payload as shown in Figure 2.6. The PDU shall be used in non-connectable undirected advertising events. The TxAdd in the Flags field indicates whether the advertiser's address in the AdvA field is public (TxAdd = 0) or random (TxAdd = 1).

Payload	
AdvA (6 octets)	AdvData (0-31 octets)

Figure 2.6: ADV_NONCONN_IND PDU Payload

The Payload field consists of AdvA and AdvData fields. The AdvA field shall contain the advertiser's public or random device address as indicated by TxAdd. The AdvData field may contain Advertising Data from the advertiser's Host.

2.3.1.4 ADV_SCAN_IND[1]

The ADV_SCAN_IND PDU has the Payload as shown in Figure 2.7. The PDU shall be used in scannable undirected advertising events. The TxAdd in the Flags field indicates whether the advertiser's address in the AdvA field is public (TxAdd = 0) or random (TxAdd = 1).

Payload	
AdvA (6 octets)	AdvData (0-31 octets)

Figure 2.7: ADV_SCAN_IND PDU Payload

The Payload field consists of AdvA and AdvData fields. The AdvA field shall contain the advertiser's public or random device address as indicated by TxAdd. The AdvData field may contain Advertising Data from the advertiser's Host.

2.3.2 Scanning PDUs

The following advertising channel PDU Types are called scanning PDUs. They are used in the following states:

- SCAN_REQ: sent by the Link Layer in the Scanning State, received by a Link Layer in the Advertising State
- SCAN_RSP: sent by the Link Layer in the Advertising State, received by a Link Layer in the Scanning State

2.3.2.1 SCAN_REQ

The SCAN_REQ PDU has the Payload as shown in Figure 2.8. The TxAdd in the Flags field indicates whether the scanner's address in the ScanA field is public (TxAdd = 0) or random (TxAdd = 1). The RxAdd in the Flags field indicates whether the advertiser's address in the AdvA field is public (RxAdd = 0) or random (RxAdd = 1).

Payload	
ScanA (6 octets)	AdvA (6 octets)

Figure 2.8: SCAN_REQ PDU Payload

The Payload field consists of ScanA and AdvA fields. The ScanA field shall contain the scanner's public or random device address as indicated by TxAdd. The AdvA field is the address of the device to which this PDU is addressed.

1. ADV_DISCOVER-IND was renamed to ADV_SCAN_IND

The AdvA field shall contain the advertiser's public or random device address as indicated by RxAdd.

Note: This packet does not contain any Host Data.

2.3.2.2 SCAN_RSP

The SCAN_RSP PDU has a format as shown in Figure 2.9. The TxAdd in the Flags field indicates whether the advertiser's address in the AdvA field is public (TxAdd = 0) or random (TxAdd = 1). The Length field indicates the size of the payload (AdvA and ScanRspData) in octets.

Payload	
AdvA (6 octets)	ScanRspData (0-31 octets)

Figure 2.9: SCAN_RSP PDU payload

The Payload field consists of AdvA and ScanRspData fields. The AdvA field shall contain the advertiser's public or random device address as indicated by TxAdd. The ScanRspData field may contain any data from the advertiser's Host.

2.3.3 Initiating PDUs

The following advertising channel PDU Type is called the initiating PDU:

- CONNECT_REQ

This PDU is sent by the Link Layer in the Initiating State and received by the Link Layer in the Advertising State.

2.3.3.1 CONNECT_REQ

The CONNECT_REQ PDU has the Payload as shown in Figure 2.10. TxAdd in the Flags field indicates whether the initiator's device address in the InitA field is public (TxAdd = 0) or random (TxAdd = 1). The RxAdd in the Flags field indicates whether the advertiser's device address in the AdvA field is public (RxAdd = 0) or random (RxAdd = 1).

Payload		
InitA (6 octets)	AdvA (6 octets)	LLData (22 octets)

Figure 2.10: CONNECT_REQ PDU payload

The format of the LLData field is shown in Figure 2.11.

LLData									
AA (4 octets)	CRCInit (3 octets)	WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

Figure 2.11: LLData field structure in CONNECT_REQ PDU's payload

The Payload field consists of InitA, AdvA and LLData fields. The InitA field shall contain the Initiator's public or random device address as indicated by TxAdd. The AdvA field shall contain the advertiser's public or random device address as indicated by RxAdd.

The LLData consists of 10 fields:

- The AA field shall contain the Link Layer connection's Access Address determined by the Link Layer following the rules specified in Section 2.1.2.
- The CRCInit field shall contain the initialization value for the CRC calculation for the Link Layer connection, as defined in Section 3.1.1. It shall be a random value, generated by the Link Layer.
- The WinSize field shall be set to indicate the *transmitWindowSize* value, as defined in Section 4.5.3 in the following manner: *transmitWindowSize* = WinSize * 1.25 ms.
- The WinOffset field shall be set to indicate the *transmitWindowOffset*value, as defined in Section 4.5.3 in the following manner:.*transmitWindowOffset* = WinOffset * 1.25 ms.
- The Interval field shall be set to indicate the *connInterval* as defined in Section 4.5.1 in the following manner:*connInterval* = Interval * 1.25 ms.
- The Latency field shall be set to indicate the *connSlaveLatency*value, as defined in Section 4.5.1 in the following manner: *connSlaveLatency* = Latency.
- The Timeout field shall be set to indicate the *connSupervisionTimeout*value, as defined in Section 4.5.2, in the following manner: *connSupervisionTimeout* = Timeout * 10 ms.
- The ChM field shall contain the channel map indicating *Used* and *Unused* data channels. Every channel is represented with a bit positioned as per the data channel index as defined in Section 1.4.1. The LSB represents data channel index 0 and the bit in position 36 represents data channel index 36. A bit value of '0' indicates that the channel is *Unused*. A bit value of '1' indicates that the channel is *Used*. The bits in positions 37, 38 and 39 are Reserved for Future Use. Note: When mapping from RF Channels to data channel index, care should be taken to remember that there is a gap where the second advertising channel is placed.
- The Hop field shall be set to indicate the *hopIncrement* used in the data channel selection algorithm as defined in Section 4.5.8.2. It shall have a random value in the range of 5 to 16.

- The SCA field shall be set to indicate the *masterSCA* used to determine the worst case Master's sleep clock accuracy as defined in Section 4.2.2. The value of the SCA field shall be set as defined in Table 2.2.

SCA	*masterSCA*
0	251 ppm to 500 ppm
1	151 ppm to 250 ppm
2	101 ppm to 150 ppm
3	76 ppm to 100 ppm
4	51 ppm to 75 ppm
5	31 ppm to 50 ppm
6	21 ppm to 30 ppm
7	0 ppm to 20 ppm

Table 2.2: SCA field encoding

2.4 DATA CHANNEL PDU

The Data Channel PDU has a 16 bit header, a variable size payload, and may include a Message Integrity Check (MIC) field.

The Data Channel PDU is as shown in Figure 2.12.

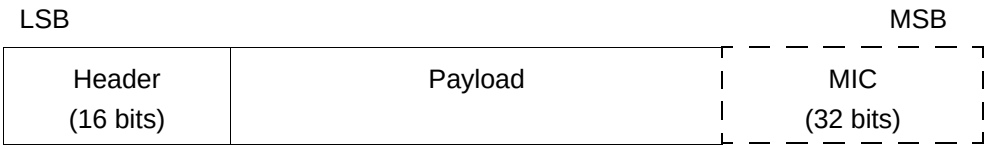


Figure 2.12: Data Channel PDU

The Header field of the Data Channel PDU is as shown in Figure 2.13.

Header						
LLID (2 bits)	NESN (1 bit)	SN (1 bit)	MD (1 bit)	RFU (3 bits)	Length (5 bits)	RFU (3 bits)

Figure 2.13: Data channel PDU header

The 16 bit Header field consists of 5 fields that are specified in Table 2.3.

The MIC field shall not be included in an un-encrypted Link Layer connection, or in an encrypted Link Layer connection with a data channel PDU with a zero length Payload.

The MIC field shall be included in an encrypted Link Layer connection, with a data channel PDU with a non-zero length Payload and shall be calculated as specified in [Vol. 6] Part E, Section 1.

The payload format depends on the LLID field of the Header. If the LLID field is 01b or 10b, the Data Channel PDU Payload field contains an LL Data PDU as defined in Section 2.4.1. If the LLID field is 11b then the Data Channel PDU Payload field contains an LL Control PDU as defined in Section 2.4.2. An LLID field of 00b is reserved.

The NESN bit of the Header is defined in Section 4.5.9.

The SN bit of the Header is defined in Section 4.5.9.

The MD bit of the Header is defined in Section 4.5.6.

The Length field of the Header indicates the length of the Payload and MIC if included. The length field has the range of 0 to 31 octets. The Payload field shall be less than or equal to 27 octets in length. The MIC is 4 octets in length.

Field name	Description
LLID	The LLID indicates whether the packet is an LL Data PDU or an LL Control PDU. 00b = Reserved 01b = LL Data PDU: Continuation fragment of an L2CAP message, or an Empty PDU. 10b = LL Data PDU: Start of an L2CAP message or a complete L2CAP message with no fragmentation. 11b = LL Control PDU
NESN	Next Expected Sequence Number
SN	Sequence Number
MD	More Data
Length	The Length field indicates the size, in octets, of the Payload and MIC, if included.

Table 2.3: Data channel PDU Header field

2.4.1 LL Data PDU

An LL Data PDU is a data channel PDU that is used to send L2CAP data. The LLID field in the Header shall be set to either 01b or 10b.

An LL Data PDU with the LLID field in the Header set to 01b, and the Length field set to 00000b, is known as an Empty PDU. The master's Link Layer may

send an Empty PDU to the slave to allow the slave to respond with any Data Channel PDU, including an Empty PDU.

An LL Data PDU with the LLID field in the Header set to 10b shall not have the Length field set to 00000b.

2.4.2 LL Control PDU

An LL Control PDU is a Data Channel PDU that is used to control the Link Layer connection.

The LL Control PDU Payload is as shown in Figure 2.14.

Payload	
Opcode (1 octet)	CtrData (0 – 22 octets)

Figure 2.14: LL control PDU payload

An LL Control PDU shall not have the Length field set to 00000b. All LL Control PDUs have a fixed length, depending on the Opcode.

The Payload field consists of Opcode and CtrData fields.

The Opcode field identifies different types of LL Control PDU, as defined in Table 2.4.

The CtrData field in the LL Control PDU is specified by the Opcode field and is defined in Section 2.4.2.1 through Section 2.4.2.14.

Opcode	Control PDU Name
0x00	LL_CONNECTION_UPDATE_REQ
0x01	LL_CHANNEL_MAP_REQ
0x02	LL_TERMINATE_IND
0x03	LL_ENC_REQ
0x04	LL_ENC_RSP
0x05	LL_START_ENC_REQ
0x06	LL_START_ENC_RSP
0x07	LL_UNKNOWN_RSP
0x08	LL_FEATURE_REQ
0x09	LL_FEATURE_RSP
0x0A	LL_PAUSE_ENC_REQ
0x0B	LL_PAUSE_ENC_RSP
0x0C	LL_VERSION_IND
0x0D	LL_REJECT_IND
0x0E-0xFF	Reserved for Future Use

Table 2.4: LL Control PDU Opcodes

If an LL Control PDU is received that is not used or not supported, the Link Layer shall respond with an LL_UNKNOWN_RSP PDU. The UnknownType field of the LL_UNKNOWN_RSP PDU shall be set to the value of the not used or not supported Opcode.

If an LL Control PDU is received with an invalid Opcode, i.e. the Opcode field is set to a value that is Reserved for Future Use, or with invalid CtrData fields, the Link Layer shall respond with an LL_UNKNOWN_RSP PDU. The Unknown-Type field of the LL_UNKNOWN_RSP PDU shall be set to the value of the invalid Opcode.

2.4.2.1 LL_CONNECTION_UPDATE_REQ

The format of the CtrData field is as shown in Figure 2.15.

CtrData					
WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	Instant (2 octets)

Figure 2.15: CtrData field of the LL_CONNECTION_UPDATE_REQ PDU

The LL_CONNECTION_UPDATE_REQ CtrData consists of six fields:

- The WinSize field shall be set to indicate the *transmitWindowSize* value, as defined in Section 4.5.3 in the following manner: *transmitWindowSize* = WinSize * 1.25 ms.
- The WinOffset field shall be set to indicate the *transmitWindowOffset value,* as defined in Section 4.5.3, in the following manner: *transmitWindowOffset* = WinOffset * 1.25 ms.
- The Interval field shall be set to indicate the *connInterval* value, as defined in Section 4.4.4, in the following manner: *connInterval* = Interval * 1.25 ms.
- The Latency field shall be set to indicate the *connSlaveLatency* value, as defined by Section 4.5.1, in the following manner: *connSlaveLatency* = Latency.
- The Timeout field shall be set to indicate the *connSupervisionTimeout* value, as defined by Section 4.5.1, in the following manner*:* -connSupervisionTimeout= Timeout * 10 ms.
- The Instant field shall be set to indicate the *connInstant value, as* defined by Section 5.1.1. The Instant Field shall have a value in the range of 1 to 32767.

2.4.2.2 LL_CHANNEL_MAP_REQ

The format of the CtrData field is shown in Figure 2.16.

CtrData	
ChM (5 octets)	Instant (2 octets)

Figure 2.16: CtrData field of the LL_CHANNEL_MAP_REQ PDU

The LL_CHANNEL_MAP_REQ CtrData consists of two fields:

- The ChM field shall contain the channel map indicating *Used* and *Unused* data channels. Every channel is represented with a bit positioned as per the data channel index defined by Section 4.5.8. The format of this field is identical to the ChM field in the CONNECT_REQ PDU (see Section 2.3.3.1).

- The Instant field shall be set to indicate the *connInstant* value, as defined by Section 5.1.2. The Instant field shall have a value in the range of 1 to 32767.

2.4.2.3 LL_TERMINATE_IND

The format of the CtrData field is shown in Figure 2.17.

CtrData
Error Code (1 octet)

Figure 2.17: CtrData field of the LL_TERMINATE_IND PDU

The LL_TERMINATE_IND CtrData consists of one field:

- The Error Code field shall be set to inform the remote device why the connection is about to be terminated. See [Vol 2] Part D for details.

2.4.2.4 LL_ENC_REQ

The format of the CtrData field is shown in Figure 2.18.

CtrData			
Rand (8 octets)	EDIV (2 octets)	SKDm (8 octets)	IVm (4 octets)

Figure 2.18: CtrData field of the LL_ENC_REQ PDU

The LL_ENC_REQ CtrData consists of four fields:

- The Rand field contains a random number that is provided by the Host and used with EDIV (see [Vol. 3] Part H, Section 2.4.4).
- The EDIV field contains the encrypted diversifier.
- The SKDm field contains the master's portion of the session key identifier.
- The IVm field contains the master's portion of the initialization vector.

2.4.2.5 LL_ENC_RSP

The format of the CtrData field is shown in Figure 2.19.

CtrData	
SKDs (8 octets)	IVs (4 octets)

Figure 2.19: CtrData field of the LL_ENC_ RSP PDU

The LL_ENC_RSP CtrData consists of two fields.

- The SKDs field shall contain the slave's portion of the session key identifier.

- The IVs field shall contain the slave's portion of the initialization vector.

2.4.2.6 LL_START_ENC_REQ

The LL_START_ENC_REQ PDU does not have a CtrData field.

2.4.2.7 LL_START_ENC_RSP

The LL_START_ENC_RSP PDU does not have a CtrData field.

2.4.2.8 LL_UNKNOWN_RSP

The format of the CtrData field is shown in Figure 2.20.

CtrData
UnknownType (1 octet)

Figure 2.20: CtrData field of the LL_UNKNOWN_RSP PDU

The LL_UNKNOWN_RSP CtrData consists of one field:

- UnknownType shall contain the Opcode field value of the received LL Control PDU.

2.4.2.9 LL_FEATURE_REQ

The format of the CtrData field is shown in Figure 2.21.

CtrData
FeatureSet (8 octets)

Figure 2.21: CtrData field of the LL_FEATURE_REQ PDU

The LL_FEATURE_REQ CtrData consists of one field:

- FeatureSet shall contain the set of supported features of the master's Link Layer.

2.4.2.10 LL_FEATURE_RSP

The format of the CtrData field is shown in Figure 2.22.

CtrData
FeatureSet (8 octets)

Figure 2.22: CtrData field of the LL_FEATURE_RSP PDU

The LL_FEATURE_RSP CtrData consists of one field:

- FeatureSet shall contain the set of used features of the slave's Link Layer.

2.4.2.11 LL_PAUSE_ENC_REQ

The LL_PAUSE_ENC_REQ packet does not have a CtrData field.

2.4.2.12 LL_PAUSE_ENC_RSP

The LL_PAUSE_ENC_RSP packet does not have a CtrData field.

2.4.2.13 LL_VERSION_IND

The format of the CtrData field is shown in Figure 2.23.

CtrData		
VersNr (1 octet)	CompId (2 octets)	SubVersNr (2 octets)

Figure 2.23: CtrData field of the LL_VERSION_IND PDU

The LL_VERSION_IND CtrData consists of three fields:

- VersNr field shall contain the version of the Bluetooth Controller specification (see Bluetooth Assigned Numbers).
- CompId field shall contain the company identifier of the manufacturer of the Bluetooth Controller (see Bluetooth Assigned Numbers).
- SubVersNr field shall contain a unique value for each implementation or revision of an implementation of the Bluetooth Controller.

2.4.2.14 LL_REJECT_IND

The format of the CtrData field is shown in Figure 2.24.

CtrData
Error Code (1 octet)

Figure 2.24: CtrData field of the LL_ REJECT_IND

Error Code shall contain the reason a request was rejected; see [Vol 2] Part D.

3 BIT STREAM PROCESSING

Bluetooth devices shall use the bitstream processing schemes as defined in the following sections.

Figure 3.1 shows the processes that may have to be carried out on the PDU.

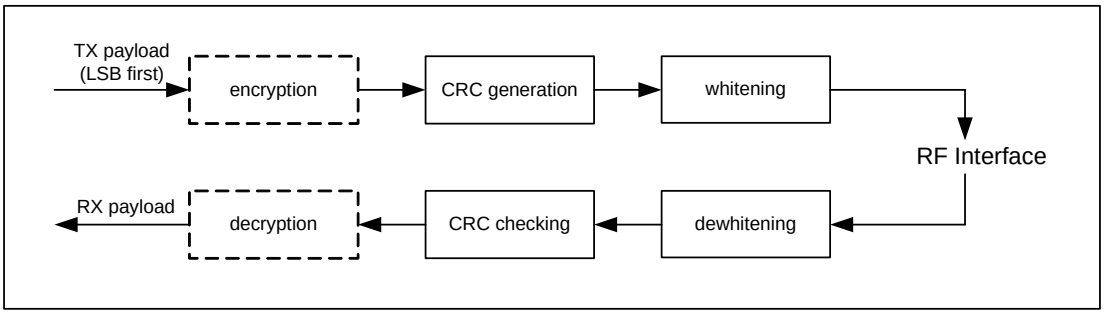


Figure 3.1: Payload bit processes

3.1 ERROR CHECKING

At packet reception, the Access Address shall be checked first. If the Access Address is incorrect, the packet shall be rejected, otherwise the packet shall be considered received. If the CRC is incorrect, the packet shall be rejected, otherwise the packet shall be considered valid. A packet shall only be processed if the packet is considered valid. A packet with an incorrect CRC may cause a connection event to continue, as specified in Section 4.5.1.

3.1.1 CRC Generation

The CRC shall be calculated on the PDU field in all Link Layer packets. If the PDU is encrypted, then the CRC shall be calculated after encryption of the PDU has been performed.

The CRC polynomial is a 24-bit CRC and all bits in the PDU shall be processed in transmitted order starting from the least significant bit. The polynomial has the form of $x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$. For every Data Channel PDU, the shift register shall be preset with the CRC initialization value set for the Link Layer connection and communicated in the CONNECT_REQ PDU. For every Advertising Channel PDU the shift register shall be preset with 0x555555.

Position 0 shall be set as the least significant bit and position 23 shall be set as the most significant bit of the initialization value. The CRC is transmitted most significant bit first, i.e. from position 23 to position 0 (see Section 1.2).

Figure 3.2 shows an example linear feedback shift register (LFSR) to generate the CRC.

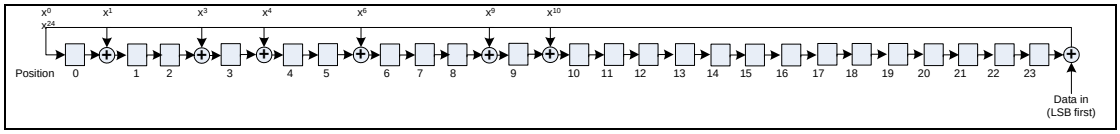


Figure 3.2: The LFSR circuit generating the CRC

3.2 DATA WHITENING

Data whitening is used to avoid long sequences of zeros or ones, e.g. 0000000b or 1111111b, in the data bit stream. Whitening shall be applied on the PDU and CRC fields of all Link Layer PDUs and is performed after the CRC in the transmitter. De-whitening is performed before the CRC in the receiver (see Figure 3.1).

The whitener and de-whitener are defined the same way, using a 7-bit linear feedback shift register with the polynomial $x^7 + x^4 + 1$. Before whitening or de-whitening, the shift register is initialized with a sequence that is derived from the channel index (data channel index or advertising channel index) in which the packet is transmitted in the following manner:

- Position 0 is set to one.
- Positions 1 to 6 are set to the channel index of the channel used when transmitting or receiving, from the most significant bit in position 1 to the least significant bit in position 6.

For example, if the channel index = 23 (0x17), the positions would be set as follows:

Position 0 = 1

Position 1 = 0

Position 2 = 1

Position 3 = 0

Position 4 = 1

Position 5 = 1

Position 6 = 1

Figure 3.3 shows an example linear feedback shift register (LFSR) to generate data whitening.

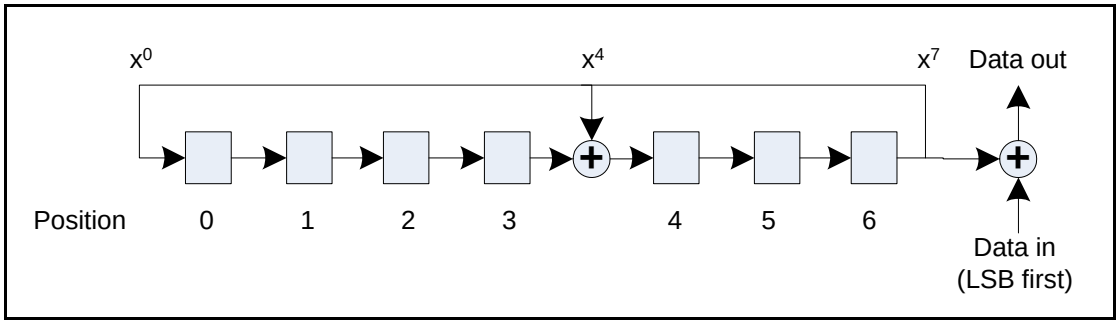


Figure 3.3: The LFSR circuit to generate data whitening

4 AIR INTERFACE PROTOCOL

The air interface protocol consists of the multiple access scheme, device discovery and link layer connection methods.

4.1 INTER FRAME SPACE

The time interval between two consecutive packets on the same channel index is called the Inter Frame Space. It is defined as the time from the end of the last bit of the previous packet to the start of the first bit of the subsequent packet. The Inter Frame Space is designated “T_IFS” and shall be 150 µs.

4.2 TIMING REQUIREMENTS

The Link Layer shall use one of two possible clock accuracies. During a connection event or advertising event the Link Layer shall use the active clock accuracy; otherwise it shall use the sleep clock accuracy.

4.2.1 Active Clock Accuracy

The average timing of packet transmission during a connection event is determined using the active clock accuracy, with a drift less than or equal to ±50 ppm. All instantaneous timings shall not deviate more than 2 µs from the average timing.

Note: This means that the start of a packet shall be transmitted 150±2 µs after the end of the previous packet.

4.2.2 Sleep Clock Accuracy

The timing of advertising events (see Section 4.4.2.2) and connection events (see Section 4.5.7) is determined using the sleep clock accuracy, with a drift less than or equal to ±500 ppm.

The instantaneous timing of the anchor point (see Section 4.5.7) shall not deviate more than 16 µs from the average timing.

Note: This means that a 1 sec connection interval with a total ±1000ppm sleep clock accuracy will give a window widening either side of the anchor point of 1ms plus 16us, assuming that the slave controller was using its sleep clock for almost the complete connection interval.

4.3 LINK LAYER DEVICE FILTERING

The Link Layer may perform device filtering based on the device address of the peer device. Link Layer Device Filtering is used by the Link Layer to minimize the number of devices to which it responds.

A Link Layer shall support Link Layer Device Filtering unless it only supports non-connectable advertising.

The filter policies for the Advertising State, Scanning State and Initiating State are independent of each other. When the Link Layer is in the Advertising State, the advertising filter policy shall be used. When the Link Layer is in the Scanning State, the scanning filter policy shall be used. When the Link Layer is in the Initiating State, the initiator filter policy shall be used. If the Link Layer does not support the Advertising State, Scanning State, or Initiating State, the corresponding filter policy is not required to be supported.

4.3.1 White List

The set of devices that the Link Layer uses for device filtering is called the White List.

A White List contains a set of White List Records used for Link Layer Device Filtering. A White List Record contains both the device address and the device address type (public or random). All Link Layers supporting Link Layer Device Filtering shall support a White List capable of storing at least one White List Record.

On reset, the White List shall be empty.

The White List is configured by the Host and is used by the Link Layer to filter advertisers, scanners or initiators. This allows the Host to configure the Link Layer to act on a request without awakening the Host.

All the device filter policies shall use the same White List.

4.3.2 Advertising Filter Policy

The advertising filter policy determines how the advertiser's Link Layer processes scan and/or connection requests.

When the Link Layer is using connectable directed advertising the advertising filter policy shall be ignored, otherwise the Link Layer shall use one of the following advertising filter policy modes which are configured by the Host:

- The Link Layer shall process scan and connection requests only from devices in the White List.
- The Link Layer shall process scan and connection requests from all devices (i.e. the White List is not in use). This is the default on reset.
- The Link Layer shall process scan requests from all devices and shall only process connection requests from devices that are in the White List.
- The Link Layer shall process connection requests from all devices and shall only process scan requests from devices that are in the White List.

Only one advertising filter policy mode shall be supported at a time.

4.3.3 Scanner Filter Policy

The scanner filter policy determines how the scanner's Link Layer processes advertising packets. The Link Layer shall use one of the following scanner filter policy modes which are configured by the Host:

- The Link Layer shall process advertising packets only from devices in the White List.
- The Link Layer shall process all advertising packets (i.e., the White List is not used). This is the default on reset.

In addition to the scanner filter policy, a connectable directed advertising packet not containing the scanner's device address shall be ignored.

Only one scanner filter policy mode shall be supported at a time.

4.3.4 Initiator Filter Policy

The initiator filter policy determines how an initiator's Link Layer processes advertising packets. The Link Layer shall use one of the following initiator filter policy modes which are configured by the Host:

- The Link Layer shall process connectable advertising packets from all devices in the White List.
- The Link Layer shall ignore the White List and process connectable advertising packets from a specific single device specified by the Host.

If the Link Layer receives a connectable directed advertising packet from an advertiser that is not contained in the White List or the single address specified by the Host, the connectable directed advertising packet shall be ignored.

Only one initiator filter policy mode shall be supported at a time.

4.4 NON-CONNECTED STATES

4.4.1 Standby State

The Standby State is the default state in the Link Layer. The Link Layer shall not send or receive packets in the Standby State. The Link Layer may leave the Standby State to enter the Advertising State, Scanning State or Initiator State.

4.4.2 Advertising State

The Link Layer shall enter the Advertising State when directed by the Host. When placed in the Advertising State, the Link Layer shall send advertising PDUs (see Section 2.3.1) in advertising events.

Each advertising event is composed of one or more advertising PDUs sent on used advertising channel indexes. The advertising event shall be closed after one advertising PDU has been sent on each of the used advertising channel indexes (see Section 4.4.2.1) or the advertiser may close an advertising event earlier to accommodate other functionality.

The time between two consecutive advertising events is defined in Section 4.4.2.2.

An advertising event can be one of the following types:

- a connectable undirected event
- a connectable directed event
- a non-connectable undirected event
- a scannable undirected event

For each advertising event type, a corresponding Advertising Channel PDU is used.

The first PDU of each advertising event shall be transmitted in the used advertising channel with the lowest advertising channel index.

The advertising event type determines the allowable response PDUs. The table below specifies the allowable responses for each advertising event.

Advertising Event Type	**PDU used in this advertising event type**	**Allowable response PDUs for advertising event**	
		SCAN_REQ	CONNECT_REQ
Connectable Undirected Event	ADV_IND	YES	YES
Connectable Directed Event	ADV_DIRECT_IND	NO	YES*
Non-connectable Undirected Event	ADV_NONCONN_IND	NO	NO
Scannable Undirected Event	ADV_SCAN_IND	YES	NO

Table 4.1: Advertising event types, PDUs used and allowable response PDUs

* Only the correctly addressed initiator may respond.

If the advertiser receives a PDU for the advertising event that is not explicitly allowed it shall be ignored. If no PDU is received or the received PDU was ignored, the advertiser shall either send an advertising PDU on the next used advertising channel index or close the advertising event.

4.4.2.1 Advertising Channel Selection

Advertising events use three predefined advertising channels. Advertising channel indexes are either used or unused.

The Link Layer shall use the advertising channel indexes as specified by the Host, and the used advertising channel indexes shall take effect when the Advertising State is entered.

4.4.2.2 Advertising Interval

For all undirected advertising events, the time between the start of two consecutive advertising events (*T_advEvent*) is computed as follows for each advertising event:

T_advEvent = advInterval + advDelay

The a*dvInterval* shall be an integer multiple of 0.625 ms in the range of 20 ms to 10.24 s. If the advertising event type is either a scannable undirected event type or a non-connectable undirected event type, the *advInterval* shall not be less than 100 ms. If the advertising event type is a connectable undirected event type, the *advInterval* can be 20 ms or greater.

The *advDelay* is a pseudo-random value with a range of 0 ms to 10 ms generated by the Link Layer for each advertising event.

As illustrated in Figure 4.1, the advertising events are perturbed in time using the advDelay.

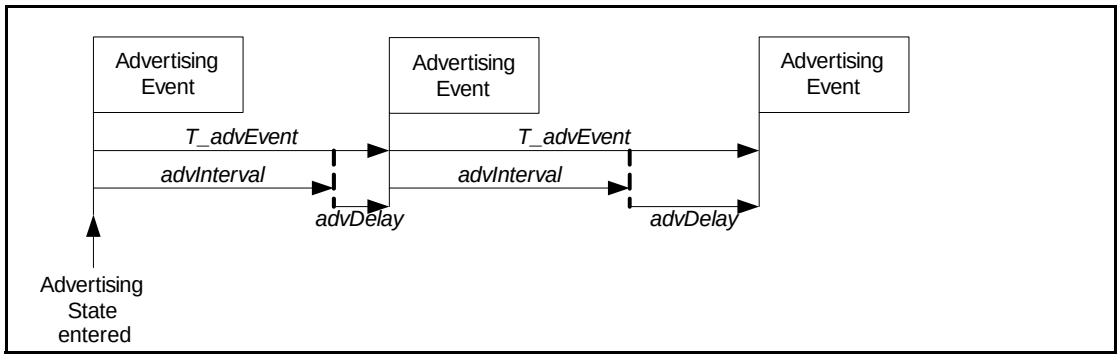


Figure 4.1: Advertising events perturbed in time using advDelay

4.4.2.3 Connectable Undirected Event Type

When the connectable undirected advertising event type is used, advertising indications (ADV_IND PDU) are sent by the Link Layer.

The connectable undirected advertising event type allows a scanner or initiator to respond with either a scan request or connect request. A scanner may send a scan request (SCAN_REQ PDU) to request additional information about the

advertiser. An initiator may send a connect request (CONNECT_REQ PDU) to request the Link Layer to enter the Connection State.

The Link Layer shall listen on the same advertising channel index for requests from scanners or initiators.

If the advertiser receives a SCAN_REQ PDU that contains its device address from a scanner allowed by the advertising filter policy, it shall reply with a SCAN_RSP PDU on the same advertising channel index. After the SCAN_RSP PDU is sent, or if the advertising filter policy prohibited processing the SCAN_REQ PDU, the advertiser shall either move to the next used advertising channel index to send another ADV_IND PDU, or close the advertising event.

If the advertiser receives a CONNECT_REQ PDU that contains its device address, from an initiator allowed by the advertising filter policy, the Link Layer shall exit the Advertising State and transition to the Connection State in the Slave Role as defined in Section 4.5.5. If the advertising filter policy prohibited processing the received CONNECT_REQ PDU, the advertiser shall either move to the next used advertising channel index to send another ADV_IND PDU, or close the advertising event.

The time between the beginning of two consecutive ADV_IND PDUs within an advertising event shall be less than or equal to 10 ms. The advertising state shall be closed within the advertising interval.

An illustration of an advertising event using all the advertising channel indexes and in which no SCAN_REQ or CONNECT_REQ PDUs are received is shown in Figure 4.2.

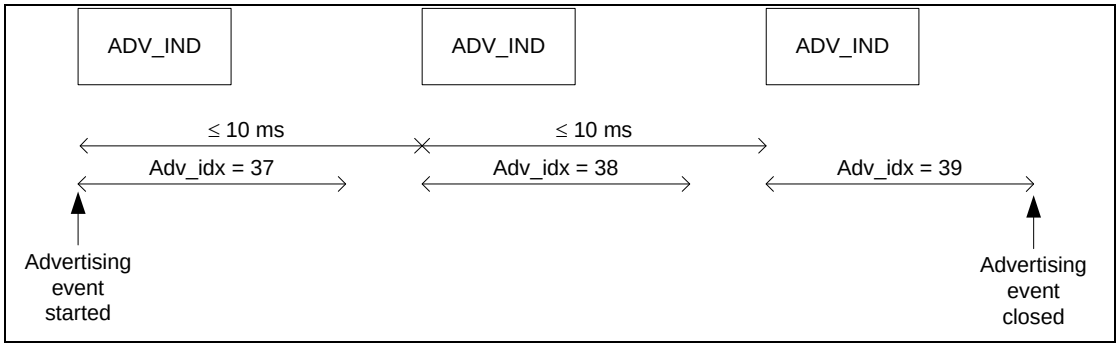


Figure 4.2: Connectable undirected advertising event with only advertising PDUs

Two illustrations of advertising events using all the advertising channel indexes during which a SCAN_REQ PDU is received and a SCAN_RSP PDU is sent are shown in Figure 4.3 and in Figure 4.4.

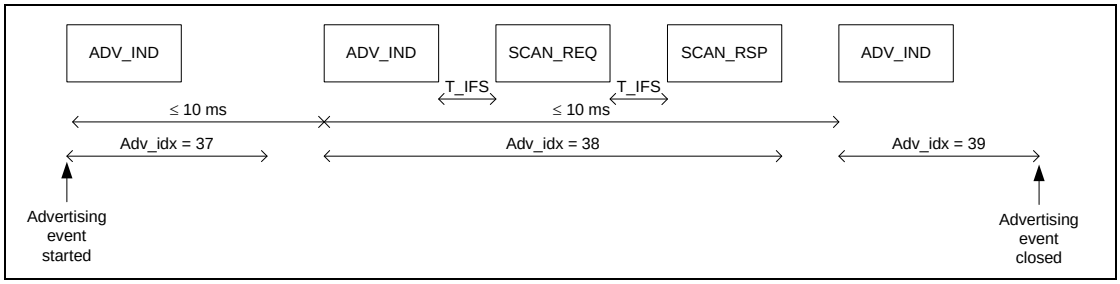


Figure 4.3: Connectable undirected advertising event with SCAN_REQ and SCAN_RSP PDUs in the middle of an advertising event

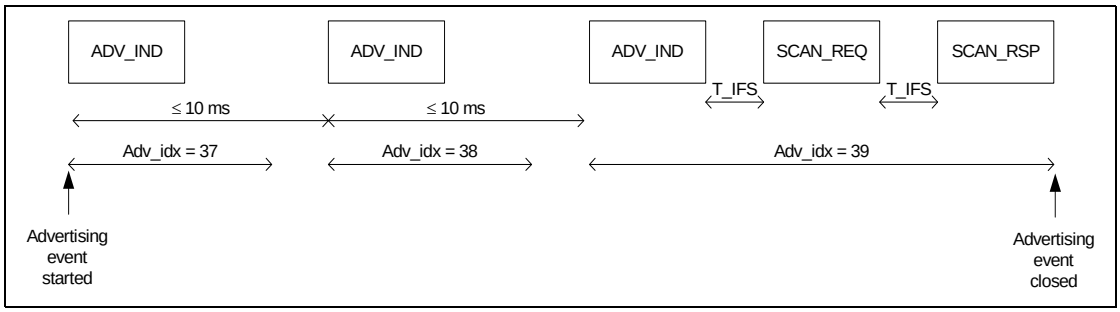
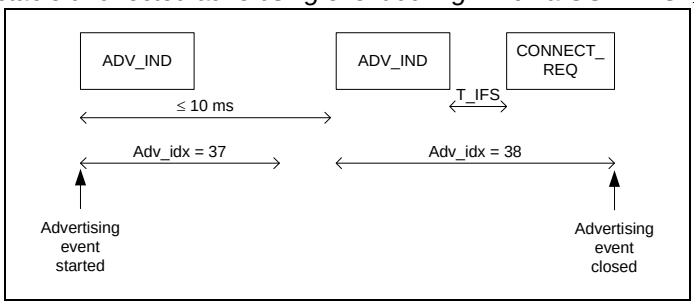


Figure 4.4: Connectable undirected advertising event with SCAN_REQ and SCAN_RSP packets at the end of an advertising event

Figure 4.5 illustrates an advertising event during which a CONNECT_REQ PDU is received on the second advertising channel index.

Figure 4.5: Connectable undirected advertising event during which a CONNECT_REQ PDU is received



4.4.2.4 Connectable Directed Event Type

When the connectable directed advertising event type is used, directed advertising indications (ADV_DIRECT_IND PDUs) are sent by the Link Layer.

The connectable directed advertising event type allows an initiator to respond with a connect request. An initiator may send a connect request (CONNECT_REQ PDU) to request the Link Layer to enter the Connection State.

The ADV_DIRECT_IND PDU contains both the initiator's device address and the advertiser's device address. Only the addressed initiator may initiate a Link

Layer connection with the advertiser by sending a CONNECT_REQ PDU to the advertiser.

After every ADV_DIRECT_IND PDU sent by the advertiser, the advertiser shall listen for CONNECT_REQ PDUs on the same advertising channel index. Any SCAN_REQ PDUs received shall be ignored.

If the advertiser receives a CONNECT_REQ PDU that contains its device address and the initiator device address is contained in the ADV_DIRECT_IND PDU, the Link Layer shall exit the Advertising State and transition to the Connection State in the Slave Role as defined in Section 4.5.5. Otherwise, the advertiser shall either move to the next used advertising channel index to send another ADV_DIRECT_IND PDU, or close the advertising event.

The time between the start of two consecutive ADV_DIRECT_IND PDUs sent on the same advertising channel index shall be less than or equal to 3. 75 ms.

The Link Layer shall exit the Advertising State no later than 1.28 s after the Advertising State was entered.

A sequence of five ADV_DIRECT_IND PDUs in two advertising events without CONNECT_REQ PDUs is shown in Figure 4.6 for the case in which all the advertising channels are used.

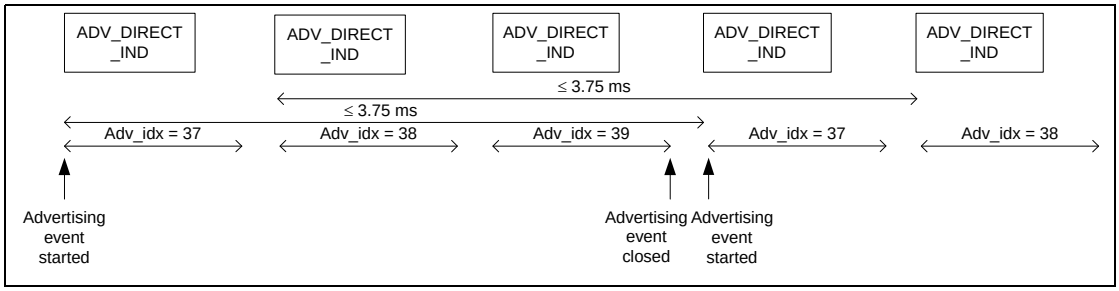


Figure 4.6: Connectable directed advertising event type with only advertising PDUs

Note: Connectable directed advertising is designed for cases in which fast Link Layer connection setup is essential (e.g., a reconnection). It is a power and bandwidth intensive advertising scheme that should only be used when fast connection setup is required.

4.4.2.5 Scannable Undirected Event Type[1]

When the scannable undirected advertising event type is used, scannable advertising indications (ADV_SCAN_IND PDUs) packets are sent by the Link Layer.

1. Discoverable Undirected Event Type was renamed to Scannable Undirected Event Type

The scannable undirected event type allows a scanner to respond with a scan request (SCAN_REQ PDU) to request additional information about the advertiser.

The Link Layer shall listen on the same advertising channel index for requests from scanners.

If the advertiser receives a SCAN_REQ PDU that contains its device address from a scanner allowed by the advertising filter policy it shall reply with a SCAN_RSP PDU on the same advertising channel index. After the SCAN_RSP PDU is sent or if the advertising filter policy prohibited processing the SCAN_REQ PDU the advertiser shall either move to the next used advertising channel index to send another ADV_SCAN_IND PDU, or close the advertising event.

The time between the beginning of two consecutive ADV_SCAN_IND PDUs within an advertising event shall be less than or equal to 10ms. The advertising event shall be closed within the advertising interval.

The structure of an advertising event in which no SCAN_REQ PDU was received is shown in Figure 4.7 for the case in which all the advertising channels are used.

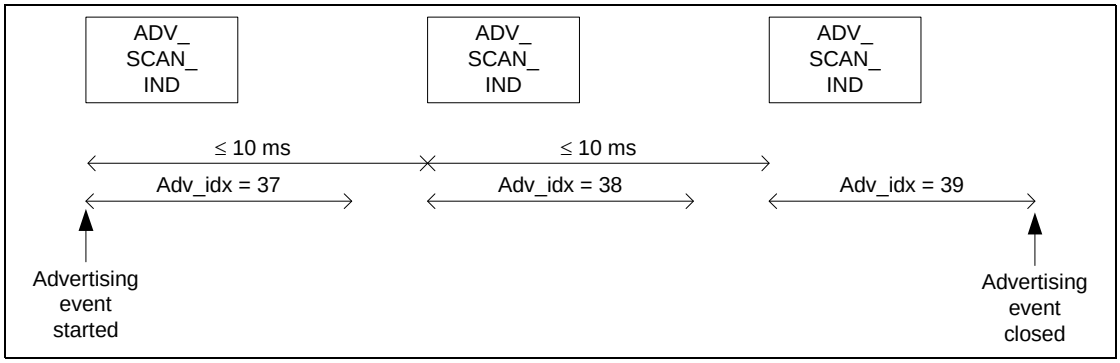


Figure 4.7: Scannable undirected advertising event with only advertising PDUs

Two example advertising events during which a SCAN_REQ PDU is received and a SCAN_RSP PDU is sent are shown in Figure 4.8 and in Figure 4.9 for the case in which all the advertising channels are used.

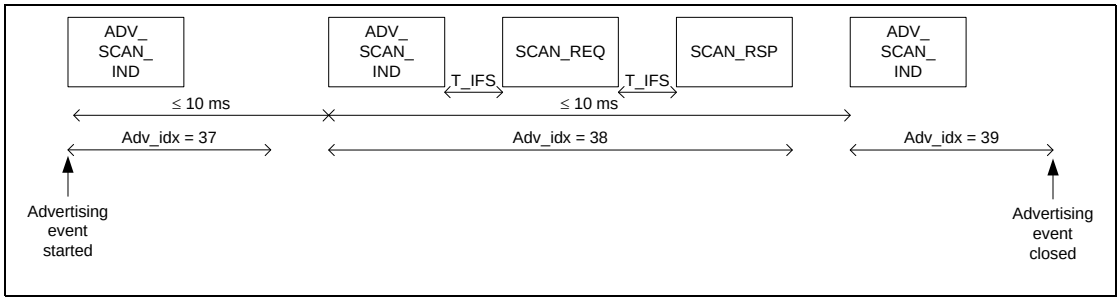


Figure 4.8: Scannable undirected advertising event with SCAN_REQ and SCAN_RSP PDUs in the middle of an advertising event

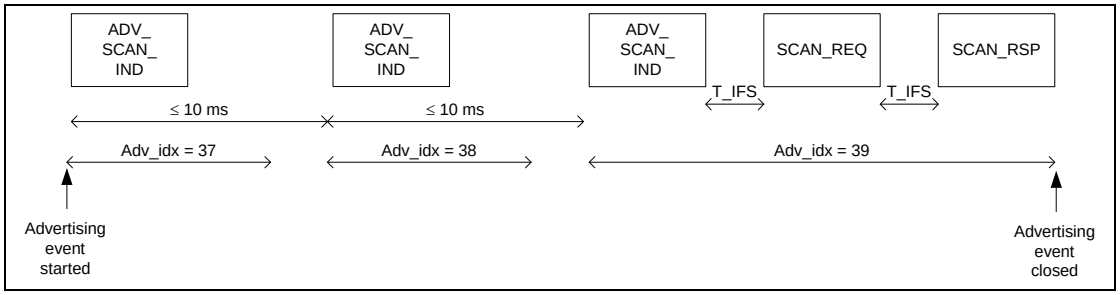


Figure 4.9: Scannable undirected advertising event with SCAN_REQ and SCAN_RSP PDUs at the end of an advertising event

4.4.2.6 Non-connectable Undirected Event Type

When the non-connectable undirected event type is used, non-connectable advertising indications (ADV_NONCONN_IND PDU) packets are sent by the Link Layer.

The non-connectable undirected event type allows a scanner to receive information contained in the ADV_NONCONN_IND PDU from the advertiser.

The advertiser shall either move to the next used advertising channel index or close the advertising event after each ADV_NONCONN_IND PDU that is sent. The Link Layer does not listen, and therefore cannot receive any requests from scanners or initiators.

The time between the beginning of two consecutive ADV_NONCONN_IND PDUs within an advertising event shall be less than or equal to 10 ms. The advertising event shall be closed within the advertising interval.

An illustration of a non-connectable advertising event is shown in Figure 4.10 for the case in which all the advertising channels are used.

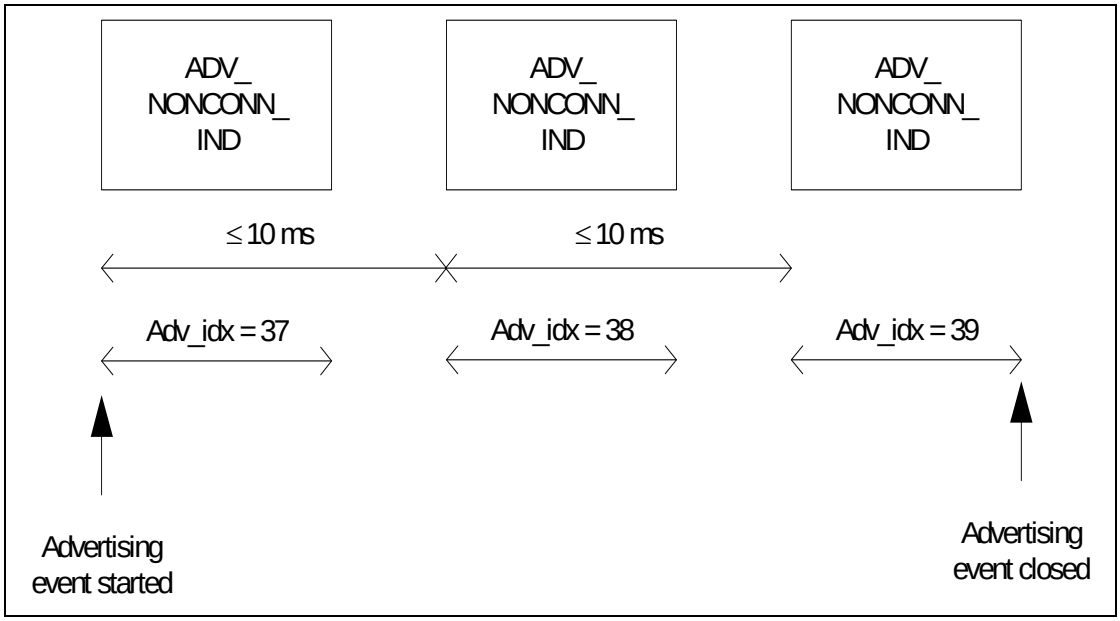


Figure 4.10: Non-connectable undirected advertising event

4.4.3 Scanning State

The Link Layer shall enter the Scanning State when directed by the Host. When scanning, the Link Layer shall listen on the advertising channel indices. There are two types of scanning, determined by the Host: passive and active.

There are no strict timing or advertising channel index selection rules for scanning.

During scanning, the Link Layer listens on an advertising channel index for the duration of the scan window, *scanWindow*. The scan interval, *scanInterval*, is defined as the interval between the start of two consecutive scan windows.

The Link Layer should listen for the complete *scanWindow* every *scanInterval* as directed by the Host unless there is a scheduling conflict. In each scan window, the Link Layer should scan on a different advertising channel index. The Link Layer shall use all the advertising channel indices.

The *scanWindow* and *scanInterval* parameters shall be less than or equal to 10.24 s. The *scanWindow* shall be less than or equal to the *scanInterval*. If the *scanWindow* and the *scanInterval* parameters are set to the same value by the Host, the Link Layer should scan continuously.

The scanner filter policy shall apply when receiving an advertising PDU when scanning.

For each non-duplicate ADV_DIRECT_IND PDU received that contains this Link Layer's device address from an advertiser the Link Layer shall send an advertising report to the Host. For each non-duplicate ADV_IND, ADV_SCAN_IND, ADV_NONCONN_IND, or SCAN_RSP PDU from an adver-

tiser, the Link Layer shall send an advertising report to the Host. The advertising report shall contain at least the advertiser's device address and advertising data or scan response data if present. Duplicate advertising reports are not required to be sent to the Host. A duplicate advertising report is an advertising report for the same device address while the Link Layer stays in the Scanning State. The advertising data may change; advertising data or scan response data is not considered significant when determining duplicate advertising reports.

4.4.3.1 Passive Scanning

When in passive scanning, the Link Layer will only receive packets; it shall not send any packets.

4.4.3.2 Active Scanning

In active scanning, the Link Layer shall listen for advertising PDUs and depending on the advertising PDU type it may request an advertiser to send additional information.

The Link Layer shall not send a SCAN_REQ PDU to an advertiser from which an ADV_DIRECT_IND PDU or ADV_NONCONN_IND PDU is received.

The Link Layer shall send at least one SCAN_REQ PDU after entering the Scanning State to advertisers from which ADV_IND or ADV_SCAN_IND PDUs are received, The Link Layer shall send further SCAN_REQ PDUs to advertisers from which ADV_IND or ADV_SCAN_IND PDUs have been received. The Link Layer should interleave SCAN_RSP PDUs to multiple advertisers.

The scanner shall run a backoff procedure to minimize collisions of SCAN_REQ PDUs from multiple scanners.

The backoff procedure uses two parameters, *backoffCount* and *upperLimit* to restrict the number of SCAN_REQ PDUs sent when collisions occur on SCAN_RSP PDUs.

Upon entering Scanning State, the *upperLimit* shall be set to one and the *backoffCount* shall be set to one.

On every received ADV_IND PDU or ADV_SCAN_IND PDU that is allowed by the scanner filter policy and a SCAN_REQ PDU is to be sent the *backoffCount* shall be decremented by one until it reaches the value of zero. The SCAN_REQ PDU shall only be sent when *backoffCount* becomes zero.

After sending a SCAN_REQ PDU the Link Layer shall listen for a SCAN_RSP PDU from that advertiser. If the SCAN_RSP PDU was not received from that advertiser, it is considered a failure otherwise it is considered a success. On every two consecutive failures, the *upperLimit* shall be doubled until it reaches

the value of 256. On every two consecutive successes, the *upperLimit* shall be halved until it reaches the value of one. After success or failure of receiving the SCAN_RSP PDU, the Link Layer shall set *backoffCount* to a new pseudo-random integer between one and *upperLimit*.

Two illustrations of advertising events using all the advertising channel indexes during which a SCAN_REQ PDU is received and a SCAN_RSP PDU is sent are shown in Figure 4.3 and in Figure 4.4.

4.4.4 Initiating State

The Link Layer shall enter the Initiating State when directed by the Host. When initiating, the Link Layer shall listen on the advertising channel indices.

There are no strict timing or advertising channel index selection rules for initiators.

During initiating, the Link Layer listens on an advertising channel index for the duration of the scan window, *scanWindow*. The scan interval, *scanInterval*, is defined as the interval between the start of two consecutive scan windows.

The Link Layer should listen for the complete *scanWindow* every *scanInterval* as directed by the Host unless there is a scheduling conflict. In each scan window, the Link Layer should listen on a different advertising channel index. The Link Layer shall use all the advertising channel indexes.

The *scanWindow* and *scanInterval* parameters shall be less than or equal to 10.24 s. The *scanWindow* shall be less than or equal to the *scanInterval*. If the *scanWindow* and the *scanInterval* parameters are set to the same value by the Host, the Link Layer should listen continuously.

If an ADV_IND PDU is received that is allowed by the initiator filter policy, the initiator shall send a CONNECT_REQ PDU to the advertiser. If an ADV_DIRECT_IND PDU with this Link Layer's device address is received that is allowed by the initiator filter policy, the initiator shall send a CONNECT_REQ PDU to the advertiser; otherwise it shall be ignored.

After sending the CONNECT_REQ PDU, the Link Layer shall exit the Initiating State, and shall transition to the Connection State in the Master Role as defined in Section 4.5.4.

4.5 CONNECTION STATE

The Link Layer enters the Connection State when an initiator sends a CONNECT_REQ PDU to an advertiser or an advertiser receives a CONNECT_REQ PDU from an initiator.

After entering the Connection State, the connection is considered to be created. The connection is not considered to be established at this point. A con-

nection is only considered to be established once a data channel packet has been received from the peer device. The only difference between a connection that is created and a connection that is established is the Link Layer connection supervision timeout value that is used (see Section 4.5.2).

When two devices are in a connection, the two devices act in different roles. A Link Layer in the Master Role is called a master. A Link Layer in the Slave Role is called a slave. The master controls the timing of a connection event. A connection event is a point of synchronization between the master and the slave.

4.5.1 Connection Events

The Link Layer in the Connection State shall only transmit Data Channel PDUs (see Section 2.4) in connection events. The master and slave shall determine the data channel index for each connection event as defined in Section 4.5.8. The same data channel index shall be used for all packets in the connection event. Each connection event contains at least one packet sent by the master.

During a connection event, the master and slave alternate sending and receiving packets. The connection event is considered open while both devices continue to send packets. The slave shall always send a packet if it receives a packet from the master regardless of a valid CRC match, except after multiple consecutive invalid CRC matches as specified in Section 4.5.6. The master may send a packet if it receives a packet from the slave regardless of a valid CRC match. The Length field of the Header is assumed to be correct even if the CRC match was invalid. If the master does not receive a packet from the slave, the master shall close the connection event.

The connection event can be closed by either device, as defined in Section 4.5.6.

The timing of connection events is determined by two parameters: connection event interval (*connInterval*), and slave latency (*connSlaveLatency*).

The start of a connection event is called an anchor point. At the anchor point, a master shall start to transmit a Data Channel PDU to the slave. The start of connection events are spaced regularly with an interval of *connInterval* and shall not overlap. The master shall ensure that a connection event closes at least T_IFS before the anchor point of the next connection event. The slave listens for the packet sent by its master at the anchor point.

The *connInterval* shall be a multiple of 1.25 ms in the range of 7.5 ms to 4.0 s. The *connInterval* is set by the Initiator's Link Layer in the CONNNECT_REQ PDU from the range given by the Host.

Slave latency allows a slave to use a reduced number of connection events. The connSlaveLatency parameter defines the number of consecutive connection events that the slave device is not required to listen for the master. The value of connSlaveLatency should not cause a Supervision Timeout (see Sec-

tion 4.5.2). connSlaveLatency shall be an integer in the range of 0 to ((connSupervisionTimeout / connInterval) - 1). connSlaveLatency shall also be less than 500. When connSlaveLatency is set to zero the slave device shall listen at every anchor point. If the slave does not receive a packet from the master after applying slave latency, it should listen at each anchor point and not apply slave latency until it receives a packet from the master.

Both the master and the slave shall have a 16-bit connection event counter (*connEventCounter*) for each Link Layer connection. It shall be set to zero on the first connection event sent by the master of the connection. It shall be incremented by one for each new connection event sent by the master; the *connEventCounter* shall wrap from 0xFFFF to 0x0000. This counter is used to synchronize Link Layer control procedures.

The slave shall increment *connEventCounter* for all connection events, even if it may not be listening to the master due to slave latency in those events.

4.5.2 Supervision Timeout

A connection can break down due to various reasons such as a device moving out of range, encountering severe interference or a power failure condition. Since this may happen without any prior warning, it is important for both the master and the slave to monitor the status of the connection.

To be able to detect link loss, both the master and the slave shall use a Link Layer connection supervision timer, $T_{LLconnSupervision}$. Upon reception of a valid packet, the timer shall be reset.

If the Link Layer connection supervision timer reaches 6 * *connInterval* before the connection is established (see Section 4.5), the connection shall be considered lost. This enables fast termination of connections that fail to establish.

Connection supervision timeout (*connSupervisionTimeout*) is a parameter that defines the maximum time between two received Data Packet PDUs before the connection is considered lost. The *connSupervisionTimeout* shall be a multiple of 10 ms in the range of 100 ms to 32.0 s and it shall be larger than (1 + *connSlaveLatency*) * *connInterval*.

If at any time in Connection State after the connection has been established and the timer reaches the *connSupervisionTimeout* value, the connection shall be considered lost.

If the connection is considered lost, the Link Layer shall not send any further packets. The Link Layer exits the Connection State and shall transition to the Standby State. The Host shall be notified of the loss of connection.

4.5.3 Connection Event Transmit Window

To allow the master to efficiently schedule connection events for multiple connections or other activities it may be involved in, the master has the flexibility to schedule the first connection event anchor point at a time of its choosing. The CONNECT_REQ PDU includes parameters to determine when the master can send its first packet in the Connection State to set the anchor point and when the slave must listen.

The CONNECT_REQ PDU includes three parameters used to determine the transmit window. The transmit window starts at *transmitWindowOffset* + 1.25 ms after the end of the CONNECT_REQ PDU, and the *transmitWindowSize* parameter shall define the size of the transmit window. The *connInterval* is used in the calculation of the maximum offset and size of the transmit window. *transmitWindowOffset* and *transmitWindowSize* are determined by the Link Layer.

The *transmitWindowOffset* shall be a multiple of 1.25 ms in the range of 0 ms to *connInterval*. The *transmitWindowSize* shall be a multiple of 1.25 ms in the range of 1.25 ms to the lesser of 10 ms and (*connInterval* - 1.25 ms*)*.

Therefore the start of the first packet will be no earlier than 1.25 ms + *transmitWindowOffset* and no later than 1.25 ms + *transmitWindowOffset* + *transmitWindowSize* after the end of the CONNECT_REQ PDU transmitted in the advertising channel.

4.5.4 Connection Setup – Master Role

After the initiator sends the CONNECT_REQ PDU the Link Layer is in Connection State in the Master Role. The master shall reset the Link Layer connection supervision timer $T_{LLconnSupervision}$. The Link Layer shall notify the Host that the connection has been created. The first connection event shall use the data channel index as specified in Section 1.4.1.

The master shall start to send the first packet within the transmit window as defined in Section 4.5.3. It is permitted that the master's first packet can extend beyond the transmit window.

The first packet sent in the Connection State by the master determines the anchor point for the first connection event, and therefore the timings of all future connection events in this connection.

The second connection event anchor point shall be *connInterval* after the first connection event anchor point. All the normal connection event transmission rules specified in Section 4.5.1 shall apply.

Two examples of the LL connection setup procedure timing from master's perspective are shown in Figure 4.11 and in Figure 4.12.

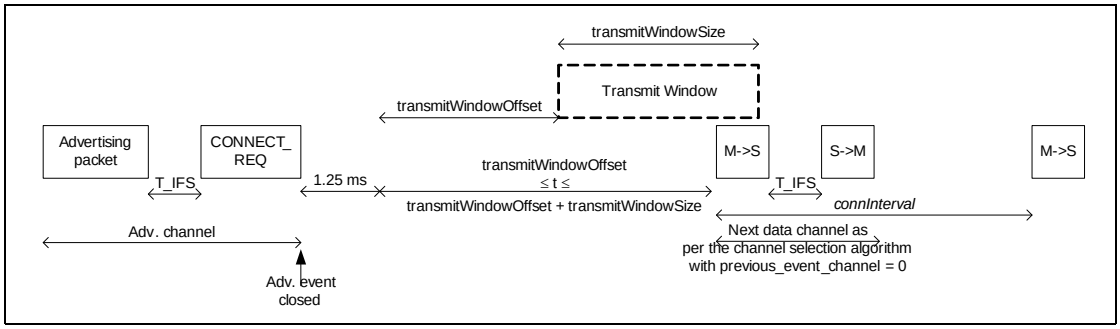


Figure 4.11: Master's view on LL connection setup with a non-zero transmitWindowOffset

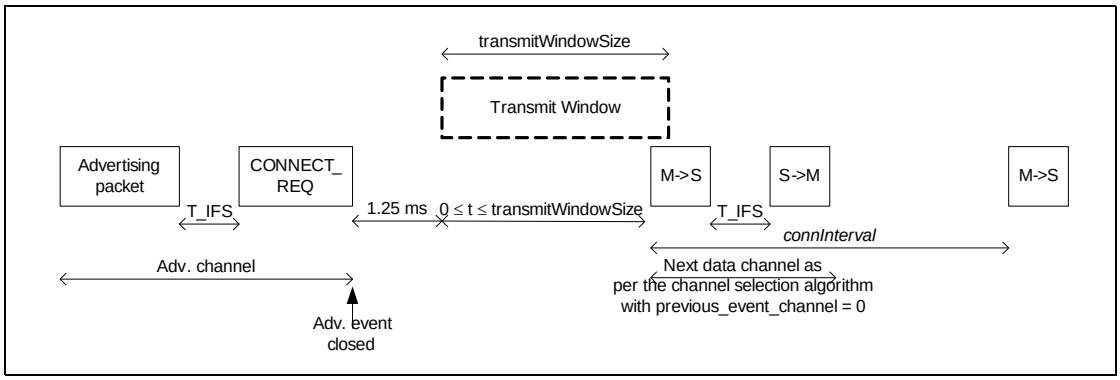


Figure 4.12: Master's view on LL connection setup with transmitWindowOffset set to zero

4.5.5 Connection Setup – Slave Role

After the advertiser receives a CONNECT_REQ PDU the Link Layer is in Connection State in the Slave Role. The slave shall reset the Link Layer connection supervision timer $T_{LLconnSupervision}$. The Link Layer shall notify the Host that the connection has been created. The first connection event shall use the data channel index as specified in Section 1.4.1.

The slave shall start to listen for the first packet within the transmit window as defined in Section 4.5.3. It is permitted that the master's first packet can extend beyond the transmit window, and therefore the slave must take this into account.

The first packet received, regardless of a valid CRC match (i.e., only the access code matches), in the Connection State by the slave determines the anchor point for the first connection event, and therefore the timings of all future connection events in this connection.

If a packet is not received in a transmit window, the slave shall attempt to receive a packet in a subsequent transmit window. A subsequent transmit window shall start *connInterval* after the start of the previous transmit window, with the same *transmitWindowSize*. The data channel index shall be the next data channel index as specified in Section 1.4.1. The *connEventCount* shall also be incremented by one.

An example of the procedure from the slave's perspective is shown in Figure 4.13 in which the slave fails to receive any part of the first packet (i.e., connEventCount = 0) from the master and acquires anchor point timing from the second packet (i.e., connEventCount = 1).

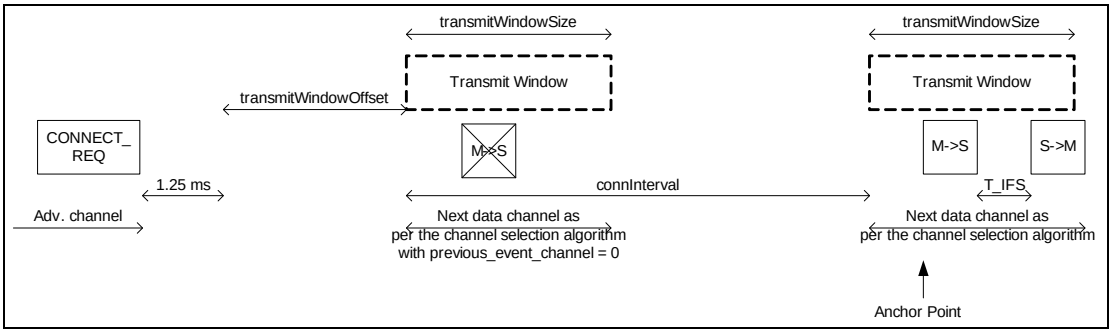


Figure 4.13: Slave closing LL connection setup in the second LL connection event

The slave shall be active in every connection event until the NESN is set to one in a master's packet at which point the slave can use slave subrating as defined in Section 4.5.1.

4.5.6 Closing Connection Events

The MD bit of the Header of the Data Channel PDU is used to indicate that the device has more data to send. If neither device has set the MD bit in their packets, the packet from the slave closes the connection event. If either or both of the devices have set the MD bit, the master may continue the connection event by sending another packet, and the slave should listen after sending its packet. If a packet is not received from the slave by the master, the master will close the connection event. If a packet is not received from the master by the slave, the slave will close the connection event.

Two consecutive packets received with an invalid CRC match within a connection event shall close the event.

MD bit usage is summarized in Table 4.2.

		Master	
		MD = 0	**MD = 1**
Slave	**MD = 0**	Master shall not send another packet, closing the connection event. Slave does not need to listen after sending its packet.	Master may continue the connection event. Slave should listen after sending its packet.
	MD = 1	Master may continue the connection event. Slave should listen after sending its packet.	Master may continue the connection event. Slave should listen after sending its packet.

Table 4.2: MD bit usage for closing connection events

4.5.7 Window Widening

Because of sleep clock accuracies (see Section 4.2.2), there is uncertainty in the slave of the exact timing of the master's anchor point. Therefore the slave is required to re-synchronize to the master's anchor point at each connection event where it listens for the master. If the slave receives a packet from the master regardless of a CRC match, the slave shall update its anchor point.

The slave calculates the time when the master will send the first packet of a connection event (*slaveExpectedAnchorPoint*) taking clock jittering, and in the case of connection setup or a connection parameter update the transmit window, into account. The slave shall also use the masters sleep clock accuracy (*masterSCA*) from the CONNECT_REQ PDU, together with its own sleep clock accuracy (*slaveSCA*) and the anchor point of the last connection event where it received a packet from the master (*timeSinceLastAnchor*) to calculate the time it needs to receive.

The increase in listening time is called the window widening. Assuming the clock inaccuracies are purely given in parts per million (ppm), it is calculated as follows:

windowWidening = ((*masterSCA* + *slaveSCA*) / 1000000) * *timeSinceLastAnchor*

During connection setup or during a connection parameter update, the slave should listen for windowWidening before the start of the transmit window and until windowWidening after the end of the transmitWindow for the master's anchor point.

At each subsequent connection event, the slave should listen for *windowWidening* before the start of the *slaveExpectedAnchorPoint* and until *windowWidening* after *slaveExpectedAnchorPoint* for the master's anchor point.

The *windowWidening* shall be smaller than ((*connInterval*/2) - T_IFS us). If the *windowWidening* reaches ((connInterval/2) - T_IFS us) in magnitude, the connection should be considered lost.

4.5.8 Data Channel Index Selection

4.5.8.1 Channel Classification

The master's Link Layer shall classify data channels into *used channels* (used for the connection) and *unused channels* (not used for the connection). This is called the channel map. The minimum number of used channels shall be 2.

The Host may provide channel classification information to the Link Layer. The Link Layer may use the information provided by the Host. The slave shall receive the channel map from the master in the CONNECT_REQ PDU. If the master changes the channel map it shall notify the slave as specified in Section 5.1.2.

4.5.8.2 Channel Selection

The channel selection algorithm consists of two stages: calculation of the unmapped channel index followed by mapping this index to a data channel index from the set of *used channels*.

The *unmappedChannel* and *lastUnmappedChannel* are the unmapped channel indices of two consecutive connection events. The *unmappedChannel* is the unmapped channel index for the current connection event. The *lastUnmappedChannel* is the unmapped channel index of the previous connection event. The *lastUnmappedChannel* shall be '0' for the first connection event of a connection.

At the start of a connection event, *unmappedChannel* shall be calculated using the following basic algorithm:

unmappedChannel = (*lastUnmappedChannel* + *hopIncrement*) mod 37

When a connection event closes, the *lastUnmappedChannel* shall be set to the value of the *unmappedChannel*.

If the *unmappedChannel* is a *used channel* according to the channel map, the channel selection algorithm shall use the *unmappedChannel* as the data channel index for the connection event.

If the *unmappedChannel* is an *unused channel* according to the channel map, the *unmappedChannel* shall be re-mapped to one of the used channels in the channel map using the following algorithm:

remappingIndex = *unmappedChannel* mod *numUsedChannels*

where *numUsedChannels* is the number of used channels in the channel map.

A remapping table is built that contains all the *used channels* in ascending order, indexed from zero. The *remappingIndex* is then used to select the data channel index for the connection event from the remapping table.

The complete procedure is as shown in Figure 4.14.

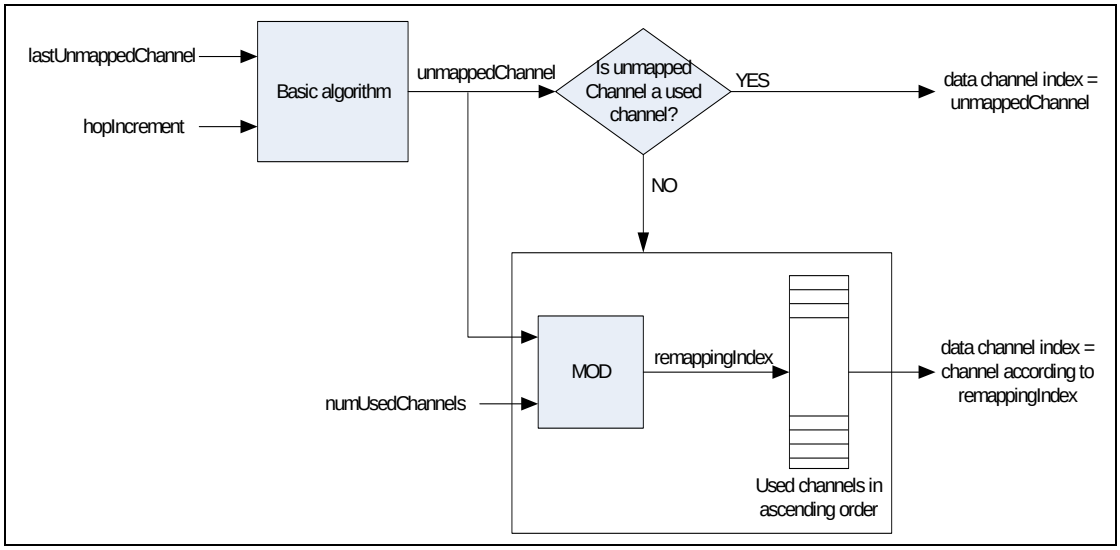


Figure 4.14: Block diagram of data channel selection algorithm

4.5.9 Acknowledgement and Flow Control

The Link Layer acknowledgement and flow control scheme shall be used in all Link Layer connections.

For each connection the Link Layer has two parameters, *transmitSeqNum* and *nextExpectedSeqNum*, each one bit in size. *transmitSeqNum* is used to identify packets sent by the Link Layer. *nextExpectedSeqNum* is used by the peer to either acknowledge the last Data Channel PDU sent, or to request resending of the last Data Channel PDU sent.

transmitSeqNum and *nextExpectedSeqNum* shall be set to zero upon entering the Connection State.

A new Data Channel PDU is a Data Channel PDU sent for the first time by the Link Layer. A last Data Channel PDU is a Data Channel PDU that is resent by the Link Layer. When resending a Data Channel PDU, the LLID field, the SN field and the payload of the sent Data Channel PDU shall be equal to those of the last Data Channel PDU sent by the Link Layer.

For each new Data Channel PDU that is sent, the SN bit of the Header shall be set to *transmitSeqNum*. If a Data Channel PDU is resent, then the SN bit shall not be changed.

Upon reception of a Data Channel PDU, the SN bit shall be compared to *nextExpectedSeqNum*. If the bits are different, then this is a resent Data Channel PDU, and *nextExpectedSeqNum* shall not be changed. If the bits are the same, then this is a new Data Channel PDU, and *nextExpectedSeqNum* may be incremented by one (see Section 4.5.9.1).

When a Data Channel PDU is sent, the NESN bit of the Header shall be set to *nextExpectedSeqNum*.

Upon receiving a Data Channel PDU, if the NESN bit of that Data Channel PDU is the same as *transmitSeqNum*, then the last sent Data Channel PDU has not been acknowledged and shall be resent. If the NESN bit of the Data Channel PDU is different from *transmitSeqNum*, then the last sent Data Channel PDU has been acknowledged, *transmitSeqNum* shall be incremented by one, and a new Data Channel PDU may be sent.

The above process is illustrated in Figure 4.15.

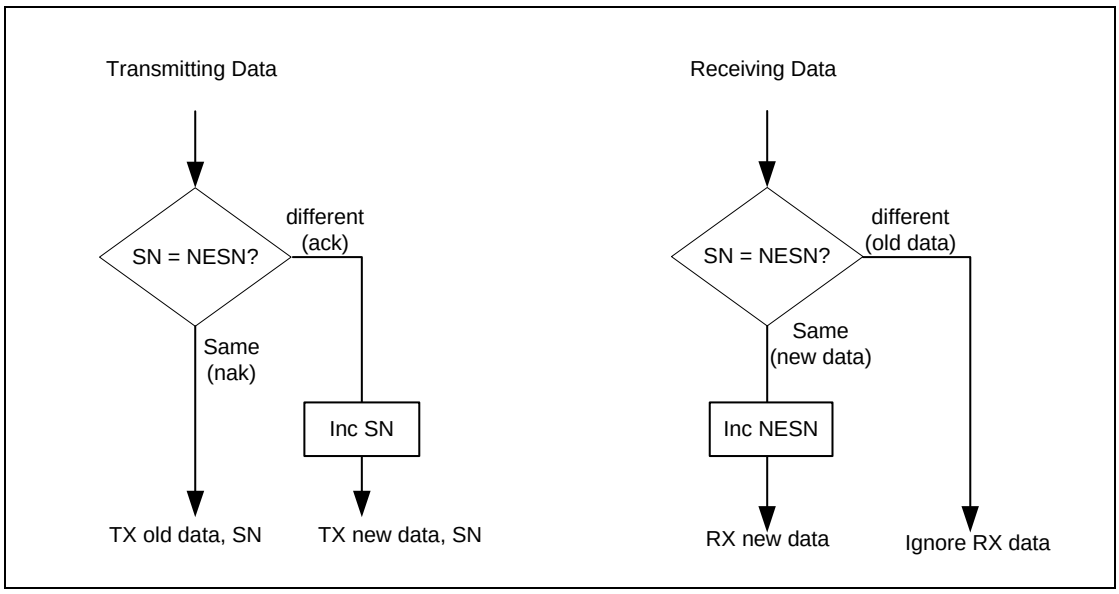


Figure 4.15: Transmit and Receive SN and NESN flow diagram

If a Data Channel PDU is received with an invalid CRC match, nextExpectedSeqNum shall not be changed; this means that the Data Channel PDU will not be acknowledged, causing the peer to resend the Data Channel PDU. Since the received Data Channel PDU has been rejected, the *nextExpectedSeqNum* from the peer device cannot be trusted, and therefore the last sent Data Channel PDU from this device was not acknowledged and must be retransmitted.

SN, NESN and MD bits shall be used from every received Data Channel PDU which has passed the CRC check. The Data Channel PDU payload shall be ignored on every received Data Channel PDU that has the same SN value as the previously received Data Channel PDU.

4.5.9.1 Flow Control

A Link Layer may not update nextExpectedSeqNum for reasons, including, but not limited to, lack of receive buffer space. This will cause the peer to resend the Data Channel PDU at a later time, thus enabling flow control.

4.6 FEATURE SUPPORT

When this information is sent from a Controller to a Host, a bit set to '0' indicates that the Link Layer Feature is not supported in this Controller; a bit set to '1' indicates that the Link Layer Feature is supported in this Controller.

When this information is sent from a Controller to a peer Controller, a bit set to '0' indicates that the Link Layer Feature shall not be used by the Controllers; a bit set to '1' indicates that the Link Layer Feature may be used by the Controllers.

The bit positions for each Link Layer Feature shall be as shown in Table 4.3. This table also shows if these bits are valid for the intended destination. If a bit is shown as not valid, using 'N', then this bit shall be ignored upon receipt.

Bit position	Link Layer Feature	Valid from Controller to Host	Valid from Host to Controller	Valid from Controller to Controller
0	LE Encryption	Y	Y	Y
1 – 63	RFU			

Table 4.3: FeatureSet field's bit mapping to Controller features

4.6.1 LE Encryption

A controller that supports LE Encryption shall support the following sections within this document:

- LL_ENC_REQ (Section 2.4.2.4)
- LL_ENC_RSP (Section 2.4.2.5)
- LL_START_ENC_REQ (Section 2.4.2.6)
- LL_START_ENC_RSP (Section 2.4.2.7)
- Encryption Start Procedure (Section 5.1.3.1)
- Encryption Pause Procedure (Section 5.1.3.2)

5 LINK LAYER CONTROL

The Link Layer Control Protocol (LLCP) is used to control and negotiate aspects of the operation of a connection between two Link Layers. This includes procedures for control of the connection, starting and pausing encryption and other link procedures.

Procedures have specific timeout rules as defined in Section 5.2. The Termination Procedure may be initiated at any time, even if any other Link Layer Control Procedure is currently active. For all other Link Layer Control Procedures, only one Link Layer Control Procedure shall be initiated in the Link Layer at a time per connection per device. A new Link Layer Control Procedure can be initiated only after a previous Link Layer Control Procedure has completed.

The prioritization of LL Control PDUs and LL Data PDUs is implementation specific. For example, a Host cannot assume that pending data will be sent when a termination of the link is requested without waiting for those data PDUs to be completed and indicated to the Host.

5.1 LINK LAYER CONTROL PROCEDURES

5.1.1 Connection Update Procedure

The Link Layer parameters for a connection (*connInterval*, *connSlaveLatency* and *connSupervisionTimeout*) may be updated after entering the Connection State. The master can update the connection parameters by sending an LL_CONNECTION_UPDATE_REQ PDU. The slave shall not send this PDU; the slave may request a change to the connection parameters using the L2CAP LE signaling channel.

The Link Layer of the master shall determine the *connInterval* from the interval range given by the Host ($connInterval_{min}$ and $connInterval_{max}$). The Link Layer shall indicate to the Host the selected interval value.

The Instant field of the LL_CONNECTION_UPDATE_REQ PDU shall be used to indicate the *connEventCount* when the updated parameters shall be applied; this is known as the instant. The master should allow a minimum of 6 connection events that the slave will be listening for before the instant occurs.

The connection interval used before the instant is known as $connInterval_{OLD}$. The connection interval contained in the LL_CONNECTION_UPDATE_REQ PDU and used at the instant and after, is known as $connInterval_{NEW}$.

The connection slave latency used before the instant is known as $connSlaveLatency_{OLD}$. The connection slave latency contained in the LL_CONNECTION_UPDATE_REQ PDU and used at the instant and after, is known as $connSlaveLatency_{NEW}$.

The connection supervision timeout used before the instant is known as $connSupervisionTimeout_{OLD}$. The connection supervision timeout contained in the LL_CONNECTION_UPDATE_REQ PDU and used at the instant and after, is known as $connSupervisionTimeout_{NEW}$. The connection supervision timer shall be reset at the instant.

For example, the interval between the previous connection event and the connection event at the instant will be $connInterval_{OLD}$. The interval between the connection event at the instant and the next connection event will be $connInterval_{NEW}$.

When a slave receives an LL_CONNECTION_UPDATE_REQ PDU where (Instant – connEventCount) modulo 65536 is less than 32767 and Instant is not equal to connEventCount, the slave shall listen to all the connection events until it has confirmation that the master has received its acknowledgement of the LL_CONNECTION_UPDATE_REQ PDU or connEventCount equals Instant. The slave shall also listen to the connection event where connEventCount equals Instant and the connection event before it.

When a slave receives an LL_CONNECTION_UPDATE_REQ PDU where (Instant – *connEventCount*) modulo 65536 is greater than or equal to 32767 (because the instant is in the past), the Link Layer of the slave shall consider the connection to be lost, the Link Layer shall exit the Connection State, and shall transition to the Standby State and shall notify the Host.

Note: The comparison of the *connEventCount* and the received Instant field is performed using modulo 65536 math (only values from 0 to 65535 are allowed), to handle the situation when the *connEventCount* field has wrapped.

The master may adjust the anchor point when deciding the timing of the first packet transmitted with new connection parameters. A transmit window is used, as defined in Section 4.5.3. The transmit window starts at $connInterval_{OLD}$ + *transmitWindowOffset* after the anchor point of the connection event before the instant. The *transmitWindowOffset* shall be a multiple of 1.25 ms in the range of 0 ms to $connInterval_{NEW}$. The *transmitWindowSize* shall be a multiple of 1.25 ms in the range of 1.25 ms to the lesser of 10 ms and ($connInterval_{NEW}$ - 1.25 ms).

The master shall start to send the first packet within the transmit window as defined in Section 4.5.3. It is permitted that the master's first packet can extend beyond the transmit window.

The first packet sent at the instant by the master determines the new anchor point for the connection events, and therefore the timings of all future connection events in this connection.

The next connection event anchor point shall be $connInterval_{NEW}$ after the connection event anchor point at the Instant. All the normal connection event transmission rules specified in Section 1.4.1, shall apply.

An example of the connection update procedures is shown in Figure 5.1.

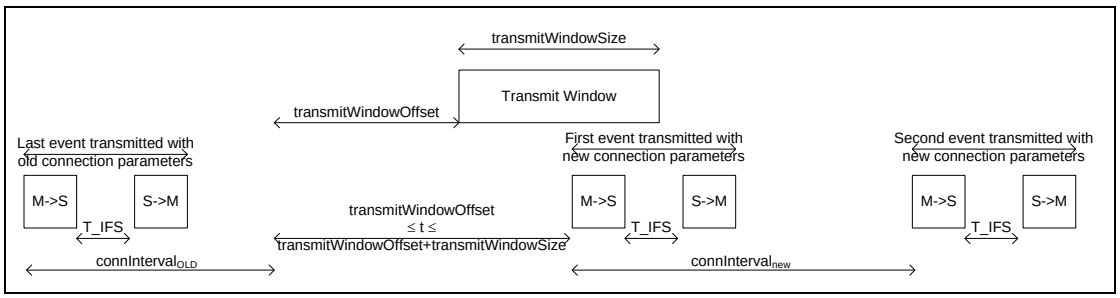


Figure 5.1: Connection event timing in the case of connection parameter update

At the start of the transmit window, the Link Layer shall reset $T_{LLconnSupervision}$.

If the Link Layer of the master transmits an LL_CONNECTION_UPDATE_REQ PDU autonomously, for example without being requested to by the Host, the Latency and Timeout parameters shall not be changed and shall remain the same as in the last LL_CONNECTION_UPDATE_REQ or CONNECT_REQ PDU, any of the other parameters (*transmitWindowSize*, *transmitWindowOffset*, *connInterval*, Instant) may be changed within the restrictions given above. Note: Autonomous updates can be used to change the anchor points to allow the master to change the scheduling of the connection due to other activities.

The Link Layer shall notify its Host if any of the three connection parameters have changed. If no connection parameters are changed, the Host would not be notified; this is called an anchor point move.

The procedure is complete when the instant has passed, and the new connection event parameters have been applied.

5.1.2 Channel Map Update Procedure

The Link Layer parameter for channel map (*channelMap*) may be updated after entering the Connection State. The master can update the channel map by sending an LL_CHANNEL_MAP_REQ PDU. The slave shall not send this PDU.

The Instant field of the LL_CHANNEL_MAP_REQ PDU shall be used to indicate the *connEventCount* when the $channelMap_{NEW}$ shall be applied; this is known as the instant. The master should allow a minimum of 6 connection events that the slave will be listening for before the instant occurs.

The channel map used before the instant is known as $channelMap_{OLD}$. The channel map contained in the LL_CHANNEL_MAP_REQ PDU and used at the instant and after, is known as $channelMap_{NEW}$.

When a slave receives an LL_CHANNEL_MAP_REQ PDU where (Instant – *connEventCount*) modulo 65536 is less than 32767, the slave shall listen to all the connection events until it has confirmation that the master has received its acknowledgement of the LL_CHANNEL_MAP_REQ PDU or *connEventCount* equals Instant.

When a slave receives an LL_CHANNEL_MAP_REQ where (Instant – *connEventCount*) modulo 65536 is greater than or equal to 32767 (because the instant is in the past), the Link Layer of the slave shall consider the connection to be lost and shall notify the Host.

Note: The comparison of the *connEventCount* and the received Instant field is performed using modulo 65536 math (only values from 0 to 65535 are allowed), to cope with the cases when the *connEventCount* field has wrapped.

When *connEventCount* is equal to the Instant field, the $channelMap_{NEW}$ shall be the current *channelMap*. The *lastUnmappedChannel* shall not be reset. If the *unmappedChannel* is an unused channel, then the $channelMap_{NEW}$ will be used when remapping. The only parameter that changes is the *channelMap*.

For example:

At connection set-up:

- initial $channelMap_{OLD}$: 0x1FFFFFFFFF (i.e., all channels enabled)
- initial *hopIncrement*: 10 (decimal)

An LL_CHANNEL_MAP_REQ PDU with the following parameters is then issued:

- Instant: 100 (decimal). Assume that no connection event count wrap-around occurred since the start of the connection.
- $channelMap_{NEW}$: 0x1FFFFFF7FF (i.e. all channels enabled except channel 11)

Channels used:

- *connEventCount* 99 --> data channel index 1 ($channelMap_{OLD}$)
- *connEventCount* 100 --> data channel index 12 (remapped from 11) ($channelMap_{NEW}$)
- *connEventCount* 101 --> data channel index 21 ($channelMap_{NEW}$)

The procedure is complete when the instant has passed, and the new channel map has been applied.

5.1.3 Encryption Procedure

The Link Layer, upon request from the Host, can enable the encryption of packets after entering the Connection State.

If the connection is not encrypted, the Link Layer shall only use the encryption start procedure.

If the connection is encrypted, the Link Layer shall first use the encryption pause procedure followed by the encryption start procedure.

5.1.3.1 Encryption Start Procedure

To enable encryption, two parameters must be exchanged, IV and SKD. Both are composed of two parts, a master part and a slave part, and exchanged in LL_ENC_REQ and LL_ENC_RSP PDUs. After these are exchanged, and the Host has notified the Link Layer of the Long Term Key to be used on this connection, encryption can be started using a three way handshake, using LL_START_ENC_REQ and LL_START_ENC_RSP PDUs.

To start encryption, the Link Layer of the master shall generate the master's part of the initialization vector (IVm) and the master's part of the session key diversifier (SKDm).

The Link Layer of the master shall finalize the sending of the current Data Channel PDU and may finalize the sending of additional Data Channel PDUs queued in the Controller. After these Data Channel PDUs are acknowledged, the Link Layer of the master shall only send Empty PDUs or LL_ENC_REQ, LL_START_ENC_REQ, LL_START_ENC_RSP, LL_TERMINATE_IND or LL_REJECT_IND PDUs.

The Link Layer of the master shall then send an LL_ENC_REQ PDU; the Rand and EDIV fields are provided by the Host.

If encryption is not supported by the Link Layer of the slave, the Link Layer of the slave shall send an LL_REJECT_IND PDU with the error code set to "Unsupported Remote Feature / Unsupported LMP Feature" (0x1A). The Link Layer of the master receiving the LL_REJECT_IND PDU shall notify the Host. The Link Layer of the master can now send LL Data Packets and LL Control Packets; these packets will not be encrypted. This procedure is complete in the master when the master receives the LL_REJECT_IND PDU from the slave. The procedure is complete in the slave when the acknowledgement for the LL_REJECT_IND PDU is received from the master.

Otherwise, when the Link Layer of the slave receives an LL_ENC_REQ PDU it shall generate the slave's part of the initialization vector (IVs) and the slave's part of the session key diversifier (SKDs), and notify the Host with the Rand and EDIV fields.

The Link Layer of the slave shall finalize the sending of the current Data Channel PDU and may finalize the sending of additional Data Channel PDUs queued in the Controller. After these Data Channel PDUs are acknowledged, the Link Layer of the slave is only allowed to send Empty PDUs or LL_ENC_RSP, LL_START_ENC_REQ, LL_START_ENC_RSP, LL_TERMINATE_IND or LL_REJECT_IND PDUs.

The Link Layer of the slave shall then send an LL_ENC_RSP PDU.

Each Link Layer shall combine the initialization vector parts and session key diversifier parts in the following manner:

SKD = SKDm || SKDs

IV = IVm || IVs

The SKDm is concatenated with the SKDs. The least significant octet of SKDm becomes the least significant octet of SKD. The most significant octet of SKDs becomes the most significant octet of SKD.

The IVm is concatenated with the IVs. The least significant octet of IVm becomes the least significant octet of IV. The most significant octet of IVs becomes the most significant octet of IV.

The Long Term Key is provided by the Host to the Link Layer in the master and slave, and one of the following three actions shall occur:

- If this procedure is being performed after a Pause Encryption Procedure, and the Host does not provide a Long Term Key, the slave shall perform the Termination Procedure with the error code “PIN or key Missing.”
- If the Host does not provide a Long Term Key, either because the event to the Host was masked out or if the Host indicates that a key is not available, the slave shall send an LL_REJECT_IND PDU with the error code set to “PIN or key Missing.” Upon receiving an LL_REJECT_IND PDU, the Link Layer shall notify the Host. The Link Layer can now send LL Data PDUs and LL Control PDUs; these packets will not be encrypted. This procedure is complete in the master when the master receives the LL_REJECT_IND PDU from the slave. The procedure is completed in the slave when the acknowledgement has been received for the LL_REJECT_IND PDU from the master.
- If the Host does provide a Long Term Key, the Link Layer of the slave shall respond to the LL_ENC_REQ PDU from the master with an LL_ENC_RSP PDU. The Link Layer shall also calculate *sessionKey* using the encryption engine with LTK as the key, and SKD as the plain text input. *sessionKey* shall be set to the output of the encryption engine.

The *sessionKey* shall be used as the key for the encryption engine for all encrypted packets.

After *sessionKey* has been calculated, the Link Layer of the slave shall send an LL_START_ENC_REQ PDU. This packet shall be sent unencrypted, and the Link Layer shall be set up to receive an encrypted packet in response.

When the Link Layer of the master receives an LL_START_ENC_REQ PDU it shall send an LL_START_ENC_RSP PDU. This PDU shall be sent encrypted and set up to receive encrypted.

When the Link Layer of the slave receives an LL_START_ENC_RSP PDU it shall transmit an LL_START_ENC_RSP PDU. This packet shall be sent encrypted.

When the Link Layer of the master receives the LL_START_ENC_RSP PDU, the connection is encrypted. The Link Layer can now send LL Data PDUs and LL Control PDUs; these PDUs will be encrypted.

The Link Layers shall notify the Hosts that the connection is encrypted.

The procedure is complete in the master when the master receives the LL_START_ENC_RSP PDU from the slave. The procedure is complete in the slave when the slave receives the LL_START_ENC_RSP PDU from the master.

5.1.3.2 Encryption Pause Procedure

To enable a new encryption key to be used without disconnecting the link, encryption must be disabled and then enabled again. During the pause, data PDUs shall not be sent unencrypted to protect the data.

The Link Layer of the master shall finalize the sending of the current Data Channel PDU and may finalize the sending of additional Data Channel PDUs queued in the Controller. After these Data Channel PDUs are acknowledged, the Link Layer of the master shall only send Empty PDUs or LL_PAUSE_ENC_REQ or LL_TERMINATE_IND PDUs.

The Link Layer of the master shall then send an LL_PAUSE_ENC_REQ PDU.

When the Link Layer of the slave receives an LL_PAUSE_ENC_REQ PDU it shall finalize the sending of the current Data Channel PDU and may finalize the sending of additional Data Channel PDUs queued in the Controller. After these Data Channel PDUs are acknowledged, the Link Layer of the slave is only allowed to send Empty PDUs or LL_PAUSE_ENC_RSP or LL_TERMINATE_IND PDUs.

The Link Layer of the slave shall then send an LL_PAUSE_ENC_RSP PDU. This packet shall be sent encrypted, and Link Layer shall be set up to receive unencrypted.

When the Link Layer of the master receives an LL_PAUSE_ENC_RSP PDU it shall set up to send and receive unencrypted. It shall then send an LL_PAUSE_ENC_RSP PDU to the slave unencrypted.

When the Link Layer of the slave receives an LL_PAUSE_ENC_RSP PDU it shall set up to also send unencrypted.

The encryption start procedure shall now be used to re-enable encryption using a new session key.

5.1.4 Feature Exchange Procedure

The Link Layer parameter for the current supported feature set (featureSet) may be exchanged after entering the Connection State. The master can initiate this procedure with an LL_FEATURE_REQ PDU, and the slave responds with an LL_FEATURE_RSP PDU. The slave cannot initiate this procedure.

The featureSet information may be cached. A Link Layer should not request this information on every connection if the information has been cached for this device. Cached information for a device may not be authoritative, and therefore an implementation must be able to accept the LL_UNKNOWN_RSP PDU if use of a feature is attempted that is not currently supported or used by the peer.

$featureSet_M$ is the feature capabilities of the Link Layer of the master. When the Link Layer of the master sends an LL_FEATURE_REQ PDU the Feature-Set field shall be set to $featureSet_M$.

$featureSet_S$ is the feature capabilities of the Link Layer of the Slave.

The $featureSet_{USED}$ is the logical AND of $featureSet_M$ and $featureSet_S$. When the Link Layer of the slave sends an LL_FEATURE_RSP PDU the FeatureSet field shall be set to $featureSet_{USED}$.

The Link Layer of the master sends an LL_FEATURE_REQ PDU. This can be sent on request from the Host or autonomously.

When the Link Layer of the slave receives an LL_FEATURE_REQ PDU it shall send an LL_FEATURE_RSP PDU. The Link Layer of the slave shall only use procedures that are indicated in $featureSet_{USED}$.

When the Link Layer of the master receives an LL_FEATURE_RSP PDU it shall only use procedures that are indicated in $featureSet_{USED}$.

An example of feature exchange is shown in Figure 5.2.

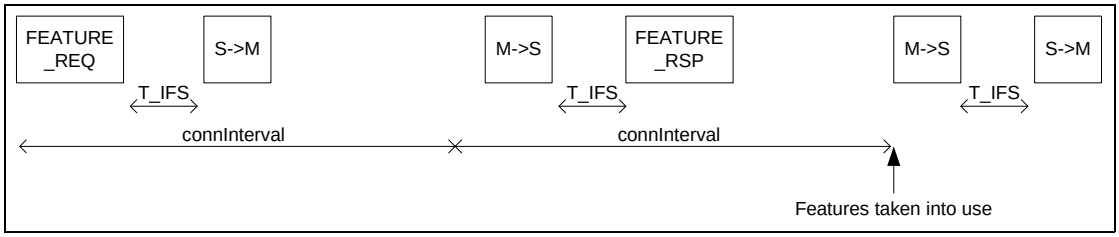


Figure 5.2: Feature Exchange Procedure

The procedure is complete in the master when the master receives the LL_FEATURE_RSP PDU from the slave.

5.1.5 Version Exchange

The Link Layer parameters for version information (*companyID*, *subVerNum*, *linkLayerVer,* as defined in Section 2.4.2.13) may be exchanged after entering the Connection State. Either the Link Layer of the master or slave can initiate this procedure by sending an LL_VERSION_IND PDU. This procedure should be used when requested by the Host. This procedure may be initiated autonomously by the Link Layer.

The Link Layer shall only queue for transmission a maximum of one LL_VERSION_IND PDU during a connection.

If the Link Layer receives an LL_VERSION_IND PDU and has not already sent an LL_VERSION_IND then the Link Layer shall send an LL_VERSION_IND PDU to the peer device.

If the Link Layer receives an LL_VERSION_IND PDU and has already sent an LL_VERSION_IND PDU then the Link Layer shall not send another LL_VERSION_IND PDU to the peer device.

The procedure has completed when an LL_VERSION_IND PDU has been received from the peer device.

5.1.6 Termination Procedure

This procedure is used for voluntary termination of a connection while in the Connection State. Voluntary termination occurs when the Host requests the Link Layer to terminate the connection. Either the Link Layer of the master or slave can initiate this procedure by sending an LL_TERMINATE_IND PDU. The termination procedure is not used in the event of the loss of the connection, for example after link supervision timeout or after a procedure timeout.

The Link Layer shall start a timer, $T_{terminate}$, when the LL_TERMINATE_IND PDU has been queued for transmission. The initiating Link Layer shall send LL_TERMINATE_IND PDUs until an acknowledgement is received or until the

timer, $T_{terminate}$, expires. The initial value for $T_{terminate}$ shall be set to value of the *connSupervisionTimeout*.

When the Link Layer receives an LL_TERMINATE_IND PDU it shall send the acknowledgement, exit the Connection State and shall transition to the Standby State.

The procedure has completed when the acknowledgement has been received.

5.2 PROCEDURE RESPONSE TIMEOUT

This section specifies procedure timeout rules that shall be applied to all the Link Layer control procedures specified in Section 5.1, except for the Connection Update and Channel Map Update procedures for which there are no timeout rules.

To be able to detect a non-responsive Link Layer Control Procedure, both the master and the slave shall use a procedure response timeout timer, T_{PRT}. Upon the initiation of a procedure, the procedure response timeout timer shall be reset and started.

Each LL Control PDU that is queued for transmission resets the procedure response timeout timer.

When the procedure completes, the procedure response timeout timer shall be stopped.

If the procedure response timeout timer reaches 40 seconds, the connection is considered lost. The Link Layer exits the Connection State and shall transition to the Standby State. The Host shall be notified of the loss of connection.