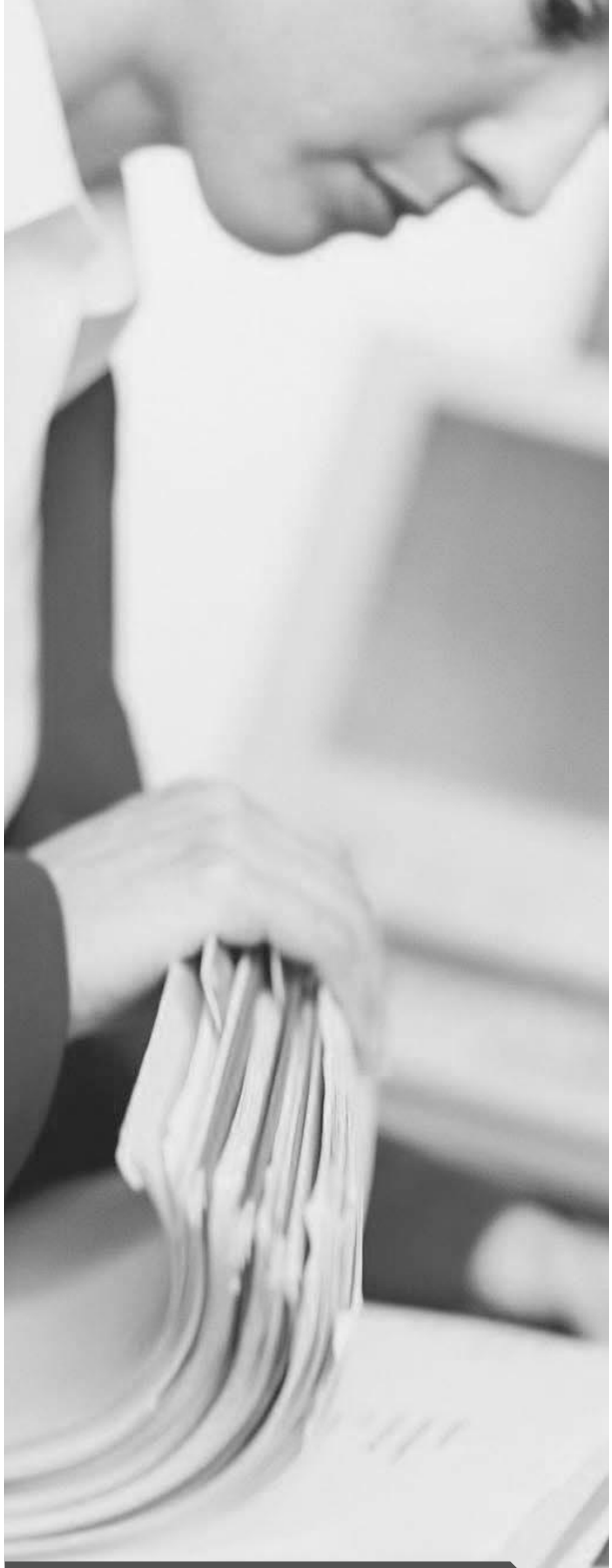


Specification of the Bluetooth System

Wireless connections made easy

Master Table of Contents & Compliance Requirements

Covered Core Package version: 1.2
Current Master TOC issued:
05 November 2003

Bluetooth®

Revision History

The Revision History is shown in the “Appendix” on page 53[vol. 0].

Contributors

The persons who contributed to this specification are listed in “Appendix” on page 53[vol. 0].

Web Site

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

Disclaimer and Copyright Notice

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

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 2

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

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

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

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

Copyright © 1999, 2000, 2001, 2002, 2003

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 4



MASTER TABLE OF CONTENTS

Bluetooth Specification
Including Core v1.2

This table of contents (TOC) covers the entire Bluetooth Specification.
In addition, each volume has a TOC and each part of a volume is preceded by a detailed TOC.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 6

MASTER TOC FOR THE BLUETOOTH SPECIFICATION

In the following table:

- The TOC for each Volume starts at the top of a page.
- The Volume No. is written in red and followed by the name of the Volume.

Note: Each Volume is a self contained book which is published and updated separately and is equipped with a TOC of its own. However, this Master TOC is also revised as soon as any of the other Volumes are updated.

- A Volume cover one or more Parts (A, B, etc.), each Part can be viewed independently and has its own TOC.

Red or blue text on the following pages indicates hypertext links that will take you directly to the indicated section, on condition that you have access to a complete specification.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 7

Petitioner Kiosoft Exhibit 1011 Part 1
Page 8

Specification Volume 0
Master Table of Contents & Compliance Requirements

Part A
MASTER TABLE OF CONTENTS

Master TOC for the Bluetooth Specification 7

Part B
BLUETOOTH COMPLIANCE REQUIREMENTS

Contents 43

1 Scope 45

2 Definitions.....47

3 Legal Aspects.....49

4 Introduction to the Bluetooth Qualification Program 51

Petitioner Kiosoft Exhibit 1011 Part 1
Page 9

Part C
APPENDIX

Contents **55**

1 Revision History **57**
1.1 [vol 0] Master TOC & Compliance Requirements 57
1.1.1 Bluetooth Compliance Requirements 57
1.2 [Vol 1] Architecture & Terminology Overview 57
1.3 [Vol 2 & 3] Core System Package 58

2 Contributors **59**
2.1 [vol 0] Master TOC & Compliance Requirements 59
2.1.1 Part B: Bluetooth Compliance Requirements 59
2.2 [vol 1] Architecture &Terminology Overview 59
2.2.1 Part A: Architectural Overview 59
2.2.2 Part B: Acronyms & Abbreviations 60
2.2.3 Part C: Changes from Bluetooth Specification v1.1 60
2.3 [Vol 2] Core System Package, Controller 61
2.3.1 Part A: Radio Specification 61
2.3.2 Part B: Baseband Specification 62
2.3.3 Part C: Link Manager Protocol 65
2.3.4 Part D: Error Codes 66
2.3.5 Part E: Bluetooth Host Controller Interface Functional Specification 67
2.3.6 Part F: Message Sequence Charts 68
2.3.7 Part G: Sample Data 69
2.3.8 Part H: Security Specification 69
2.4 [Vol 3] Core System Package, Host 70
2.4.1 Part A: Logical Link Control and Adaptation Protocol Specification 70
2.4.2 Part B: Service Discovery Protocol (SDP) 71
2.4.3 Part C Generic Access Profile 72
2.4.4 Part D: Test Support 72

Petitioner Kiosoft Exhibit 1011 Part 1
Page 10

Specification Volume 1
Architecture & Terminology Overview

Table of Contents **5**

Part A
ARCHITECTURE

Contents **11**

1 General Description **13**
1.1 Overview of Operation 13
1.2 Nomenclature 15

2 Core System Architecture **19**
2.1 Core Architectural Blocks 22
2.1.1 Channel manager 22
2.1.2 L2CAP resource manager 22
2.1.3 Device manager 23
2.1.4 Link manager 23
2.1.5 Baseband resource manager 23
2.1.6 Link controller 24
2.1.7 RF 24

3 Data Transport Architecture **25**
3.1 Core Traffic Bearers 26
3.1.1 Framed data traffic 27
3.1.2 Unframed data traffic 28
3.1.3 Reliability of traffic bearers 28
3.2 Transport Architecture Entities 30
3.2.1 Bluetooth generic packet structure 30
3.3 Physical Channels 32
3.3.1 Basic piconet channel 33
3.3.2 Adapted piconet channel 34
3.3.3 Inquiry scan channel 35
3.3.4 Page scan channel 36
3.4 Physical Links 37
3.4.1 Links supported by the basic and adapted piconet physical channel 38
3.4.2 Links supported by the scanning physical channels 39

Petitioner Kiosoft Exhibit 1011 Part 1
Page 11

3.5 Logical Links and Logical Transports....39
3.5.1 Casting....41
3.5.2 Scheduling and acknowledgement scheme....41
3.5.3 Class of data....42
3.5.4 Asynchronous connection-oriented (ACL)....42
3.5.5 Synchronous connection-oriented (SCO)....43
3.5.6 Extended synchronous connection-oriented (eSCO)....44
3.5.7 Active slave broadcast (ASB)....44
3.5.8 Parked slave broadcast (PSB)....45
3.5.9 Logical links....46
3.5.10 ACL Control Logical Link (ACL-C)....47
3.5.11 User Asynchronous/Isochronous Logical Link (ACL-U) 47
3.5.12 User Synchronous/Extended Synchronous Logical Links (SCO-S/eSCO-S)....47
3.6 L2CAP Channels....48

4 Communication Topology....49

4.1 Piconet Topology....49
4.2 Operational Procedures and Modes....51
4.2.1 Inquiry (Discovering) Procedure....51
4.2.2 Paging (Connecting) Procedure....52
4.2.3 Connected mode....52
4.2.4 Hold mode....53
4.2.5 Sniff mode....53
4.2.6 Parked state....54
4.2.7 Role switch procedure....54

Petitioner Kiosoft Exhibit 1011 Part 1
Page 12

Part B
ACRONYMS & ABBREVIATIONS

1 List of Acronyms and Abbreviations....57

2 Abbreviations of the Specification Names 65

Part C
CHANGES FROM BLUETOOTH SPECIFICATION V 1.1

Contents 69

1 New Features....71

2 Changes in Wording 73

2.1 IEEE Language Update 73

2.1.1 Shall 74

2.1.2 Must....74

2.1.3 Will....74

2.1.4 Should 74

2.1.5 May....75

2.1.6 Can....75

2.2 Nomenclature Changes 75

3 Structure Changes 77

4 Deprecated Specifications 79

4.1 Deprecated Features 79

4.2 Deprecated Profiles....79

Petitioner Kiosoft Exhibit 1011 Part 1
Page 13

Specification Volume 2
Core System Package
[Controller volume]

Table of Contents 5

Part A
RADIO SPECIFICATION

Contents 27

1 Scope 29

2 Frequency Bands and Channel Arrangement 31

3 Transmitter Characteristics 33
3.1 Modulation Characteristics 34
3.2 Spurious Emissions 35
3.2.1 In-band spurious emission 35
3.3 Radio Frequency Tolerance 36

4 Receiver Characteristics 37
4.1 Actual Sensitivity Level 37
4.2 Interference Performance 37
4.3 Out-of-Band Blocking 38
4.4 Intermodulation Characteristics 38
4.5 Maximum Usable Level 39
4.6 Receiver Signal Strength Indicator 39
4.7 Reference Signal Definition 39

5 Appendix A 41

6 Appendix B 43

Petitioner Kiosoft Exhibit 1011 Part 1
Page 14

Part B
BASEBAND SPECIFICATION

Contents

1 General Description 53
1.1 Bluetooth Clock 54
1.2 Bluetooth Device Addressing 55
1.2.1 Reserved addresses 55
1.3 Access Codes 56
2 Physical Channels 57
2.1 Physical Channel Definition 58
2.2 Basic Piconet Physical Channel 58
2.2.1 Master-slave definition 58
2.2.2 Hopping characteristics 59
2.2.3 Time slots 59
2.2.4 Piconet clocks 60
2.2.5 Transmit/receive timing 60
2.3 Adapted Piconet Physical Channel 63
2.3.1 Hopping characteristics 63
2.4 Page Scan Physical Channel 64
2.4.1 Clock estimate for paging 64
2.4.2 Hopping characteristics 64
2.4.3 Paging procedure timing 65
2.4.4 Page response timing 66
2.5 Inquiry Scan Physical Channel 68
2.5.1 Clock for inquiry 68
2.5.2 Hopping characteristics 68
2.5.3 Inquiry procedure timing 68
2.5.4 Inquiry response timing 68
2.6 Hop Selection 70
2.6.1 General selection scheme 70
2.6.2 Selection kernel 74
2.6.3 Adapted hop selection kernel 77
2.6.4 Control word 78
3 Physical Links 83
3.1 Link Supervision 83

4 Logical Transports 85
4.1 General 85
4.2 Logical Transport Address (LT_ADDR) 85
4.3 Synchronous Logical Transports 86
4.4 Asynchronous Logical Transport 86
4.5 Transmit/Receive Routines 87
4.5.1 TX Routine 87
4.5.2 RX routine 90
4.5.3 Flow control 91
4.6 Active Slave Broadcast Transport 92
4.7 Parked Slave Broadcast Transport 93
4.7.1 Parked member address (PM_ADDR) 93
4.7.2 Access request address (AR_ADDR) 93
5 Logical Links 95
5.1 Link Control Logical Link (LC) 95
5.2 ACL Control Logical Link (ACL-C) 95
5.3 User Asynchronous/Isochronous Logical Link (ACL-U) 95
5.3.1 Pausing the ACL-U logical link 96
5.4 User Synchronous Data Logical Link (SCO-S) 96
5.5 User Extended Synchronous Data Logical Link (eSCO-S) 96
5.6 Logical Link Priorities 96
6 Packets 97
6.1 General Format 97
6.2 Bit Ordering 97
6.3 Access Code 98
6.3.1 Access code types 98
6.3.2 Preamble 99
6.3.3 Sync word 99
6.3.4 Trailer 102
6.4 Packet Header 103
6.4.1 LT_ADDR 103
6.4.2 TYPE 103
6.4.3 FLOW 104
6.4.4 ARQN 104
6.4.5 SEQN 104
6.4.6 HEC 104

Petitioner Kiosoft Exhibit 1011 Part 1
Page 16

6.5 Packet Types 105
6.5.1 Common packet types 106
6.5.2 SCO packets 109
6.5.3 eSCO packets 110
6.5.4 ACL packets 111
6.6 Payload Format 112
6.6.1 Synchronous data field 112
6.6.2 Asynchronous data field 113
6.7 Packet Summary 116

7 Bitstream Processing 117
7.1 Error Checking 118
7.1.1 HEC generation 118
7.1.2 CRC generation 119
7.2 Data Whitening 121
7.3 Error Correction 122
7.4 FEC Code: Rate 1/3 122
7.5 FEC Code: Rate 2/3 123
7.6 ARQ Scheme 124
7.6.1 Unnumbered ARQ 124
7.6.2 Retransmit filtering 127
7.6.3 Flushing payloads 130
7.6.4 Multi-slave considerations 130
7.6.5 Broadcast packets 130

8 Link Controller Operation 133
8.1 Overview of States 133
8.2 Standby State 134
8.3 Connection Establishment Substates 134
8.3.1 Page scan substate 134
8.3.2 Page substate 136
8.3.3 Page response substates 139
8.4 Device Discovery Substates 143
8.4.1 Inquiry scan substate 144
8.4.2 Inquiry substate 145
8.4.3 Inquiry response substate 147
8.5 Connection State 148

Petitioner Kiosoft Exhibit 1011 Part 1
Page 17

8.6 Active Mode 149
8.6.1 Polling in the active mode 150
8.6.2 SCO 150
8.6.3 eSCO 151
8.6.4 Broadcast scheme 154
8.6.5 Role switch 155
8.6.6 Scatternet 157
8.6.7 Hop sequence switching 158
8.6.8 Channel classification and channel map selection 161
8.6.9 Power Management 162
8.7 Sniff Mode 163
8.7.1 Sniff Transition Mode 164
8.8 Hold Mode 165
8.9 Park State 165
8.9.1 Beacon train 166
8.9.2 Beacon access window 168
8.9.3 Parked slave synchronization 169
8.9.4 Parking 170
8.9.5 Master-initiated unparking 171
8.9.6 Slave-initiated unparking 171
8.9.7 Broadcast scan window 172
8.9.8 Polling in the park state 172

9 Audio 173
9.1 LOG PCM CODEC 173
9.2 CVSD CODEC 173
9.3 Error Handling 176
9.4 General Audio Requirements 176
9.4.1 Signal levels 176
9.4.2 CVSD audio quality 176

10 List of Figures 177

11 List of Tables 181

Appendix A:
General Audio Recommendations 182

Appendix B:
Timers 185

Appendix C:
Recommendations for AFH Operation in Park, Hold and Sniff 187

Petitioner Kiosoft Exhibit 1011 Part 1
Page 18

Part C
LINK MANAGER PROTOCOL

Contents **191**

1 Introduction **195**

2 General Rules **197**

2.1 Message Transport 197
2.2 Synchronization 197
2.3 Packet Format 198
2.4 Transactions 199
2.4.1 LMP Response Timeout 200
2.5 Error Handling 200
2.5.1 Transaction collision resolution 201
2.6 Procedure Rules 201
2.7 General Response Messages 202
2.8 LMP Message Constraints 202

3 Device Features **203**

3.1 General Description 203
3.2 Feature Definitions 203
3.3 Feature Mask Definition 207
3.4 Link Manager Interoperability policy 208

4 Procedure Rules **209**

4.1 Connection Control 209
4.1.1 Connection establishment 209
4.1.2 Detach 210
4.1.3 Power control 211
4.1.4 Adaptive frequency hopping 213
4.1.5 Channel classification 216
4.1.6 Link supervision 218
4.1.7 Channel quality driven data rate change (CQDDR) 219
4.1.8 Quality of service (QoS) 220
4.1.9 Paging scheme parameters 222
4.1.10 Control of multi-slot packets 223
4.2 Security 224
4.2.1 Authentication 224
4.2.2 Pairing 226
4.2.3 Change link key 229
4.2.4 Change current link key type 230
4.2.5 Encryption 232
4.2.6 Request supported encryption key size 236

4.3 Informational Requests 237
4.3.1 Timing accuracy 237
4.3.2 Clock offset 238
4.3.3 LMP version 238
4.3.4 Supported features 239
4.3.5 Name request 241
4.4 Role Switch 242
4.4.1 Slot offset 242
4.4.2 Role switch 243
4.5 Modes of Operation 245
4.5.1 Hold mode 245
4.5.2 Park state 247
4.5.3 Sniff mode 253
4.6 Logical Transports 256
4.6.1 SCO logical transport 256
4.6.2 eSCO logical transport 259
4.7 Test Mode 264
4.7.1 Activation and deactivation of test mode 264
4.7.2 Control of test mode 265
4.7.3 Summary of test mode PDUs 266

5 Summary 269
5.1 PDU Summary 269
5.2 Parameter Definitions 277
5.3 Default Values 284

6 List of Figures 285

7 List of Tables 289

Petitioner Kiosoft Exhibit 1011 Part 1
Page 20

Part D
ERROR CODES

Contents **293**

1 Overview of Error Codes **295**

1.1 Usage Descriptions 295
1.2 HCI Command Errors 295
1.3 List of Error Codes 296

2 Error Code Descriptions **299**

2.1 Unknown HCI Command (0X01) 299
2.2 Unknown Connection Identifier (0X02) 299
2.3 Hardware Failure (0X03) 299
2.4 Page Timeout (0X04) 299
2.5 Authentication Failure (0X05) 299
2.6 PIN Missing (0X06) 299
2.7 Memory Capacity Exceeded (0X07) 299
2.8 Connection Timeout (0X08) 300
2.9 Connection Limit Exceeded (0X09) 300
2.10 Synchronous Connection Limit to a Device Exceeded (0X0A) 300
2.11 ACL Connection Already Exists (0X0B) 300
2.12 Command Disallowed (0X0C) 300
2.13 Connection Rejected due to Limited Resources (0X0D) 300
2.14 Connection Rejected due to Security Reasons (0X0E) 300
2.15 Connection Rejected due to Unacceptable BD_ADDR (0X0F) 301
2.16 Connection Accept Timeout Exceeded (0X10) 301
2.17 Unsupported Feature or Parameter Value (0X11) 301
2.18 Invalid HCI Command Parameters (0X12) 301
2.19 Remote User Terminated Connection (0X13) 301
2.20 Remote Device Terminated Connection due to Low Resources (0X14) 302
2.21 Remote Device Terminated Connection due to Power Off (0X15) 302
2.22 Connection Terminated by Local Host (0X16) 302
2.23 Repeated Attempts (0X17) 302
2.24 Pairing not Allowed (0X18) 302
2.25 Unknown LMP PDU (0X19) 302
2.26 Unsupported Remote Feature (0X1A) 302
2.27 SCO Offset Rejected (0X1B) 302
2.28 SCO Interval Rejected (0X1C) 303
2.29 SCO Air Mode Rejected (0X1D) 303
2.30 Invalid LMP Parameters (0X1E) 303
2.31 Unspecified Error (0X1F) 303
2.32 Unsupported LMP Parameter Value (0X20) 303

Petitioner Kiosoft Exhibit 1011 Part 1
Page 21

2.33 Role Change Not Allowed (0X21) 303
2.34 LMP Response Timeout (0X22) 303
2.35 LMP Error Transaction Collision (0X23) 304
2.36 LMP PDU Not Allowed (0X24) 304
2.37 Encryption Mode Not Acceptable (0X25) 304
2.38 Link Key Can Not be Changed (0X26) 304
2.39 Requested Qos Not Supported (0X27) 304
2.40 Instant Passed (0X28) 304
2.41 Pairing with Unit Key Not Supported (0X29) 304
2.42 Different Transaction Collision (0x2a) 304
2.43 QoS Unacceptable Parameter (0X2C) 304
2.44 QoS Rejected (0X2D) 305
2.45 Channel Classification Not Supported (0X2E) 305
2.46 Insufficient Security (0X2F) 305
2.47 Parameter out of Mandatory Range (0X30) 305
2.48 Role Switch Pending (0X32) 305
2.49 Reserved Slot Violation (0X34) 305
2.50 Role Switch Failed (0X35) 305

Petitioner Kiosoft Exhibit 1011 Part 1
Page 22

Part E
HOST CONTROLLER INTERFACE FUNCTIONAL SPECIFICATION

Contents **309**

1 Introduction **317**
1.1 Lower Layers of the Bluetooth Software Stack 317

2 Overview of Host Controller Transport Layer **319**

3 Overview of Commands and Events **321**
3.1 Generic Events 322
3.2 Device Setup 322
3.3 Controller Flow Control 323
3.4 Controller Information 323
3.5 Controller Configuration 324
3.6 Device Discovery 325
3.7 Connection Setup 327
3.8 Remote Information 329
3.9 Synchronous Connections 330
3.10 Connection State 331
3.11 Piconet Structure 332
3.12 Quality of Service 333
3.13 Physical Links 334
3.14 Host Flow Control 335
3.15 Link Information 336
3.16 Authentication and Encryption 337
3.17 Testing 339
3.18 Alphabetical List of Commands and Events 340

4 HCI Flow Control **345**
4.1 Host to Controller Data Flow Control 345
4.2 Controller to Host Data Flow Control 346
4.3 Disconnection Behaviour 347
4.4 Command Flow Control 347
4.5 Command Error Handling 348

5 HCI Data Formats **349**
5.1 Introduction 349
5.2 Data and Parameter Formats 349
5.3 Connection Handles 350
5.4 Exchange of HCI-Specific Information 351
5.4.1 HCI Command Packet 351
5.4.2 HCI ACL Data Packets 353
5.4.3 HCI Synchronous Data Packets 355
5.4.4 HCI Event Packet 356

Petitioner Kiosoft Exhibit 1011 Part 1
Page 23

6 HCI Configuration Parameters 357
6.1 Scan Enable 357
6.2 Inquiry Scan Interval 357
6.3 Inquiry Scan Window 358
6.4 Inquiry Scan Type 358
6.5 Inquiry Mode 358
6.6 Page Timeout 359
6.7 Connection Accept Timeout 359
6.8 Page Scan Interval 360
6.9 Page Scan Window 360
6.10 Page Scan Period Mode 360
6.11 Page Scan Type 361
6.12 Voice Setting 361
6.13 PIN Type 362
6.14 Link Key 362
6.15 Authentication Enable 362
6.16 Encryption Mode 363
6.17 Failed Contact Counter 364
6.18 Hold Mode Activity 364
6.19 Link Policy Settings 365
6.20 Flush Timeout 366
6.21 Num Broadcast Retransmissions 366
6.22 Link Supervision Timeout 367
6.23 Synchronous Flow Control Enable 367
6.24 Local Name 368
6.25 Class Of Device 368
6.26 Supported Commands 369

7 HCI Commands and Events 373
7.1 Link Control Commands 373
7.1.1 Inquiry Command 373
7.1.2 Inquiry Cancel Command 375
7.1.3 Periodic Inquiry Mode Command 376
7.1.4 Exit Periodic Inquiry Mode Command 379
7.1.5 Create Connection Command 380
7.1.6 Disconnect Command 383
7.1.7 Create Connection Cancel Command 384
7.1.8 Accept Connection Request Command 386
7.1.9 Reject Connection Request Command 388
7.1.10 Link Key Request Reply Command 389
7.1.11 Link Key Request Negative Reply Command 391
7.1.12 PIN Code Request Reply Command 392

Petitioner Kiosoft Exhibit 1011 Part 1
Page 24

7.1.13 PIN Code Request Negative Reply Command 394
7.1.14 Change Connection Packet Type Command 395
7.1.15 Authentication Requested Command.....398
7.1.16 Set Connection Encryption Command 399
7.1.17 Change Connection Link Key Command 400
7.1.18 Master Link Key Command 401
7.1.19 Remote Name Request Command 402
7.1.20 Remote Name Request Cancel Command 404
7.1.21 Read Remote Supported Features Command.....405
7.1.22 Read Remote Extended Features Command 406
7.1.23 Read Remote Version Information Command.....407
7.1.24 Read Clock Offset Command.....408
7.1.25 Read LMP Handle Command 409
7.1.26 Setup Synchronous Connection Command 411
7.1.27 Accept Synchronous Connection Request Command 415
7.1.28 Reject Synchronous Connection Request Command .419
7.2 Link Policy Commands.....420
7.2.1 Hold Mode Command 420
7.2.2 Sniff Mode Command.....422
7.2.3 Exit Sniff Mode Command.....425
7.2.4 Park State Command 426
7.2.5 Exit Park State Command 428
7.2.6 QoS Setup Command 429
7.2.7 Role Discovery Command.....431
7.2.8 Switch Role Command.....432
7.2.9 Read Link Policy Settings Command 433
7.2.10 Write Link Policy Settings Command 434
7.2.11 Read Default Link Policy Settings Command 436
7.2.12 Write Default Link Policy Settings Command 437
7.2.13 Flow Specification Command 438
7.3 Controller & Baseband Commands.....440
7.3.1 Set Event Mask Command.....440
7.3.2 Reset Command 442
7.3.3 Set Event Filter Command 443
7.3.4 Flush Command 448
7.3.5 Read PIN Type Command 450
7.3.6 Write PIN Type Command.....451
7.3.7 Create New Unit Key Command 452

7.3.8 Read Stored Link Key Command....453
7.3.9 Write Stored Link Key Command....454
7.3.10 Delete Stored Link Key Command....456
7.3.11 Write Local Name Command....457
7.3.12 Read Local Name Command....458
7.3.13 Read Connection Accept Timeout Command....459
7.3.14 Write Connection Accept Timeout Command....460
7.3.15 Read Page Timeout Command....461
7.3.16 Write Page Timeout Command....462
7.3.17 Read Scan Enable Command....463
7.3.18 Write Scan Enable Command....464
7.3.19 Read Page Scan Activity Command....465
7.3.20 Write Page Scan Activity Command....467
7.3.21 Read Inquiry Scan Activity Command....468
7.3.22 Write Inquiry Scan Activity Command....469
7.3.23 Read Authentication Enable Command....470
7.3.24 Write Authentication Enable Command....471
7.3.25 Read Encryption Mode Command....472
7.3.26 Write Encryption Mode Command....473
7.3.27 Read Class of Device Command....474
7.3.28 Write Class of Device Command....475
7.3.29 Read Voice Setting Command....476
7.3.30 Write Voice Setting Command....477
7.3.31 Read Automatic Flush Timeout Command....478
7.3.32 Write Automatic Flush Timeout Command....479
7.3.33 Read Num Broadcast Retransmissions Command....480
7.3.34 Write Num Broadcast Retransmissions Command....481
7.3.35 Read Hold Mode Activity Command....482
7.3.36 Write Hold Mode Activity Command....483
7.3.37 Read Transmit Power Level Command....484
7.3.38 Read Synchronous Flow Control Enable Command...486
7.3.39 Write Synchronous Flow Control Enable Command...487
7.3.40 Set Controller To Host Flow Control Command....488
7.3.41 Host Buffer Size Command....489
7.3.42 Host Number Of Completed Packets Command....491
7.3.43 Read Link Supervision Timeout Command....493
7.3.44 Write Link Supervision Timeout Command....494
7.3.45 Read Number Of Supported IAC Command....496

Petitioner Kiosoft Exhibit 1011 Part 1
Page 26

7.3.46 Read Current IAC LAP Command 497
7.3.47 Write Current IAC LAP Command....498
7.3.48 Read Page Scan Period Mode Command 500
7.3.49 Write Page Scan Period Mode Command 501
7.3.50 Set AFH Host Channel Classification Command 502
7.3.51 Read Inquiry Scan Type Command 503
7.3.52 Write Inquiry Scan Type Command 504
7.3.53 Read Inquiry Mode Command 505
7.3.54 Write Inquiry Mode Command 506
7.3.55 Read Page Scan Type Command 507
7.3.56 Write Page Scan Type Command 508
7.3.57 Read AFH Channel Assessment Mode Command 509
7.3.58 Write AFH Channel Assessment Mode Command 510
7.4 Informational Parameters....512
7.4.1 Read Local Version Information Command....512
7.4.2 Read Local Supported Commands Command....514
7.4.3 Read Local Supported Features Command....515
7.4.4 Read Local Extended Features Command 516
7.4.5 Read Buffer Size Command....518
7.4.6 Read BD_ADDR Command....520
7.5 Status Parameters....521
7.5.1 Read Failed Contact Counter Command 521
7.5.2 Reset Failed Contact Counter Command 523
7.5.3 Read Link Quality Command 524
7.5.4 Read RSSI Command....525
7.5.5 Read AFH Channel Map Command 527
7.5.6 Read Clock Command 529
7.6 Testing Commands 531
7.6.1 Read Loopback Mode Command 531
7.6.2 Write Loopback Mode Command....532
7.6.3 Enable Device Under Test Mode Command 535
7.7 Events 537
7.7.1 Inquiry Complete Event....537
7.7.2 Inquiry Result Event 538
7.7.3 Connection Complete Event....540
7.7.4 Connection Request Event....541
7.7.5 Disconnection Complete Event 543
7.7.6 Authentication Complete Event....544

Petitioner Kiosoft Exhibit 1011 Part 1
Page 27

7.7.7 Remote Name Request Complete Event 545
7.7.8 Encryption Change Event 546
7.7.9 Change Connection Link Key Complete Event 547
7.7.10 Master Link Key Complete Event 548
7.7.11 Read Remote Supported Features Complete Event ... 549
7.7.12 Read Remote Version Information Complete Event ... 550
7.7.13 QoS Setup Complete Event 551
7.7.14 Command Complete Event 553
7.7.15 Command Status Event 554
7.7.16 Hardware Error Event 555
7.7.17 Flush Occurred Event 555
7.7.18 Role Change Event 556
7.7.19 Number Of Completed Packets Event 557
7.7.20 Mode Change Event 558
7.7.21 Return Link Keys Event 560
7.7.22 PIN Code Request Event 561
7.7.23 Link Key Request Event 562
7.7.24 Link Key Notification Event 563
7.7.25 Loopback Command Event 564
7.7.26 Data Buffer Overflow Event 564
7.7.27 Max Slots Change Event 565
7.7.28 Read Clock Offset Complete Event 566
7.7.29 Connection Packet Type Changed Event 567
7.7.30 QoS Violation Event 569
7.7.31 Page Scan Repetition Mode Change Event 570
7.7.32 Flow Specification Complete Event 571
7.7.33 Inquiry Result with RSSI Event 573
7.7.34 Read Remote Extended Features Complete Event 575
7.7.35 Synchronous Connection Complete Event 576
7.7.36 Synchronous Connection Changed event 578

8 List of Figures 581

9 List of Tables 583

Appendix A:
Deprecated Commands, Events and Configuration Parameters 585

Petitioner Kiosoft Exhibit 1011 Part 1
Page 28

Part F
MESSAGE SEQUENCE CHARTS

Contents **595**

1 Introduction **599**
1.1 Notation 599
1.2 Flow of Control 600
1.3 Example MSC 600

2 Services Without Connection Request **601**
2.1 Remote Name Request 601
2.2 One-time Inquiry 602
2.3 Periodic Inquiry 604

3 ACL Connection Establishment and Detachment **607**
3.1 Connection Setup 608

4 Optional Activities After ACL Connection Establishment **615**
4.1 Authentication Requested 615
4.2 Set Connection Encryption 616
4.3 Change Connection Link Key 617
4.4 Master Link Key 618
4.5 Read Remote Supported Features 620
4.6 Read Remote Extended Features 620
4.7 Read Clock Offset 621
4.8 Read Remote Version Information 621
4.9 QOS Setup 622
4.10 Switch Role 622

5 Synchronous Connection Establishment and Detachment **625**
5.1 Synchronous Connection Setup 625

6 Sniff, Hold and Park **631**
6.1 Sniff Mode 631
6.2 Hold Mode 632
6.3 Park State 634

7 Buffer Management, Flow Control **637**

8 Loopback Mode **639**
8.1 Local Loopback Mode 639
8.2 Remote Loopback Mode 641

9 List of Figures **643**

Petitioner Kiosoft Exhibit 1011 Part 1
Page 29

Part G
SAMPLE DATA

Contents **647**

1 Encryption Sample Data **649**
- 1.1 Generating Kc' from Kc, ... 649
- 1.2 First Set of Sample Data 652
- 1.3 Second Set of Sample Data 660
- 1.4 Third Set of Samples 668
- 1.5 Fourth Set of Samples 676

2 Frequency Hopping Sample Data **685**
- 2.1 First set 686
- 2.2 Second set 692
- 2.3 Third set 698

3 Access Code Sample Data **705**

4 HEC and Packet Header Sample Data **709**

5 CRC Sample Data **711**

6 Complete Sample Packets **713**
- 6.1 Example of DH1 Packet 713
- 6.2 Example of DM1 Packet 714

7 Whitening Sequence Sample Data **715**

8 FEC Sample Data **719**

9 Encryption Key Sample Data **721**
- 9.1 Four Tests of E1 721
- 9.2 Four Tests of E21 726
- 9.3 Three Tests of E22 728
- 9.4 Tests of E22 With Pin Augmenting 730
- 9.5 Four Tests of E3 740

Petitioner Kiosoft Exhibit 1011 Part 1
Page 30

Part H
SECURITY SPECIFICATION

Contents **747**

1 Security Overview **749**

2 Random Number Generation **751**

3 Key Management **753**

3.1 Key Types 753

3.2 Key Generation and Initialization 755

3.2.1 Generation of initialization key, ... 756

3.2.2 Authentication 756

3.2.3 Generation of a unit key 756

3.2.4 Generation of a combination key 757

3.2.5 Generating the encryption key 758

3.2.6 Point-to-multipoint configuration 759

3.2.7 Modifying the link keys 760

3.2.8 Generating a master key 760

4 Encryption **763**

4.1 Encryption Key Size Negotiation 764

4.2 Encryption of Broadcast Messages 764

4.3 Encryption Concept 765

4.4 Encryption Algorithm 766

4.4.1 The operation of the cipher 768

4.5 LFSR Initialization 769

4.6 Key Stream Sequence 772

5 Authentication **773**

5.1 Repeated Attempts 775

6 The Authentication And Key-Generating Functions **777**

6.1 The Authentication Function E1 777

6.2 The Functions Ar and A'r 779

6.2.1 The round computations 779

6.2.2 The substitution boxes "e" and "l" 779

6.2.3 Key scheduling 780

6.3 E2-Key Generation Function for Authentication 781

6.4 E3-Key Generation Function for Encryption 783

7 List of Figures **785**

8 List of Tables **787**

Specification Volume 3
Core System Package
[Host volume]

Table of Contents **5**

Part A
LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL SPECIFICATION

Contents **17**

1 Introduction **21**
1.1 L2CAP Features 21
1.2 Assumptions 25
1.3 Scope 25
1.4 Terminology 26

2 General Operation **29**
2.1 Channel Identifiers 29
2.2 Operation Between Devices 29
2.3 Operation Between Layers 31
2.4 Modes of Operation 31

3 Data Packet Format **33**
3.1 Connection-oriented Channel in Basic L2CAP Mode 33
3.2 Connectionless Data Channel in Basic L2CAP Mode 34
3.3 Connection-oriented Channel in Retransmission/Flow Control Modes 35
3.3.1 L2CAP header fields 35
3.3.2 Control field (2 octets) 36
3.3.3 L2CAP SDU length field (2 octets) 38
3.3.4 Information payload field (0 to 65531 octets) 38
3.3.5 Frame check sequence (2 octets) 39
3.3.6 Invalid frame detection 40

4 Signalling Packet Formats **41**
4.1 Command Reject (code 0x01) 43
4.2 Connection Request (code 0x02) 44
4.3 Connection Response (code 0x03) 46
4.4 Configuration Request (code 0x04) 47
4.5 Configuration Response (code 0x05) 50
4.6 Disconnection Request (code 0x06) 52
4.7 Disconnection Response (code 0x07) 53
4.8 Echo Request (code 0x08) 53
4.9 Echo Response (code 0x09) 54

Petitioner Kiosoft Exhibit 1011 Part 1
Page 32

4.10 Information Request (code 0x0A) 54
4.11 Information Response (code 0x0B)....55
4.12 Extended Feature Mask....56

5 Configuration Parameter Options....57
5.1 Maximum Transmission Unit (MTU)....57
5.2 Flush Timeout Option....59
5.3 Quality of Service (QoS) Option....60
5.4 Retransmission and Flow Control Option....64

6 State Machine....67
6.1 General rules for the state machine:....67
6.1.1 CLOSED state....68
6.1.2 WAIT_CONNECT_RSP state....69
6.1.3 WAIT_CONNECT state....69
6.1.4 CONFIG state....70
6.1.5 OPEN state....73
6.1.6 WAIT_DISCONNECT state....73
6.2 Timers events....74
6.2.1 RTX....74
6.2.2 ERTX....75

7 General Procedures....79
7.1 Configuration Process....79
7.1.1 Request path....80
7.1.2 Response path....80
7.2 Fragmentation and Recombination....81
7.2.1 Fragmentation of L2CAP PDUs....81
7.2.2 Recombination of L2CAP PDUs....82
7.3 Encapsulation of SDUs....83
7.3.1 Segmentation of L2CAP SDUs....83
7.3.2 Reassembly of L2CAP SDUs....84
7.3.3 Segmentation and fragmentation....84
7.4 Delivery of Erroneous L2CAP SDUs....85
7.5 Operation with Flushing....85
7.6 Connectionless Data Channel....86

8 Procedures for Flow Control and Retransmission 87
8.1 Information Retrieval 87
8.2 Function of PDU Types for Flow Control and Retransmission ... 87
8.2.1 Information frame (I-frame) 87
8.2.2 Supervisory Frame (S-frame) 87
8.3 Variables and Sequence Numbers 89
8.3.1 Sending peer 89
8.3.2 Receiving peer 91
8.4 Retransmission Mode 93
8.4.1 Transmitting frames 93
8.4.2 Receiving I-frames 95
8.4.3 I-frames pulled by the SDU reassembly function 95
8.4.4 Sending and receiving acknowledgements 96
8.4.5 Receiving REJ frames 97
8.4.6 Waiting acknowledgements 97
8.4.7 Exception conditions 97
8.5 Flow Control Mode 99
8.5.1 Transmitting I-frames 99
8.5.2 Receiving I-frames 100
8.5.3 I-frames pulled by the SDU reassembly function 100
8.5.4 Sending and receiving acknowledgements 100
8.5.5 Waiting acknowledgements 101
8.5.6 Exception conditions 101
9 List of Figures 103
10 List of Tables 105

Appendix A: Configuration MSCs 107

Part B
SERVICE DISCOVERY PROTOCOL (SDP)

Contents **113**

1 Introduction **115**
1.1 General Description 115
1.2 Motivation 115
1.3 Requirements 115
1.4 Non-requirements and Deferred Requirements 116
1.5 Conventions 116
1.5.1 Bit And Byte Ordering Conventions 116

2 Overview **117**
2.1 SDP Client-Server Interaction 117
2.2 Service Record 118
2.3 Service Attribute 120
2.4 Attribute ID 120
2.5 Attribute Value 121
2.6 Service Class 121
2.6.1 A Printer Service Class Example 122
2.7 Searching for Services 123
2.7.1 UUID 123
2.7.2 Service Search Patterns 124
2.8 Browsing for Services 124
2.8.1 Example Service Browsing Hierarchy 125

3 Data Representation **127**
3.1 Data Element 127
3.2 Data Element Type Descriptor 127
3.3 Data Element Size Descriptor 128
3.4 Data Element Examples 129

4 Protocol Description **131**
4.1 Transfer Byte Order 131
4.2 Protocol Data Unit Format 131
4.3 Partial Responses and Continuation State 133
4.4 Error Handling 133
4.4.1 SDP_ErrorResponse PDU 134
4.5 ServiceSearch Transaction 135
4.5.1 SDP_ServiceSearchRequest PDU 135
4.5.2 SDP_ServiceSearchResponse PDU 136
4.6 ServiceAttribute Transaction 138
4.6.1 SDP_ServiceAttributeRequest PDU 138
4.6.2 SDP_ServiceAttributeResponse PDU 140

Petitioner Kiosoft Exhibit 1011 Part 1
Page 35

4.7 ServiceSearchAttribute Transaction 141
4.7.1 SDP_ServiceSearchAttributeRequest PDU 141
4.7.2 SDP_ServiceSearchAttributeResponse PDU 143

5 Service Attribute Definitions 145
5.1 Universal Attribute Definitions 145
5.1.1 ServiceRecordHandle Attribute 145
5.1.2 ServiceClassIDList Attribute 146
5.1.3 ServiceRecordState Attribute 146
5.1.4 ServiceID Attribute 146
5.1.5 ProtocolDescriptorList Attribute 147
5.1.6 BrowseGroupList Attribute 148
5.1.7 LanguageBaseAttributeIDList Attribute 148
5.1.8 ServiceInfoTimeToLive Attribute 149
5.1.9 ServiceAvailability Attribute 150
5.1.10 BluetoothProfileDescriptorList Attribute 150
5.1.11 DocumentationURL Attribute 151
5.1.12 ClientExecutableURL Attribute 151
5.1.13 IconURL Attribute 152
5.1.14 ServiceName Attribute 152
5.1.15 ServiceDescription Attribute 153
5.1.16 ProviderName Attribute 153
5.1.17 Reserved Universal Attribute IDs 153
5.2 ServiceDiscoveryServer Service Class Attribute Definitions ... 154
5.2.1 ServiceRecordHandle Attribute 154
5.2.2 ServiceClassIDList Attribute 154
5.2.3 VersionNumberList Attribute 154
5.2.4 ServiceDatabaseState Attribute 155
5.2.5 Reserved Attribute IDs 155
5.3 BrowseGroupDescriptor Service Class Attribute Definitions ... 155
5.3.1 ServiceClassIDList Attribute 155
5.3.2 GroupID Attribute 156
5.3.3 Reserved Attribute IDs 156

Appendix A:
Background Information 157

Appendix B:
Example SDP Transactions 158

Petitioner Kiosoft Exhibit 1011 Part 1
Page 36

Part C
GENERIC ACCESS PROFILE

Contents **173**
Foreword **177**

1 Introduction **179**
1.1 Scope 179
1.2 Symbols and conventions 179
1.2.1 Requirement status symbols 179
1.2.2 Signaling diagram conventions 180
1.2.3 Notation for timers and counters 180

2 Profile overview **181**
2.1 Profile stack 181
2.2 Configurations and roles 181
2.3 User requirements and scenarios 182
2.4 Profile fundamentals 183
2.5 Conformance 183

3 User interface aspects **185**
3.1 The user interface level 185
3.2 Representation of Bluetooth parameters 185
3.2.1 Bluetooth device address (BD_ADDR) 185
3.2.2 Bluetooth device name (the user-friendly name) 185
3.2.3 Bluetooth passkey (Bluetooth PIN) 186
3.2.4 Class of Device 187
3.3 Pairing 188

4 Modes **189**
4.1 Discoverability modes 189
4.1.1 Non-discoverable mode 190
4.1.2 Limited discoverable mode 190
4.1.3 General discoverable mode 191
4.2 Connectability modes 193
4.2.1 Non-connectable mode 193
4.2.2 Connectable mode 193
4.3 Pairing modes 195
4.3.1 Non-pairable mode 195
4.3.2 Pairable mode 195

Petitioner Kiosoft Exhibit 1011 Part 1
Page 37

5 Security aspects 197
5.1 Authentication 197
5.1.1 Purpose 197
5.1.2 Term on UI level 197
5.1.3 Procedure 198
5.1.4 Conditions 198
5.2 Security modes 198
5.2.1 Security mode 1 (non-secure) 200
5.2.2 Security mode 2 (service level enforced security) 200
5.2.3 Security modes 3 (link level enforced security) 200
6 Idle mode procedures 201
6.1 General inquiry 201
6.1.1 Purpose 201
6.1.2 Term on UI level 201
6.1.3 Description 202
6.1.4 Conditions 202
6.2 Limited inquiry 202
6.2.1 Purpose 202
6.2.2 Term on UI level 203
6.2.3 Description 203
6.2.4 Conditions 203
6.3 Name discovery 204
6.3.1 Purpose 204
6.3.2 Term on UI level 204
6.3.3 Description 204
6.3.4 Conditions 205
6.4 Device discovery 205
6.4.1 Purpose 205
6.4.2 Term on UI level 205
6.4.3 Description 206
6.4.4 Conditions 206
6.5 Bonding 206
6.5.1 Purpose 206
6.5.2 Term on UI level 206
6.5.3 Description 207
6.5.4 Conditions 208

Petitioner Kiosoft Exhibit 1011 Part 1
Page 38

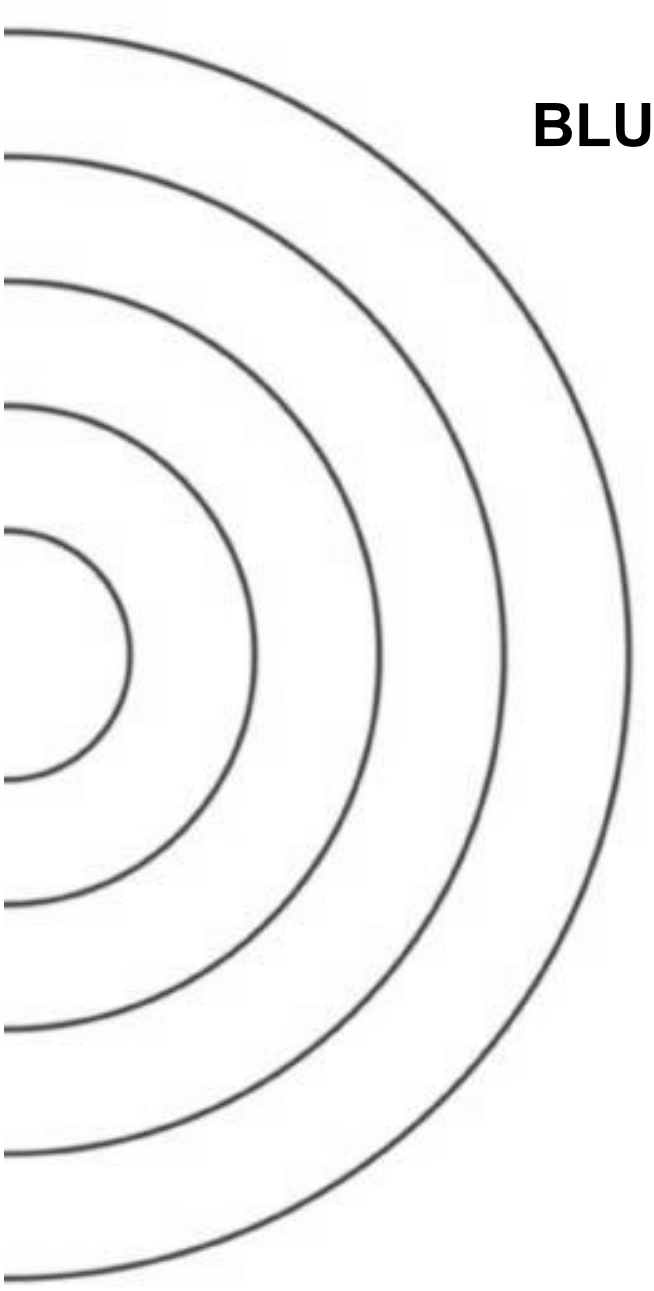
7 Establishment procedures 209
7.1 Link establishment 209
7.1.1 Purpose 209
7.1.2 Term on UI level 209
7.1.3 Description 210
7.1.4 Conditions 211
7.2 Channel establishment....212
7.2.1 Purpose 212
7.2.2 Term on UI level 212
7.2.3 Description 212
7.2.4 Conditions 213
7.3 Connection establishment....214
7.3.1 Purpose 214
7.3.2 Term on UI level 214
7.3.3 Description 214
7.3.4 Conditions 215
7.4 Establishment of additional connection....215
8 Definitions....217
8.1 General definitions 217
8.2 Connection-related definitions....217
8.3 Device-related definitions....218
8.4 Procedure-related definitions 219
8.5 Security-related definitions....219
9 Appendix A (Normative): Timers and constants 221
10 Appendix B (Informative): Information flows of related procedures....223
11 References....227

Part D
TEST SUPPORT

Contents **231**

1 Test Methodology **233**
1.1 Test Scenarios 233
1.1.1 Test setup 233
1.1.2 Transmitter Test 234
1.1.3 LoopBack test 239
1.1.4 Pause test 242
1.2 References 242
2 Test Control Interface (TCI) **243**
2.1 Introduction 243
2.1.1 Terms used 243
2.1.2 Usage of the interface 243
2.2 TCI Configurations 244
2.2.1 Bluetooth RF requirements 244
2.2.2 Bluetooth protocol requirements 245
2.2.3 Bluetooth profile requirements 246
2.3 TCI Configuration and Usage 247
2.3.1 Transport layers 247
2.3.2 Baseband and link manager qualification 248
2.3.3 HCI qualification 250

Petitioner Kiosoft Exhibit 1011 Part 1
Page 40



BLUETOOTH COMPLIANCE REQUIREMENTS

This document specifies the requirements for Bluetooth compliance.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 42

CONTENTS

1 Scope 45

2 Definitions 47

3 Legal Aspects 49

4 Introduction to the Bluetooth Qualification Program 51

Petitioner Kiosoft Exhibit 1011 Part 1
Page 44

1 SCOPE

The Bluetooth Promoters and the Bluetooth Adopters have signed the applicable Bluetooth Agreements. These agreements grant Bluetooth Promoters and Bluetooth Adopters a Bluetooth license for "products which comply with the Specification".

This document specifies the requirements which must be met by a Bluetooth Promoter or Bluetooth Adopter to demonstrate that a particular product does "comply with the Specification", thereby qualifying that particular product to be subject to the rights extended by the applicable Bluetooth Agreement.

The Bluetooth Qualification Program is the process by which a Bluetooth Promoter or a Bluetooth Adopter demonstrates that a particular product meets the requirements specified herein. This document provides an introduction to the Bluetooth Qualification Program. The Bluetooth Qualification Program is defined in the Bluetooth Qualification Program Reference Document (PRD) available on the Bluetooth Web site.

Regulatory requirements and governmental type approval requirements are outside the scope of this document.

Requirements on Trademark usage are outside the scope of this document and the Qualification Program. However, successful completion of the Qualification program is a precondition to usage of the Bluetooth Trademark. Details on trademark usage can be found in the applicable Bluetooth agreements and Bluetooth Trademark License Agreement.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 45

Petitioner Kiosoft Exhibit 1011 Part 1
Page 46

2 DEFINITIONS

Bluetooth Agreements - Promoter, Early Adopters or Adopters Agreement, whichever is applicable.

Bluetooth License - All the rights, granted in the Promoter, Early Adopters or Adopters Agreements and the Bluetooth Trademark License

Bluetooth Qualification Administrator (BQA) - Responsible for administering the Bluetooth Qualification Program on behalf of the BQRB.

Bluetooth Qualification Body (BQB) - An individual person recognized by the BQRB to be responsible for checking declarations and documents against requirements, verifying the authenticity of product test reports, and listing products on the official Bluetooth Qualified Products List.

Bluetooth Qualification process - The process defined in the Bluetooth Qualification Program Reference Document (PRD) for Bluetooth Qualification.

Bluetooth Qualification program - The implementation of the Bluetooth Qualification process.

Bluetooth Qualification Review Board (BQRB) - Responsible for managing, reviewing and improving the Bluetooth Qualification program. The Bluetooth SIG promoter companies appoint delegates to the BQRB.

Bluetooth Qualification Test Facility (BQTF) - A test facility officially accredited by the BQRB to test Bluetooth products.

Bluetooth System Specification - The adopted version of this document.

Bluetooth Trademark - As defined in the Bluetooth Promoter, Early Adopters Agreement or Adopters Agreement for 1.0 Bluetooth Core Specification, or the Bluetooth Trademark License Agreement, whichever is applicable.

Compliant Portion - Only those specific portions of products (hardware, software or combinations thereof) that: (i) implement and are compliant with the Bluetooth System Specifications, (ii) are qualified pursuant to the Qualification Process, (iii) are within the bounds of the Scope and (iv) meet the requirements set forth in the Compliance Requirements.

Scope - As used in the definition of Compliant Portion, the protocols and data formats needed for Bluetooth interoperability, and the electrical signaling characteristics solely to the extent disclosed with particularity in the Bluetooth System Specifications where the sole purpose of such disclosure is to enable

products to interoperate, interconnect or communicate as defined within the Bluetooth System Specifications. For clarification, the Scope shall not include (i) any enabling technologies that may be necessary to make or use any prod-

Petitioner Kiosoft Exhibit 1011 Part 1
Page 47

uct or portion thereof that complies with the Bluetooth System Specifications, but are not themselves expressly set forth in the Bluetooth System Specifications (e.g., semiconductor manufacturing technology, compiler technology, object oriented technology, basic operating system technology, etc.); or (ii) the implementation of other published specifications developed elsewhere but referred to in the body of the Bluetooth System Specifications; or (iii) any portions of any product and any combinations thereof the purpose or function of which is not required for compliance with the Bluetooth System Specifications; or (iv) Application Programming Interfaces, applications, or user interfaces; including the technology used to generate, display or interact with a user.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 48

3 LEGAL ASPECTS

Bluetooth adopters legal rights and obligations are governed by their applicable Bluetooth Agreements.

The Bluetooth Specification has been created according to our best knowledge, to meet regulatory requirements worldwide. Regulatory certification, as such, is not a part of the Bluetooth Qualification requirements, yet is a requirement in all markets. It is the sole responsibility of each manufacturer to ensure that their products have all necessary regulatory approvals for the markets where they are intended to be sold or used.

A product must complete Bluetooth Qualification as one of the requirements for “complying with the Specification”. Under all applicable Bluetooth Agreements, every product is required to successfully complete the Bluetooth Qualification process. The rights granted under the applicable Bluetooth agreements do not apply to products which have not completed the Qualification process.

Sanctions may be invoked by holders of intellectual property, against any company responsible for producing or trading (a) products containing elements of the Bluetooth Interface, as defined in applicable Bluetooth Agreements, that do not comply with the Specification, (b) which are not Compliant Portions, or (c) products, containing elements of the Bluetooth Interface, that have not completed Bluetooth Qualification.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 49

Petitioner Kiosoft Exhibit 1011 Part 1
Page 50

4 INTRODUCTION TO THE BLUETOOTH QUALIFICATION PROGRAM

The Bluetooth Qualification program is the framework that establishes the qualification rules and procedures. This section introduces the framework. The Bluetooth Qualification Program as well as Qualification requirements are defined in the Bluetooth Qualification Program Reference Document (PRD) available on the Bluetooth web site. Additionally, Guidelines for Bluetooth assigned names and numbers are defined in the PRD.

The Program defines the following entities:

- Bluetooth Qualification Review Board (BQRB)
- Bluetooth Qualification Administrator (BQA)
- Bluetooth Qualification Test Facility (BQTF)
- Bluetooth Qualification Body (BQB)

Figure 4.1introduces the process by which a Member qualifies a product.

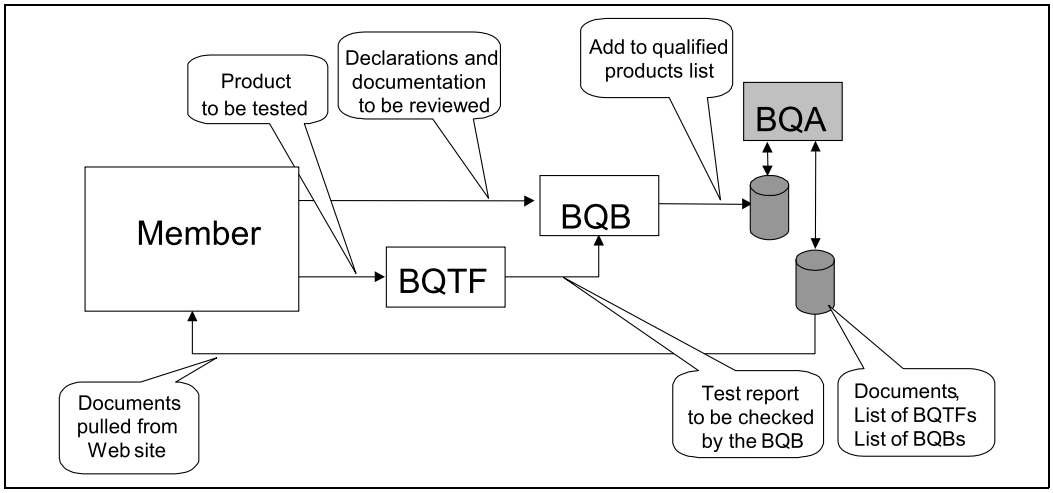


Figure 4.1:

An entity intending to produce and/or market a product, which includes Bluetooth functionality, must first become a Bluetooth Member by executing the applicable Bluetooth Agreements available via the Bluetooth Web site. Next, the Member must demonstrate that each particular product design complies with the Bluetooth System Specifications (including Compliance Requirements). This is accomplished when the particular product is listed as a.qualified product after successfully completing the Bluetooth Qualification process. For purposes of the rights granted under the applicable Bluetooth Agreement, the condition of Qualification is met from the date of listing. Under all applicable Bluetooth Agreements, every product is required to successfully complete the Bluetooth Qualification process. The rights granted under the applicable Blue-

tooth agreements do not apply to products which have not completed the Qualification process.

The Member selects a BQB from the list of recognized BQBs found on the Bluetooth web site. The BQB can assist the Member throughout the entire product qualification process. The Member prepares a compliance folder and works with the BQB to develop a product test plan. If the product test plan requires testing services of a BQTF, the Member can select a BQTF that has been accredited to perform the type of testing required. All necessary documentation to demonstrate compliance to the specification is provided to the BQB by the Member.

The BQB reviews the submittals and prepares a qualified product notice. In the event the BQB has a question regarding the conformance or interoperability of a product the BQB may, with the Member's permission, submit information via the BQA to the BQRB for a ruling. In the event the BQRB must be consulted, the Member will be requested to prepare a submission according to BQRB guidelines.

When all current requirements are satisfied, the BQB informs the Member that the product is ready for listing as a Bluetooth qualified product. With the consent of the Member, the BQB posts the product information to the official Qualified Product List.

A fee for product listing shall be paid by the BQB prior to product listing to finance the administration associated with the Qualification program.

Any Member can view the Bluetooth Qualified Product List. Confidential product information will not be published.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 52



APPENDIX

Petitioner Kiosoft Exhibit 1011 Part 1
Page 54

CONTENTS

1 Revision History 57
1.1 [vol 0] Master TOC & Compliance Requirements 57
1.1.1 Bluetooth Compliance Requirements 57
1.2 [Vol 1] Architecture & Terminology Overview 57
1.3 [Vol 2 & 3] Core System Package 58
2 Contributors 59
2.1 [vol 0] Master TOC & Compliance Requirements 59
2.1.1 Part B: Bluetooth Compliance Requirements 59
2.2 [vol 1] Architecture &Terminology Overview 59
2.2.1 Part A: Architectural Overview 59
2.2.2 Part B: Acronyms & Abbreviations 60
2.2.3 Part C: Changes from Bluetooth Specification v1.1 60
2.3 [Vol 2] Core System Package, Controller 61
2.3.1 Part A: Radio Specification 61
2.3.2 Part B: Baseband Specification 62
2.3.3 Part C: Link Manager Protocol 65
2.3.4 Part D: Error Codes 66
2.3.5 Part E: Bluetooth Host Controller Interface Functional Specification 67
2.3.6 Part F: Message Sequence Charts 68
2.3.7 Part G: Sample Data 69
2.3.8 Part H: Security Specification 69
2.4 [Vol 3] Core System Package, Host 70
2.4.1 Part A: Logical Link Control and Adaptation Protocol Specification 70
2.4.2 Part B: Service Discovery Protocol (SDP) 71
2.4.3 Part C Generic Access Profile 72
2.4.4 Part D: Test Support 72

Petitioner Kiosoft Exhibit 1011 Part 1
Page 55

Petitioner Kiosoft Exhibit 1011 Part 1
Page 56

1 REVISION HISTORY

Public versions are marked with bold in the tables below.

Beginning with the v1.2 of the Core System Package all Bluetooth specification documents, protocols and profiles, are transferred to a new partitioning comprising 13 volumes instead of the two (Core and Profiles) previously used.

For more detailed information about changes between version published before v1.2, please see the Appendix I “Revision History” in v1.1.

1.1 [VOL 0] MASTER TOC & COMPLIANCE REQUIREMENTS

1.1.1 Bluetooth Compliance Requirements

Rev	Date	Comments
v1.2	Nov 05 2003	This Part was moved from the Core volume. No content changes been made to this document since v1.1

1.2 [VOL 1] ARCHITECTURE & TERMINOLOGY OVERVIEW

Rev	Date	Comments
v1.2	Nov 05 2003	New volume with informational content. This volume will always be updated in parallel with the Core System volumes

Petitioner Kiosoft Exhibit 1011 Part 1
Page 57

1.3 [VOL 2 & 3] CORE SYSTEM PACKAGE

Rev	Date	Comments
v1.2	Nov 05 2003	New features added in v1.2: - Architectual overview - Faster connection - Adaptive frequency hopping - Extended SCO links - Enhanced error detection and flow control - Enhanced synchronization capability - Enhanced flow specification The Core System Package now comprises two volumes and the text has gone through a radical change both in terms of structure and nomenclature. The language is also more precise and is adapted to meet the IEEE standard. The following parts are moved from the Core System Package to other volumes or has been deprecated: RFCOMM [vol 7], Object Exchange (IrDA Interoperability) [vol 8], TCS [vol 9], Interoperability Requirements for Bluetooth as a WAP Bearer [vol 6], HCI USB Transport Layer [vol4], HCI RS232 Transport Layer [vol 4], HCI UART Transport Layer [vol 4], Bluetooth Compliance Requirements [vol 0], Optional Paging Schemes [deprecated]
1.1	Feb 22nd 2001	The specification was updated with Errata items previously published on the website. The Bluetooth Assigned Numbers appendix was lifted out from the specification to allow continuos maintenance on the website.
1.0B	Dec 1st 1999	The specification was updated with Errata items previously published on the website and was revised from a linguistic point of view. The following parts was added: Interoperability Requirements for Bluetooth as a WAP Bearer, Test Control Interface, Sample Data (appendix), Bluetooth Audio (appendix), Baseband Timers (appendix) and Optional Paging Scheme (appendix)
1.0a	July 26th 1999	The first version of the Bluetooth Specification published on the public website. Added part: Bluetooth Compliance Requirements.
1.0 draft	July 5th 1999	The following parts was added: Service Discovery Protocol (SDP), Telephony Control Specification (TCS), Bluetooth Assigned Numbers (appendix) and Message Sequence Charts (appendix)
0.9	April 30th 1999	The following parts was added: IrDA Interoperability, HCI RS232 Transport Layer, HCI UART Transport Layer and Test Mode
0.8	Jan 21st 1999	The following parts was added: Radio Specification, L2CAP, RFCOMM, HCI & HCI USB Transport Layer
0.7	Oct 19th 1998	This first version only included Baseband and Link Manager Protocol

Petitioner Kiosoft Exhibit 1011 Part 1
Page 58

2 CONTRIBUTORS

2.1 [VOL 0] MASTER TOC & COMPLIANCE REQUIREMENTS

The Bluetooth Specification was compiled and edited by Dan Sönnerstam, Pyramid Communication AB.

2.1.1 Part B: Bluetooth Compliance Requirements

BQRB (Editor)	
Wayne Park	3Com Corporation
Lawrence Jones	ComBit, Inc.
Gary Robinson	IBM Corporation
Georges Seuron	IBM Corporation
Rick Jessop	Intel Corporation
John Webb	Intel Corporation
Bruce Littlefield	Lucent Technologies, Inc.
Brian A. Redding	Motorola, Inc.
Waldemar Hontscha	Nokia Corporation
Petri Morko	Nokia Corporation
Magnus Hansson	Telefonaktiebolaget LM Ericsson
Magnus Sommansson	Telefonaktiebolaget LM Ericsson
Göran Svennarp	Telefonaktiebolaget LM Ericsson
Warren Allen	Toshiba Corporation
John Shi	Toshiba Corporation

2.2 [VOL 1] ARCHITECTURE &TERMINOLOGY OVERVIEW

2.2.1 Part A: Architectural Overview

Jennifer Bray	CSR
Robin Heydon	CSR
Henrik Hedlund	Ericsson
Martin van der Zee	Ericsson
Andy Glass	Microsoft
Brian Redding	Motorola
Jürgen Schnitzler	Nokia
Thomas Müller	Nokia
Joel Linsky	Silicon Wave
Terry Bourk	Silicon Wave
Michael Hasling	Tality
Yoshimitsu Shimojo	Toshiba
Toshiki Kizu	Toshiba
John Mersh	TTPCom

Petitioner Kiosoft Exhibit 1011 Part 1
Page 59

2.2.2 Part B: Acronyms & Abbreviations

Dan Sönnerstam	Pyramid Communication AB

2.2.3 Part C: Changes from Bluetooth Specification v1.1

Tom Siep	Bluetooth SIG Inc.
Robin Heydon	CSR
Henrik Hedlund	Ericsson
Rob Davies	Philips
Joel Linsky	Silicon Wave
John Mersh	TTPCom

Petitioner Kiosoft Exhibit 1011 Part 1
Page 60

2.3 [VOL 2] CORE SYSTEM PACKAGE, CONTROLLER

2.3.1 Part A: Radio Specification

Version 1.2

Tom Siep	Bluetooth SIG Inc.
Jennifer Bray	CSR
Robin Heydon	CSR
Morten Gade	Digianswer/Motorola
Henrik Hedlund	Ericsson
Stefan Agnani	Ericsson
Robert Kokke	Ericsson
Roland Hellfajer	Infineon
Thomas Müller	Nokia
Antonio Salloum	Philips
Joel Linsky	Silicon Wave
Steve Hall	Silicon Wave
Oren Eliezer	Texas Instruments
Mike Fitton	Toshiba

Previous versions

Steve Williams	3Com Corporation
Todor V. Cooklov	Aware
Poul Hove Kristensen	Digianswer A/S
Kurt B. Fischer	Hyper Corporation
Kevin D. Marquess	Hyper Corporation
Troy Beukema	IBM Corporation
Brian Gaucher	IBM Corporation
Jeff Schiffer	Intel Corporation
James P. Gilb	Mobilian
Rich L. Ditch	Motorola, Inc.
Paul Burgess	Nokia Corporation
Olaf Joeressen	Nokia Corporation
Thomas Müller	Nokia Corporation
Arto T. Palin	Nokia Corporation
Steven J. Shellhammer	Symbol
Sven Mattisson	Telefonaktiebolaget LM Ericsson
Lars Nord (Section Owner)	Telefonaktiebolaget LM Ericsson
Anders Svensson	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Allen Hotari	Toshiba Corporation

2.3.2 Part B: Baseband Specification

Version 1.2

P G Madhavan	Agere
Hongbing Gan	Bandspeed, Inc.
Tod Sizer	Bell Labs
Alexander Thoukydides	CSR
Jennifer Bray	CSR
Robin Heydon	CSR
Kim Schneider	Digianswer/Motorola
Knud Dyring-Olsen	Digianswer/Motorola
Niels Nielsen	Digianswer/Motorola
Henrik Andersen	Digianswer/Motorola
Christian Gehrmann	Ericsson
Henrik Hedlund	Ericsson
Jan Åberg	Ericsson
Martin van der Zee	Ericsson
Rakesh Taori	Ericsson
Jaap Haartsen	Ericsson
Stefan Züerbes	Ericsson
Roland Hellfajer	Infineon
YC Maa	Integrated Programmable Communications, Inc.
HungKun Chen	Integrated Programmable Communications, Inc.
Steve McGowan	Intel
Adrian Stephens	Mobilian Corporation
Jim Lansford	Mobilian Corporation
Eric Meihofer	Motorola
Arto Palin	Nokia
Carmen Kuhl	Nokia
Hannu Laine	Nokia
Jürgen Schnitzler	Nokia
Päivi Ruuska	Nokia
Thomas Müller	Nokia
Antonio Salloum	Philips
Harmke de Groot	Philips
Marianne van de Casteelee	Philips
Rob Davies	Philips
Roland Matthijssen	Philips
Joel Linsky (section owner)	Silicon Wave
Terry Bourk	Silicon Wave
Gary Schneider	Symbol Technologies, Inc.
Stephen J. Shellhammer	Symbol Technologies, Inc.
Michael Hasling	Tality
Amihai Kidron	Texas Instruments
Dan Michael	Texas Instruments

Eli Dekel	Texas Instruments
Jie Liang	Texas Instruments
Oren Eliezer	Texas Instruments
Tally Shina	Texas Instruments
Yariv Raveh	Texas Instruments
Anuj Batra	Texas Instruments
Katsuhiro Kinoshita	Toshiba
Toshiki Kizu	Toshiba
Yoshimitsu Shimojo	Toshiba
Charles Sturman	TTPCom
John Mersh	TTPCom
Sam Turner	TTPCom
Christoph Scholtz	University of Bonn
Simon Baatz	University of Bonn

Previous versions

Kevin D. Marquess	Hyper Corporation
Chatschik Bisdikian	IBM Corporation
Kris Fleming	Intel Corporation
James P. Gilb	Mobilian
David E. Cypher	NIST
Nada Golmie	NIST
Olaf Joeressen	Nokia Corporation
Thomas Müller	Nokia Corporation
Charlie Mellone	Motorola, Inc.
Harmke de Groot	Philips
Terry Bourk	Silicon Wave
Steven J. Shellhammer	Symbol
Jaap Haartsen	Telefonaktiebolaget LM Ericsson
Henrik Hedlund (Section Owner)	Telefonaktiebolaget LM Ericsson
Tobias Melin	Telefonaktiebolaget LM Ericsson
Joakim Persson	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Onn Haran	Texas Instruments
Thomas M. Siep	Texas Instruments
Ayse Findikli	Zeevo, Inc.

Previous versions [Encryption Sample Data, appendix]

Thomas Müller	Nokia Corporation
Thomas Sander	Nokia Corporation
Joakim Persson (Section Owner)	Telefonaktiebolaget LM Ericsson

Previous versions [Bluetooth Audio, appendix]

Magnus Hansson	Telefonaktiebolaget LM Ericsson
Fisseha Mekuria	Telefonaktiebolaget LM Ericsson
Mats Omrin	Telefonaktiebolaget LM Ericsson
Joakim Persson (Section Owner)	Telefonaktiebolaget LM Ericsson

Previous versions [Baseband Timers, appendix]

David E. Cyper	NIST
Jaap Haartsen (Section Owner)	Telefonaktiebolaget LM Ericsson
Joakim Persson	Telefonaktiebolaget LM Ericsson
Ayse Findikli	Zeevo, Inc.

2.3.3 Part C: Link Manager Protocol

Version 1.2

Jennifer Bray	CSR
Robin Heydon	CSR
Simon Morris	CSR
Alexander Thoukydides	CSR
Kim Schneider	Digianswer/Motorola
Knud Dyring-Olsen	Digianswer/Motorola
Henrik Andersen	Digianswer/Motorola
Jan Åberg	Ericsson
Martin van der Zee	Ericsson
Roland Hellfajer	Infineon
YC Maa	Integrated Programmable Communications, Inc.
Steve McGowan	Intel
Tod Sizer	Lucent Technologies
Adrian Stephens	Mobilian
Jürgen Schnitzler	Nokia
Thomas Müller	Nokia
Carmen Kuhl	Nokia
Arto Palin	Nokia
Thomas Müller	Nokia
Roland Matthijssen	Philips
Rob Davies	Philips
Harmke de Groot	Philips
Antonio Salloum	Philips
Joel Linsky	Silicon Wave
Terry Bourk	Silicon Wave
Yariv Raveh	Texas Instruments
Tally Shina	Texas Instruments
Amihai Kidron	Texas Instruments
Yoshimitsu Shimojo	Toshiba
Toshiki Kizu	Toshiba
John Mersh (section owner)	TTPCom
Sam Turner	TTPCom

Previous versions

Kim Schneider	Digianswer A/S
Toru Aihara	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Kris Fleming	Intel Corporation
David E. Cypher	NIST
Thomas Busse	Nokia Corporation
Julien Corthial	Nokia Corporation
Olaf Joeressen	Nokia Corporation
Thomas Müller	Nokia Corporation
Dong Nguyen	Nokia Corporation
Harmke de Groot	Philips
Terry Bourk	Silicon Wave
Johannes Elg	Telefonaktiebolaget LM Ericsson
Jaap Haartsen	Telefonaktiebolaget LM Ericsson
Tobias Melin (Section Owner)	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Onn Haran	Texas Instruments
John Mersh	TTPCom

2.3.4 Part D: Error Codes

Version 1.2

Robin Heydon (section owner)	CSR
Roland Hellfajer	Infineon
Joel Linsky	Silicon Wave
John Mersh	TTPCom

This part was earlier included in the LMP and HCI functional Specifications.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 66

2.3.5 Part E: Bluetooth Host Controller Interface Functional Specification

Version 1.2

Robin Heydon (section owner	CSR
Jennifer Bray	CSR
Alexander Thoukydides	CSR
Knud Dyring-Olsen	Digianswer/Motorola
Henrik Andersen	Digianswer/Motorola
Jan Åberg	Ericsson
Martin van der Zee	Ericsson
Don Liechty	Extended Systems
Kevin Marquess	Hyper Corp
Roland Hellfajer	Infineon
YC Maa	Integrated Programmable Communications, Inc.
Steve McGowan	Intel
Tod Sizer	Lucent Technologies
Tsuyoshi Okada	Matsushita Electric Industrial Co. Ltd
Andy Glass	Microsoft
Adrian Stephens	Mobilian
Len Ott	Motorola
Jürgen Schnitzler	Nokia
Thomas Müller	Nokia
Rene Tischer	Nokia
Rob Davies	Philips
Antonio Salloum	Philips
Joel Linsky	Silicon Wave
Terry Bourk	Silicon Wave
Len Ott	Socket Communications
Randy Erman	Taiyo Yuden
Yoshimitsu Shimojo	Toshiba
Toshiki Kizu	Toshiba
Katsuhiro Kinoshita	Toshiba
Sam Turner	TTPCom
John Mersh	TTPCom

Previous versions

Todor Cooklev	3Com Corporation
Toru Aihara	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Nathan Lee	IBM Corporation
Akihiko Mizutani	IBM Corporation
Les Cline	Intel Corporation
Bailey Cross	Intel Corporation
Kris Fleming	Intel Corporation
Robert Hunter	Intel Corporation
Jon Inouye	Intel Corporation

Srikanth Kambhatla	Intel Corporation
Steve Lo	Intel Corporation
Vijay Suthar	Intel Corporation
Bruce P. Kraemer	Intersil
Greg Muchnik	Motorola, Inc.
David E. Cypher	NIST
Thomas Busse	Nokia Corporation
Julien Courthial	Nokia Corporation
Thomas Müller	Nokia Corporation
Dong Nguyen	Nokia Corporation
Jürgen Schnitzler	Nokia Corporation
Fujio Watanabe	Nokia Corporation
Christian Zechlin	Nokia Corporation
Johannes Elg	Telefonaktiebolaget LM Ericsson
Christian Johansson (Section Owner)	Telefonaktiebolaget LM Ericsson
Patrik Lundin	Telefonaktiebolaget LM Ericsson
Tobias Melin	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Thomas M. Siep	Texas Instruments
Masahiro Tada	Toshiba Corporation
John Mersh	TTPCom

2.3.6 Part F: Message Sequence Charts

Version 1.2

Tom Siep	Bluetooth SIG Inc.
Robin Heydon (section owner)	CSR
Simon Morris	CSR.
Jan Åberg	Ericsson
Christian Gehrmann	Ericsson
Joel Linsky	Silicon Wave
John Mersh	TTPCom

Previous versions

Todor Cooklev	3Com Corporation
Toru Aihara	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Nathan Lee	IBM Corporation
Kris Fleming	Intel Corporation.
Greg Muchnik	Motorola, Inc.
David E. Cypher	NIST
Thomas Busse	Nokia Corporation
Dong Nguyen (Section Owner)	Nokia Corporation
Fujio Watanabe	Nokia Corporation
Christian Johansson	Telefonaktiebolaget LM Ericsson
Tobias Melin	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments

Petitioner Kiosoft Exhibit 1011 Part 1
Page 68

2.3.7 Part G: Sample Data

Version 1.2

Joel Linsky	Silicon Wave

Previous versions

Thomas Müller	Nokia Corporation
Thomas Sander	Nokia Corporation
Joakim Persson (Section Owner)	Telefonaktiebolaget LM Ericsson

2.3.8 Part H: Security Specification

Please see Part B (Section 2.3.2 on page 62). The Security specification was until recently a part of the Baseband specification.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 69

2.4 [VOL 3] CORE SYSTEM PACKAGE, HOST

2.4.1 Part A: Logical Link Control and Adaptation Protocol Specification

Version 1.2

Tom Siep	Bluetooth SIG Inc.
Carsten Andersen (section owner	CSR
Jennifer Bray	CSR
Jan Åberg	Ericsson
Martin van der Zee	Ericsson
Sam Ravnborg	Ericsson
Stefan Agnani	Ericsson
Steve McGowan	Intel
Joby Lafky	Microsoft
Doron Holan	Microsoft
Andy Glass	Microsoft
Brian Redding	Motorola
Jürgen Schnitzler	Nokia
Thomas Müller	Nokia
Rob Davies	Philips
Terry Bourk	Silicon Wave
Michael Hasling	Tality

Previous versions

Jon Burgess	3Com Corporation
Paul Moran	3Com Corporation
Doug Kogan	Extended Systems
Kevin D. Marquess	Hyper Corporation
Toru Aihara	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Kris Fleming	Intel Corporation
Uma Gadamsetty	Intel Corporation
Robert Hunter	Intel Corporation
Jon Inouye	Intel Corporation
Steve C. Lo	Intel Corporation
Chunrong Zhu	Intel Corporation
Sergey Solyanik	Microsoft Corporation
David E. Cypher	NIST
Nada Golmie	NIST
Thomas Busse	Nokia Corporation
Rauno Makinen	Nokia Corporation
Thomas Müller	Nokia Corporation
Petri Nykänen	Nokia Corporation
Peter Ollikainen	Nokia Corporation
Petri O. Nurminen	Nokia Corporation

Petitioner Kiosoft Exhibit 1011 Part 1
Page 70

Johannes Elg	Telefonaktiebolaget LM Ericsson
Jaap Haartsen	Telefonaktiebolaget LM Ericsson
Elco Nijboer	Telefonaktiebolaget LM Ericsson
Ingemar Nilsson	Telefonaktiebolaget LM Ericsson
Stefan Runesson	Telefonaktiebolaget LM Ericsson
Gerrit Slot	Telefonaktiebolaget LM Ericsson
Johan Sörensen	Telefonaktiebolaget LM Ericsson
Goran Svennarp	Telefonaktiebolaget LM Ericsson
Mary A. DuVal	Texas Instruments
Thomas M. Siep	Texas Instruments
Kinoshita Katsuhiro	Toshiba Corporation

2.4.2 Part B: Service Discovery Protocol (SDP)

Version 1.2

Previous versions

Ned Plasson	3Com Corporation
John Avery	Convergence
Jason Kronz	Convergence
Chatschik Bisdikian	IBM Corporation
Parviz Kermani	IBM Corporation
Brent Miller	IBM Corporation
Dick Osterman	IBM Corporation
Bob Pascoe	IBM Corporation
Jon Inouye	Intel Corporation
Srikanth Kambhatla	Intel Corporation
Jay Eaglstun	Motorola, Inc.
Dale Farnsworth (Section Owner)	Motorola, Inc.
Jean-Michel Rosso	Motorola, Inc.
Jan Grönholm	Nokia Corporation
Kati Rantala	Nokia Corporation
Thomas Müller	Nokia Corporation
Johannes Elg	Telefonaktiebolaget LM Ericsson
Kazuaki Iwamura	Toshiba Corporation

2.4.3 Part C Generic Access Profile

Version 1.2

Jennifer Bray	CSR
Alexander Thoukydides	CSR
Christian Gehrmann	Ericsson
Henrik Hedlund	Ericsson
Jan Åberg	Ericsson
Stefan Agnani	Ericsson
Thomas Müller	Nokia
Joel Linsky	Silicon Wave
Terry Bourk	Silicon Wave
Katsuhiro Kinoshita	Toshiba

Previous versions

Ken Morley	3Com Corporation
Chatschik Bisdikian	IBM Corporation
Jon Inouye	Intel Corporation
Brian Redding	Motorola, Inc.
David E. Cypher	NIST
Stephane Bouet	Nokia Corporation
Thomas Müller	Nokia Corporation
Martin Roter	Nokia Corporation
Johannes Elg	Telefonaktiebolaget LM Ericsson
Patric Lind (Section Owner)	Telefonaktiebolaget LM Ericsson
Erik Slotboom	Telefonaktiebolaget LM Ericsson
Johan Sörensen	Telefonaktiebolaget LM Ericsson

2.4.4 Part D: Test Support

Version 1.2

Emilio Mira Escartis	Cetecom
Robin Heydon	CSR
Jennifer Bray	CSR
Stefan Agnani (section owner)	Ericsson
Terry Bourk	Silicon Wave
Joel Linsky	Silicon Wave
Michael Hasling	Tality

Previous versions [Test Mode]

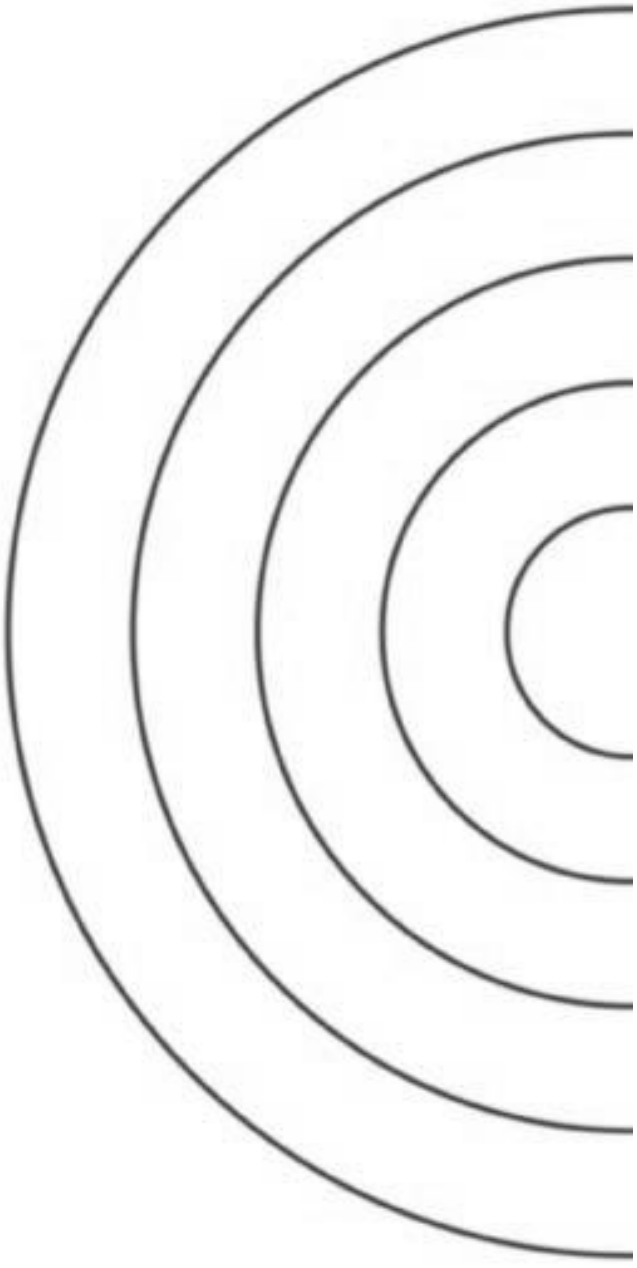
Jeffrey Schiffer	Intel Corporation
David E. Cypher	NIST
Daniel Bencak	Nokia Corporation
Arno Kefenbaum	Nokia Corporation
Thomas Müller (Section Owner)	Nokia Corporation

Roland Schmale	Nokia Corporation
Fujio Watanabe	Nokia Corporation
Stefan Agnani	Telefonaktiebolaget LM Ericsson
Mårten Mattsson	Telefonaktiebolaget LM Ericsson
Tobias Melin	Telefonaktiebolaget LM Ericsson
Lars Nord	Telefonaktiebolaget LM Ericsson
Fredrik Töörn	Telefonaktiebolaget LM Ericsson
John Mersh	TTPCom
Ayse Findikli	Zeevo, Inc.

Previous versions [Test Control Interface]

Mike Feldman	Motorola, Inc.
Thomas Müller	Nokia Corporation
Stefan Agnani (Section Owner)	Telefonaktiebolaget LM Ericsson
Mårten Mattsson	Telefonaktiebolaget LM Ericsson
Dan Sönnerstam	Telefonaktiebolaget LM Ericsson

Petitioner Kiosoft Exhibit 1011 Part 1
Page 74





Specification of the Bluetooth System

Wireless connections made easy

Architecture & Terminology Overview

Version 1.2
05 November 2003

Bluetooth®

Revision History

The Revision History is shown in the “Appendix” on page 53[vol. 0].

Contributors

The persons who contributed to this specification are listed in the , “Appendix” on page 53[vol. 0].

Web Site

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

Disclaimer and Copyright Notice

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

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 78

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

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

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

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

Copyright © 1999, 2000, 2001, 2002, 2003

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 80

TABLE OF CONTENTS

Part A
ARCHITECTURE

Contents **11**

1 General Description **13**
1.1 Overview of Operation 13
1.2 Nomenclature 15

2 Core System Architecture **19**
2.1 Core Architectural Blocks 22
2.1.1 Channel manager 22
2.1.2 L2CAP resource manager 22
2.1.3 Device manager 23
2.1.4 Link manager 23
2.1.5 Baseband resource manager 23
2.1.6 Link controller 24
2.1.7 RF 24

3 Data Transport Architecture **25**
3.1 Core Traffic Bearers 26
3.1.1 Framed data traffic 27
3.1.2 Unframed data traffic 28
3.1.3 Reliability of traffic bearers 28
3.2 Transport Architecture Entities 30
3.2.1 Bluetooth generic packet structure 30
3.3 Physical Channels 32
3.3.1 Basic piconet channel 33
3.3.1.1 Overview 33
3.3.1.2 Characteristics 33
3.3.1.3 Topology 34
3.3.1.4 Supported layers 34
3.3.2 Adapted piconet channel 34
3.3.2.1 Overview 34
3.3.3 Inquiry scan channel 35
3.3.3.1 Overview 35
3.3.3.2 Characteristics 35
3.3.3.3 Topology 36
3.3.3.4 Supported layers 36

Petitioner Kiosoft Exhibit 1011 Part 1
Page 81

3.3.4 Page scan channel 36
3.3.4.1 Overview 36
3.3.4.2 Characteristics 36
3.3.4.3 Topology 37
3.3.4.4 Supported layers 37
3.4 Physical Links 37
3.4.1 Links supported by the basic and adapted piconet physical channel 38
3.4.1.1 Active physical link 38
3.4.1.2 Parked physical link 38
3.4.2 Links supported by the scanning physical channels 39
3.5 Logical Links and Logical Transports 39
3.5.1 Casting 41
3.5.2 Scheduling and acknowledgement scheme 41
3.5.3 Class of data 42
3.5.4 Asynchronous connection-oriented (ACL) 42
3.5.5 Synchronous connection-oriented (SCO) 43
3.5.6 Extended synchronous connection-oriented (eSCO) 44
3.5.7 Active slave broadcast (ASB) 44
3.5.8 Parked slave broadcast (PSB) 45
3.5.9 Logical links 46
3.5.10 ACL Control Logical Link (ACL-C) 47
3.5.11 User Asynchronous/Isochronous Logical Link (ACL-U) 47
3.5.12 User Synchronous/Extended Synchronous Logical Links (SCO-S/eSCO-S) 47
3.6 L2CAP Channels 48
4 Communication Topology 49
4.1 Piconet Topology 49
4.2 Operational Procedures and Modes 51
4.2.1 Inquiry (Discovering) Procedure 51
4.2.2 Paging (Connecting) Procedure 52
4.2.3 Connected mode 52
4.2.4 Hold mode 53
4.2.5 Sniff mode 53
4.2.6 Parked state 54
4.2.7 Role switch procedure 54

Petitioner Kiosoft Exhibit 1011 Part 1
Page 82

Part B
ACRONYMS & ABBREVIATIONS

1 List of Acronyms and Abbreviations....57

2 Abbreviations of the Specification Names 65

Part C
CHANGES FROM BLUETOOTH SPECIFICATION V 1.1

Contents 69

1 New Features....71

2 Changes in Wording 73

2.1 IEEE Language Update 73

2.1.1 Shall 74

2.1.2 Must....74

2.1.3 Will....74

2.1.4 Should 74

2.1.5 May....75

2.1.6 Can....75

2.2 Nomenclature Changes 75

3 Structure Changes 77

4 Deprecated Specifications 79

4.1 Deprecated Features 79

4.2 Deprecated Profiles....79

Petitioner Kiosoft Exhibit 1011 Part 1
Page 83

Petitioner Kiosoft Exhibit 1011 Part 1
Page 84



Architecture



Petitioner Kiosoft Exhibit 1011 Part 1
Page 86

CONTENTS

1 General Description 13
1.1 Overview of Operation 13
1.2 Nomenclature 15
2 Core System Architecture 19
2.1 Core Architectural Blocks 22
2.1.1 Channel manager 22
2.1.2 L2CAP resource manager 22
2.1.3 Device manager 23
2.1.4 Link manager 23
2.1.5 Baseband resource manager 23
2.1.6 Link controller 24
2.1.7 RF 24
3 Data Transport Architecture 25
3.1 Core Traffic Bearers 26
3.1.1 Framed data traffic 27
3.1.2 Unframed data traffic 28
3.1.3 Reliability of traffic bearers 28
3.2 Transport Architecture Entities 30
3.2.1 Bluetooth generic packet structure 30
3.3 Physical Channels 32
3.3.1 Basic piconet channel 33
3.3.1.1 Overview 33
3.3.1.2 Characteristics 33
3.3.1.3 Topology 34
3.3.1.4 Supported layers 34
3.3.2 Adapted piconet channel 34
3.3.2.1 Overview 34
3.3.3 Inquiry scan channel 35
3.3.3.1 Overview 35
3.3.3.2 Characteristics 35
3.3.3.3 Topology 36
3.3.3.4 Supported layers 36
3.3.4 Page scan channel 36
3.3.4.1 Overview 36
3.3.4.2 Characteristics 36
3.3.4.3 Topology 37
3.3.4.4 Supported layers 37

3.4 Physical Links 37
3.4.1 Links supported by the basic and adapted piconet physical channel 38
3.4.1.1 Active physical link 38
3.4.1.2 Parked physical link 38
3.4.2 Links supported by the scanning physical channels 39
3.5 Logical Links and Logical Transports 39
3.5.1 Casting 41
3.5.2 Scheduling and acknowledgement scheme 41
3.5.3 Class of data 42
3.5.4 Asynchronous connection-oriented (ACL) 42
3.5.5 Synchronous connection-oriented (SCO) 43
3.5.6 Extended synchronous connection-oriented (eSCO) 44
3.5.7 Active slave broadcast (ASB) 44
3.5.8 Parked slave broadcast (PSB) 45
3.5.9 Logical links 46
3.5.10 ACL Control Logical Link (ACL-C) 47
3.5.11 User Asynchronous/Isochronous Logical Link (ACL-U) 47
3.5.12 User Synchronous/Extended Synchronous Logical Links (SCO-S/eSCO-S) 47
3.6 L2CAP Channels 48

4 Communication Topology 49

4.1 Piconet Topology 49
4.2 Operational Procedures and Modes 51
4.2.1 Inquiry (Discovering) Procedure 51
4.2.2 Paging (Connecting) Procedure 52
4.2.3 Connected mode 52
4.2.4 Hold mode 53
4.2.5 Sniff mode 53
4.2.6 Parked state 54
4.2.7 Role switch procedure 54

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. Key features are robustness, low power, and low cost. Many features of the core specification are optional, allowing product differentiation.

The Bluetooth core system consists of an RF transceiver, baseband, and protocol stack. The system offers services that enable the connection of devices and the exchange of a variety of classes of data between these devices.

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

1.1 OVERVIEW OF OPERATION

The Bluetooth RF (physical layer) 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. RF operation uses a shaped, binary FM 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).

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 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 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 in the ISM band. The hopping pattern may be adapted to exclude a portion of the frequencies that are used by interferring devices. The adaptive hopping technique improves Bluetooth co-existence with static (non-hopping) ISM systems when these 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 32 - Section 3.6 on page 48 but are introduced here to aid the understanding of the remainder of this section.

Within a physical channel, a physical link is formed between any two devices that transmit packets in either direction between them. In a piconet physical channel 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 signalling. 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 90

1.2 NOMENCLATURE

Where the following terms appear in the specification they have the meaning given in Table 1.1 on page 15.

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.
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. See Section 3.5.7 on page 44
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 Bluetooth 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 sub-system containing the Bluetooth RF, baseband, resource controller, link manager, device manager and a Bluetooth HCI.
Bluetooth Device	A Bluetooth Device is 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.
BD_ADDR	The Bluetooth Device Address, BD_ADDR, is used to identify a Bluetooth device.
Bluetooth HCI	The Bluetooth 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.

Table 1.1: Nomenclature.

Bluetooth Host	A Bluetooth Host is a computing device, peripheral, cellular telephone, access point to PSTN network or LAN, etc. A Bluetooth Host attached to a Bluetooth Controller may communicate with other Bluetooth Hosts attached to their Bluetooth Controllers as well.
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 Bluetooth device in range that periodically listens on its page scan physical channel and will respond to a page on that channel.
Connected devices	Two Bluetooth devices in the same piconet and with a physical link between them.
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.)
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.
coverage area	The area where two Bluetooth devices can exchange messages with acceptable quality and performance.
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 and pairing (if the 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 Bluetooth device in range that periodically listens on an inquiry scan physical channel and will respond to an inquiry on that channel. Discoverable device are normally also connectable.
Inquiring device	A Bluetooth device that is carrying out the inquiry 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.

Table 1.1: Nomenclature.

Inquiry scan	A procedure where a Bluetooth device listens for inquiry messages received on its inquiry scan physical channel.
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 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.
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 in order to authenticate each device to the other
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 Channel	Identical to an L2CAP channel, but deprecated due to an alternative meaning in Bluetooth 1.1
Logical link	The lowest architectural level used to offer independent data transport services to clients of the Bluetooth system.
Logical transport	Used in Bluetooth to represent commonality between different logical links due to shared acknowledgement protocol and link identifiers.
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.
Paired device	A Bluetooth device with which a link key has been exchanged (either before connection establishment was requested or during connecting phase).

Table 1.1: Nomenclature.

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.
Physical Channel	Characterised 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-level connection between two devices established using paging.
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 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.
Piconet Master	The device in a piconet whose Bluetooth Clock and Bluetooth Device Address are used to define the piconet physical channel characteristics.
Piconet Slave	Any 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 paring 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. Section 3.5.8 on page 45.
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 device appears as silent to a remote device if it does not respond to inquiries made by the remote device.
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 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). 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 19 except for SDP that is not shown for clarity.

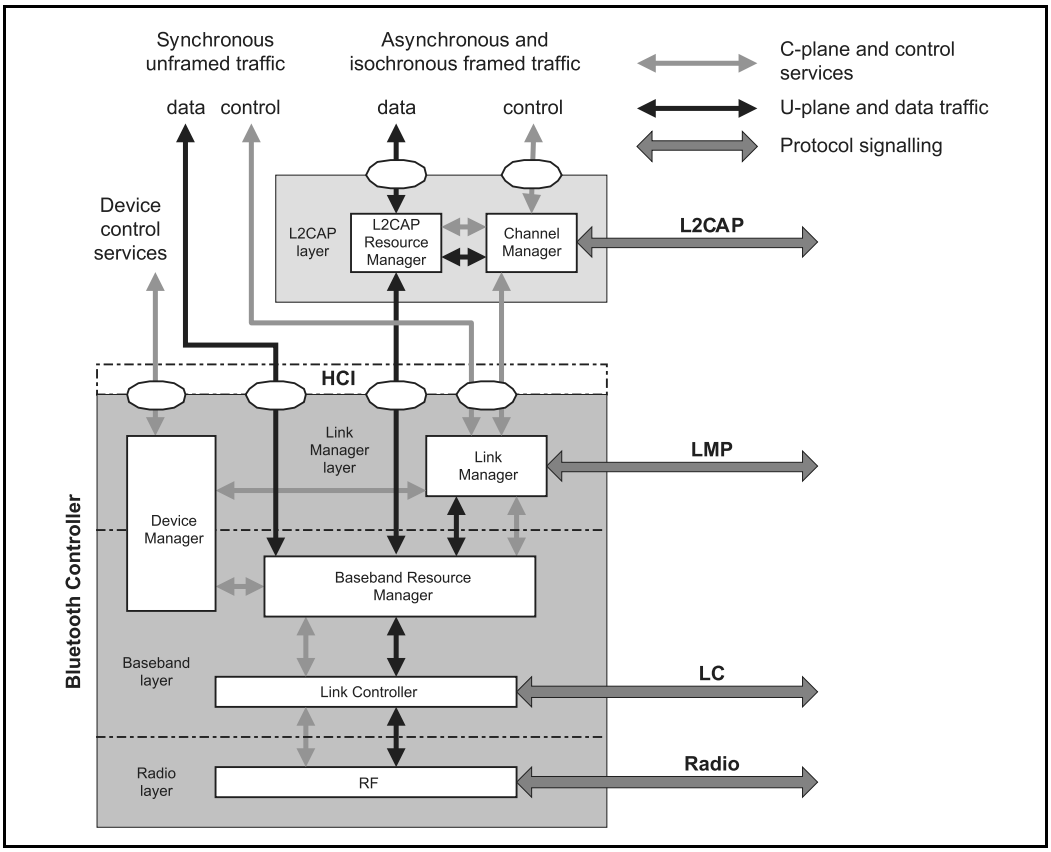


Figure 2.1: Bluetooth core system architecture

Figure 2.1 on page 19 shows the four lowest layers, each with its associated communication protocol. The lowest three layers are sometimes grouped into a subsystem known as the Bluetooth controller. This is a common implementation involving a standard physical communications interface between the Bluetooth controller and remainder of the Bluetooth system including the L2CAP, service and higher layers (known as the Bluetooth host.) Although this interface is optional the architecture is designed to allow for its existence and characteristics. The Bluetooth specification enables inter-operability between independent Bluetooth systems by defining the protocol messages exchanged between equivalent layers, and also inter-operability between independent

Bluetooth sub-systems by defining a common interface between Bluetooth controllers and Bluetooth hosts.

A number of functional blocks are shown and the path of services and data between these. 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 inter-operability. Thus the functional blocks in Figure 2.1 on page 19 are shown in order to aid description of the system behaviour. An implementation may be different from the system shown in Figure 2.1 on page 19.

Standard interactions are defined for all inter-device operation, where Bluetooth devices exchange protocol signalling according to the Bluetooth specification. The Bluetooth core system protocols are the Radio (RF) protocol, Link Control (LC) protocol, Link Manager (LM) protocol and Logical Link Control and Adaptation protocol (L2CAP), all of that are fully defined in subsequent parts of the Bluetooth specification. In addition the Service Discovery Protocol (SDP) is a service layer protocol required by all 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 behaviour 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 sub-system is defined such that the Bluetooth controller may be considered a standard part. In this configuration the Bluetooth controller operates the lowest three layers and the L2CAP layer is contained with the rest of the Bluetooth application in a host system. The standard interface is called the Host to Controller Interface (HCI) and its service access points are represented by the ellipses on the upper edge of the Bluetooth controller sub-system in Figure 2.1 on page 19. Implementation of this standard service interface is optional.

As the Bluetooth architecture is defined with the possibility of separate host and controller communicating through an HCI, a number of general assumptions are made. The Bluetooth controller is 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 96

The Baseband layer provides the basic 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.

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 RF interface, which is common to all Bluetooth systems, and through the Test Control Interface (TCI), which is only required for conformance testing.

The tester uses exchanges with the Implementation Under Test (IUT) through the RF interface to ensure the correct responses to requests from remote devices. The tester controls the IUT through the TCI to cause the IUT to originate exchanges through the RF 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 Bluetooth 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 19. 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 behavioural aspects of the system outlined below and specified in subsequent parts of the Bluetooth specification.

2.1.1 Channel manager

The channel manager is responsible for creating, managing and destroying 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 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.

2.1.2 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 Bluetooth controller resource exhaustion. This is required because the architectural model does not assume that the Bluetooth 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 98

2.1.3 Device manager

The device manager is the functional block in the baseband that controls the general behaviour of the Bluetooth device. It is responsible for all operation of the Bluetooth system that is not directly related to data transport, such as inquiring for the presence of other nearby Bluetooth devices, connecting to other 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 behaviour implied by a number of the HCI commands, such as managing the device local name, any stored link keys, and other functionality.

2.1.4 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.)

The LM 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 on the physical link, or the adjustment of QoS settings for a logical link.

2.1.5 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 behaviour that requires use of the Bluetooth radio. 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 the scheduling of a logical link results in changing 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.

2.1.6 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 signalling (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 signalling is normally associated with the resource manager's scheduler.

2.1.7 RF

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 100

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 25.

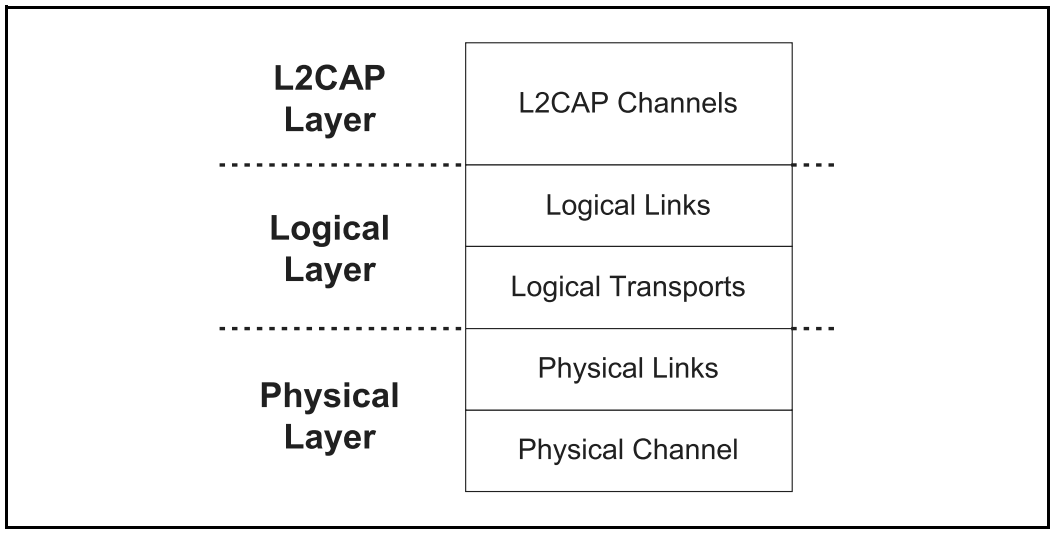


Figure 3.1: Bluetooth generic data transport architecture

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 behaviour.)

The Bluetooth 1.1 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 behaviour.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 101

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 26 below (for ease of representation this is shown with higher layers to the left and lower layers to the right).

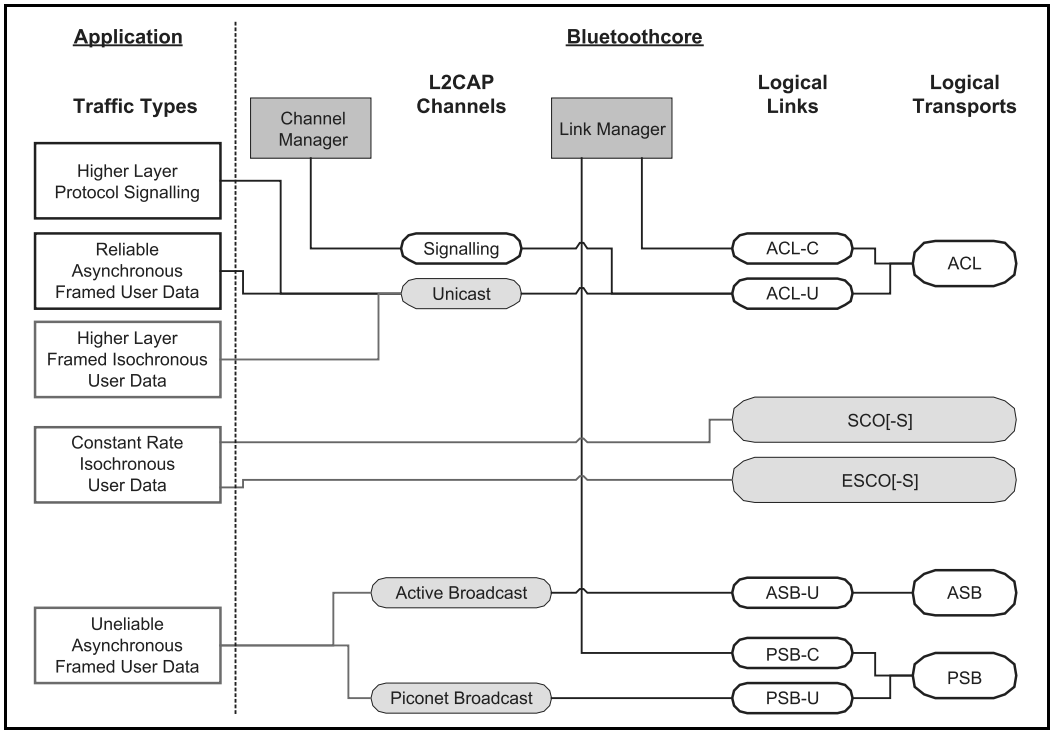


Figure 3.2: Bluetooth traffic bearers

The core traffic bearers that are available to applications are shown in Figure 3.2 on page 26 as the shaded rounded rectangles. The architectural layers that are defined to provide these services are described in Section 2 on page 19. 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 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, or the ACL-U logical link when the L2CAP layer is being discussed.

The mapping of application traffic types to Bluetooth core traffic bearers in Figure 3.2 on page 26 is based on matching the traffic characteristics with the

Petitioner Kiosoft Exhibit 1011 Part 1
Page 102

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 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 26 is only acceptable if the characteristics of the application traffic type are preserved.

Figure 3.2 on page 26 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 L2CAP channels may be created for transport of unicast (point-to-point) data between two Bluetooth devices. A connectionless L2CAP channel exists for broadcasting 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. Traffic on the broadcast L2CAP channel is uni-directional. Unicast L2CAP channels may be uni-directional or bi-directional.

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.

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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 103

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. 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 number 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.

3.1.3 Reliability of traffic bearers

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 acknowledgement/repeat request (ARQ) protocol. This provides an enhanced reliability by re-transmitting 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 104

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 characterised 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.

3.2 TRANSPORT ARCHITECTURE ENTITIES

The Bluetooth transport architecture entities are shown in Figure 3.3 on page 30 and are described from the lowest layer upwards in the subsequent sections.

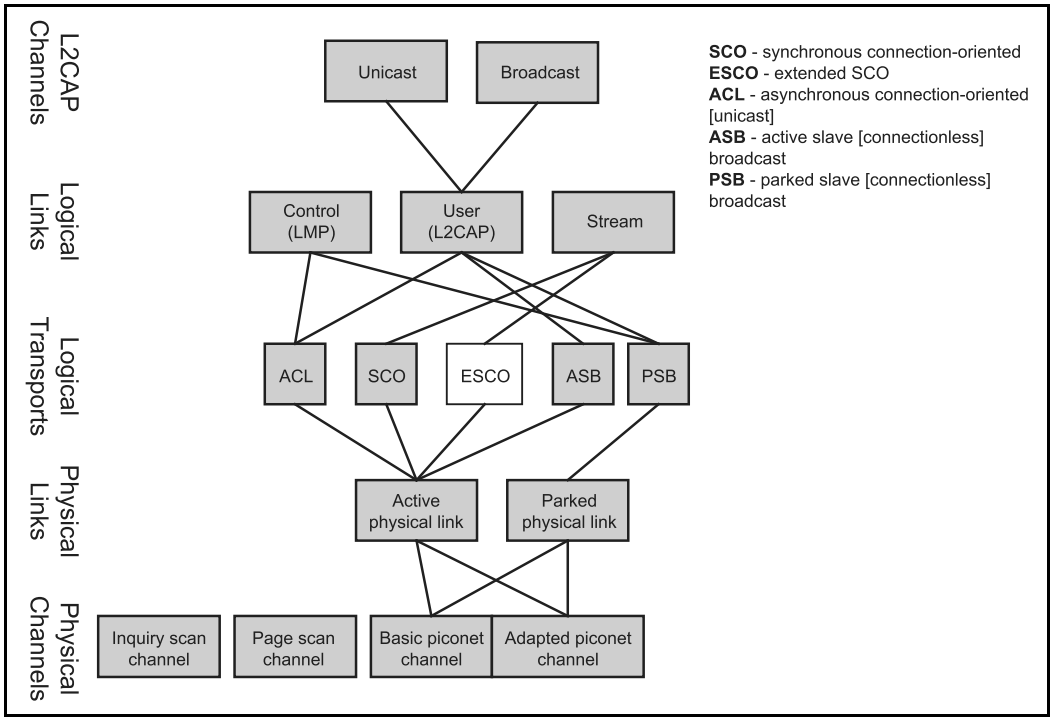


Figure 3.3: Overview of transport architecture entities and hierarchy

3.2.1 Bluetooth generic packet structure

The general packet structure nearly reflects the architectural layers found in the Bluetooth system. The packet structure is designed for optimal use in normal operation. It is shown in Figure 3.4 on page 30.

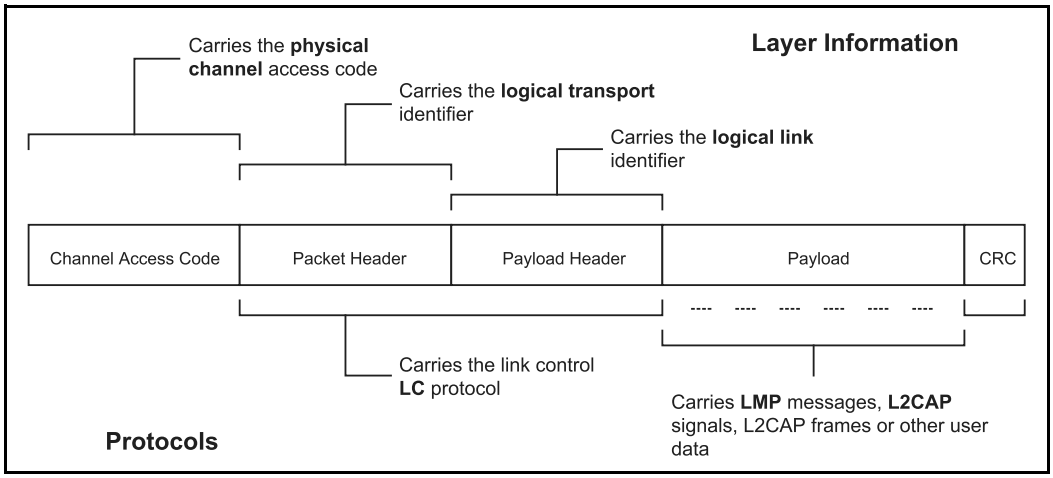


Figure 3.4: Bluetooth packet structure

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 Bluetooth 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 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 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 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.

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.

3.3 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 characterised by an RF 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.

Two Bluetooth devices use a shared physical channel for communication. To achieve this their transceivers need to be tuned to the same RF 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 always present at the start of every transmitted packet.

Four Bluetooth physical channels are defined. Each is optimised 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 physical channels are used for discovering Bluetooth devices (the inquiry scan channel) and for connecting Bluetooth devices (the page scan channel.)

A Bluetooth device can only use one of these 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 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 capable of connecting simultaneously to more than one physical channel, but the specification does not assume that this is possible.

3.3.1 Basic piconet channel

3.3.1.1 Overview

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

3.3.1.2 Characteristics

The basic piconet channel is characterised by a pseudo-random sequence hopping through the RF channels. The hopping sequence is unique for the piconet and is determined by the Bluetooth device address of the master. The phase in the hopping sequence is determined by the Bluetooth clock of the master. 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 RF hop frequency. Consecutive hops correspond to different RF 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 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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 109

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.

3.3.1.3 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.

3.3.1.4 Supported layers

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

3.3.2 Adapted piconet channel

3.3.2.1 Overview

The adapted piconet channel differs from the basic piconet channel in two ways. First the frequencies on which the slaves transmit are the same as the preceding master transmit frequency. In other words the frequency is not recomputed between master and subsequent slave packets. The second way in which the adapted piconet channel differs from the basic piconet channel is that the adapted type can 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 would have selected 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 110

3.3.3 Inquiry scan channel

3.3.3.1 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 responses to these requests. 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.

3.3.3.2 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 channel remains passive 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 of these devices in order 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. This is done at a faster rate, allowing the inquiring device to cover all inquiry scan frequencies in a reasonably short time period.

3.3.3.3 Topology

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

3.3.3.4 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.

3.3.4 Page scan channel

3.3.4.1 Overview

A connectable device (one that is prepared to accept connections) does so using an page scan channel. A connectable device listens for page requests on its page scan channel and 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 an page request on each frequency and listening for any response.

3.3.4.2 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 characterised 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 which case it is able to predict the phase of the target device's page scan channel. It may use this information to optimise the synchronization of the paging and page scanning process and speed up the formation of the connection.

3.3.4.3 Topology

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

3.3.4.4 Supported layers

During the exchange of packets between an 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.

3.4 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. The access code packet field, together with the clock and address of the master Bluetooth device are used to identify a physical channel. However 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.

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 such properties. In the case of physical links with modifiable properties the LM protocol is used to adapt these properties. As the LM protocol 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 signalling.

In the situation where a transmission is broadcasted 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 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 properties 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 behaviour. 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 signalling 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 signalling 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 signal 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 Links supported by the scanning physical channels

In the case of the 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 link are not further elaborated.

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 behaviour. 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 1.1 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) signalling are carried in the packet header, and for some logical links identification is carried in the

payload header. Control signalling that does not require single slot response times is carried out using the LMP protocol.

Table 3.1 on page 40 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	Overview
Asynchronous Connection-Oriented (ACL[1])	Control (LMP) ACL-C User (L2CAP) ACL-U	Active physical link, basic or adapted physical channel	Reliable or time-bounded, bi-directional, point-to-point.
Synchronous Connection-Oriented (SCO)	Stream (unframed) SCO-S	Active physical link, basic or adapted physical channel	Bi-directional, symmetric, point-to-point, AV channels. Used for 64Kb/s constant rate data.
Extended Synchronous Connection-Oriented (eSCO)	Stream (unframed) eSCO-S	Active physical link, basic or adapted physical channel	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	Active physical link, basic or adapted physical channel.	Unreliable, uni-directional broadcast to any devices synchronised with the physical channel. Used for broadcast L2CAP groups.
Parked slave broadcast (PSB)	Control (LMP) PSB-C, User (L2CAP) PSB-U	Parked physical link, basic or adapted physical channel.	Unreliable, uni-directional broadcast to all piconet devices. Used for LMP and L2CAP traffic to parked devices, and for access requests from parked devices.

Table 3.1: Logical transport types.

1. It is clear that the most obvious abbreviation for Asynchronous Connection-Oriented logical transport is ACO. However, this acronym has an alternative meaning from the Bluetooth 1.1 specification. To avoid confusion between two possible meanings for ACO the decision was made to retain the ACL abbreviation for the Asynchronous Connection-oriented Logical transport.

The names given to the logical links and logical transports reflect some of the names used in Bluetooth 1.1, in order to provide some degree of familiarity and continuation. However these names do not reflect a consistent scheme, which is outlined below.

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. There are no multicast links defined in Bluetooth 1.2,

- *Unicast links.* Unicast links exist between exactly two endpoints. Traffic may be sent in either direction on unicast links. All unicast links are connection-oriented, meaning that a connection procedure takes place before the link may be used. In the case of the default ACL links, the connection procedure is an implicit step within the general paging procedure used to form ad-hoc piconets.
- *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 that 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 in Bluetooth 1.2, though the default ACL link can be configured to operate in this fashion.

- *Synchronous links.* Synchronous links provide a method of associating the Bluetooth piconet clock with the transported data. 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) data or user data. The user data category is sub-divided 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 signalling 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 Asynchronous connection-oriented (ACL)

The asynchronous connection-oriented (ACL) logical transport is used to carry LMP and L2CAP control signalling and best effort asynchronous user data. The ACL logical transport uses a simple 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.

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 active 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.

3.5.5 Synchronous connection-oriented (SCO)

The synchronous connection-oriented (SCO) logical transport is a symmetric, point-to-point channel 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.

The reuse of the default ACL's LT_ADDR for SCO logical transports is due to legacy behaviour from the Bluetooth 1.1 specification. In this earlier version of Bluetooth the LT_ADDR (then known as the active member address) was used to identify the piconet member associated with each transmission. This was not easily extensible for enabling more logical links, and so the purpose of this field was redefined for the new features. Some Bluetooth 1.1 features, however, do not cleanly fit into the more formally described architecture.

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 signalling 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.

3.5.6 Extended synchronous connection-oriented (eSCO)

The extended synchronous connection-oriented (eSCO) logical transport is a symmetric or asymmetric, point-to-point link 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 signalling 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.

3.5.7 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 signalling or LMP control signalling.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 120

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.

3.5.8 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 signalling.) 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, modifi-

cations 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.

3.5.9 Logical links

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 122

page 26.) In particular the 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.

3.5.10 ACL Control Logical Link (ACL-C)

The ACL Control Logical Link (ACL-C) is used to carry LMP signalling 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.

3.5.11 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 favour of another L2CAP frame.)

3.5.12 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 lifetime 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. Bluetooth has some limitations when supporting variable-rate isochronous data concurrently with reliable user data.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 123

3.6 L2CAP CHANNELS

L2CAP provides a multiplexing role allowing many different applications to share the resources of an ACL-U logical link between two devices. 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.

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 logical links, and to carry out a simple level of scheduling, selecting SDUs according to relative priority.

L2CAP can provide a 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 124

4 COMMUNICATION TOPOLOGY

4.1 PICONET TOPOLOGY

Any time a Bluetooth link is formed it is within the context of a piconet. A piconet consists of two or more devices that occupy the same physical channel (which means that they are synchronized to 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.

Within a common location a number of independent piconets may exist. Each piconet has a different physical channel (that is a different master device and an independent piconet clock 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 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.

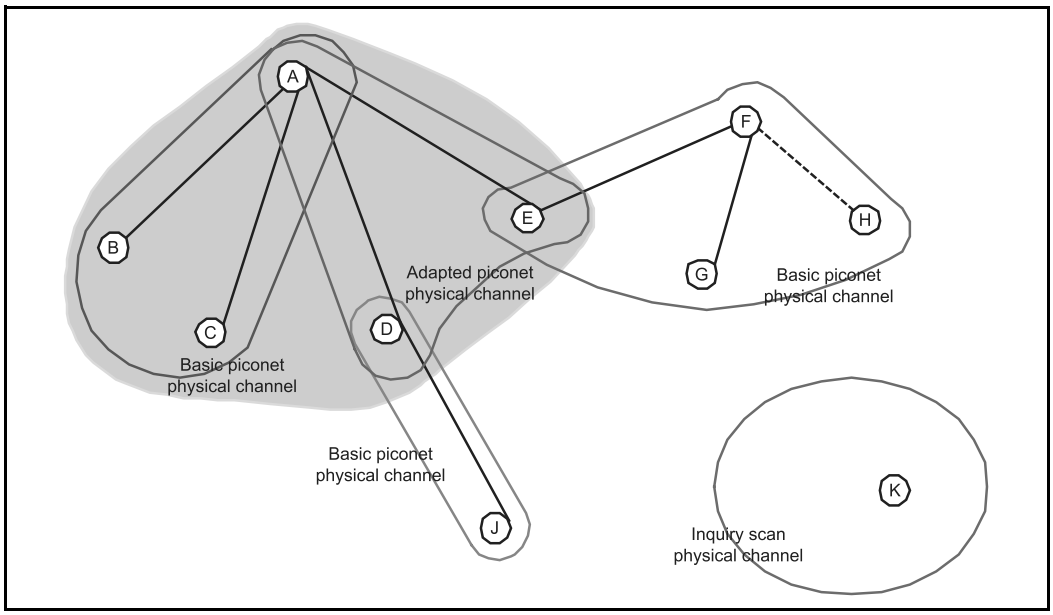


Figure 4.1: Example Bluetooth topology

In Figure 4.1 on page 50 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 126

Physical links (one per slave device) are represented in the diagram by lines connecting the devices. The solid lines represent an active physical link, and the dashed line represents a parked physical link. Device H is parked, and hence the physical link between the master (F) and the slave (H) is shown as parked.

Logical transports, logical links and L2CAP channels are used to provide capabilities for the transport of data, but are not shown on this diagram. They are described in more detail in Section 3.5 on page 39 and Section 3.6 on page 48.

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 that Bluetooth device. As Bluetooth is an ad-hoc wireless communications technology there are also 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 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.

4.2.2 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 in a piconet. 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.

4.2.3 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 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. Default ACL-C and ACL-U logical links are always created during the connection process, and these are used for LMP messages and the L2CAP signalling channel respectively.

It is noted that two default logical links are created when two units are initially connected. 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 signalling 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.

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 128

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.

4.2.4 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.

4.2.5 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.

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.

4.2.6 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.

4.2.7 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 are removed and rebuilt, with the exception of the ASB and PSB logical transports that are implied by the topology and are not preserved. 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 130



Acronyms & Abbreviations



Petitioner Kiosoft Exhibit 1011 Part 1
Page 132

1 LIST OF ACRONYMS AND ABBREVIATIONS

Acronym or abbreviation	Writing out in full	Which means
A		
ACK	Acknowledge	
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	
AFH	Adaptive Frequency Hopping	
AG	Audio Gateway	
AHS	Adapted Hop Sequence	
AR_ADDR	Access Request Address	
ARQ	Automatic Repeat reQuest	
ASB	Active Slave Broadcast [logical transport]	Unreliable, uni-directional broadcast to any devices synchronised with the physical channel.
ASB-U	ASB User [logical link] (L2CAP)	
B		
BB	BaseBand	
BCH	Bose, Chaudhuri & Hocquenghem	Type of code The persons who discovered these codes in 1959 (H) and 1960 (B&C)
BD_ADDR	Bluetooth Device Address	
BER	Bit Error Rate	
BT	Bandwidth Time	
BT	Bluetooth	
C		
CAC	Channel Access Code	
CC	Call Control	
CL	Connectionless	
CODEC	COder DECoder	

Table 1.1: Acronyms and Abbreviations.

Acronym or abbreviation	Writing out in full	Which means
COF	Ciphering Offset	
CRC	Cyclic Redundancy Check	
CVSD	Continuous Variable Slope Delta Modulation	
D		
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	
DH	Data-High Rate	Data packet type for high rate data
DIAC	Dedicated Inquiry Access Code	
DM	Data - Medium Rate	Data packet type for medium rate data
DTE	Data Terminal Equipment	In serial communications, DTE refers to a device at the endpoint of the communications path; typically a computer or terminal.
DTMF	Dual Tone Multiple Frequency	
DUT	Device Under Test	
DV	Data Voice	Data packet type for data and voice
E		
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
ETSI	European Telecommunications Standards Institute	

Table 1.1: Acronyms and Abbreviations.

Acronym or abbreviation	Writing out in full	Which means
F		
FCC	Federal Communications Commission	
FEC	Forward Error Correction code	
FH	Frequency Hopping	
FHS	Frequency Hop Synchronization	
FIFO	First In First Out	
FSK	Frequency Shift Keying	type of modulation
FW	Firmware	
G		
GFSK	Gaussian Frequency Shift Keying	
GIAC	General Inquiry Access Code	
GM	Group Management	
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	
IEEE	Institute of Electronic and Electrical Engineering	
IETF	Internet Engineering Task Force	
IP	Internet Protocol	
IrDA	Infra-red Data Association	
IrMC	Ir Mobile Communications	
ISDN	Integrated Services Digital Networks	
ISM	Industrial, Scientific, Medical	
IUT	Implementation Under Test	

Table 1.1: Acronyms and Abbreviations.

Acronym or abbreviation	Writing out in full	Which means
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	
LCSS	Link Controller Service Signalling	
LFSR	Linear Feedback Shift Register	
LLID	Logical Link Identifier	
LM	Link Manager	
LMP	Link Manager Protocol	For LM peer to peer communication
LSB	Least Significant Bit	
LT_ADDR	Logical Transport ADDRess	
M		
M	Master or Mandatory	
MAC	Medium Access Control	
MAPI	Messaging Application Procedure Interface	
MMI	Man Machine Interface	
MS	Mobile Station	
MS	Multiplexing sublayer	
MSB	Most Significant Bit	
MSC	Message Sequence Chart	
MTU	Maximum Transmission Unit	
N		
NAK	Negative Acknowledge	

Table 1.1: Acronyms and Abbreviations.

Acronym or abbreviation	Writing out in full	Which means
NAP	Non-significant Address Part	
O		
O	Optional	
OBEX	OBject EXchange protocol	
OCF	OpCode Command Field	
OGF	OpCode Group Field	
P		
PCM	Pulse Coded Modulation	
PCMCIA	Personal Computer Memory Card International Association	
PDU	Protocol Data Unit	a message
PIN	Personal Identification Number	
PM_ADDR	Parked Member Address	
PN	Pseudo-random Noise	
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)	
PSTN	Public Switched Telephone Network	
Q		
QoS	Quality of Service	
R		
RAND	Random number	
RF	Radio Frequency	
RFC	Request For Comments	
RFCMode	Retransmission and Flow Control Mode	

Table 1.1: Acronyms and Abbreviations.

Acronym or abbreviation	Writing out in full	Which means
RFCOMM		Serial cable emulation protocol based on ETSI TS 07.10
RSSI	Received Signal Strength Indication	
RX	Receive	
S		
S	Slave	
SAP	Service Access Points	
SAR	Segmentation and Reassembly	
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	
SDDB	Service Discovery Database	
SDP	Service Discovery Protocol	
SDU	Service Data Unit	
SEQN	Sequential Numbering scheme	
SRES	Signed Response	
SS	Supplementary Services	
SSI	Signal Strength Indication	
SUT	System Under Test	
SW	Software	
T		
TBD	To Be Defined	
TC	Test Control	Test Control layer for the test interface
TCI	Test Control Interface	
TCP/IP	Transport Control Protocol/Internet Protocol	
TCS	Telephony Control protocol Specification	

Table 1.1: Acronyms and Abbreviations.

Acronym or abbreviation	Writing out in full	Which means
TDD	Time-Division Duplex	
TX	Transmit	
U		
UAP	Upper Address Part	
UART	Universal Asynchronous receiver Transmitter	
UI	User Interface	
UI	Unnumbered Information	
ULAP	Upper and Lower Address Parts	
USB	Universal Serial Bus	
UUID	Universal Unique Identifier	
W		
WAP	Wireless Application Protocol	
WUG	Wireless User Group	

Table 1.1: Acronyms and Abbreviations.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 140

2 ABBREVIATIONS OF THE SPECIFICATION NAMES

Acronym or abbreviation	Writing out in full	Placement in Specification
A2DP	Advanced Audio Distribution Profile Specification	vol 10 part C
AVCTP	A/V Control Transport Protocol Specification	vol 10 part F
AVDTP	A/V Distribution Transport Profile Specification	vol 10 part A
AVRCP	A/V Remote Control Profile Specification	vol 10 part G
BB	Baseband Specification	vol 2 part B
BIP	Basic Imaging Profile	vol 8 part E
BNEP	Bluetooth Network Encapsulation Protocol Specification	vol 6 part A
BPP	Basic Printing Profile Specification	vol 8 part F
CIP	Common ISDN Access Profile Specification	vol 12 part A
CTP	Cordless Telephony Profile Specification	vol 9 part B
DUN	Dial-Up Networking Profile Specification	vol 7 part C
ESDP / UPNP	Extended Service Discovery Profile	vol 6 part D
FAX	Fax Profile Specification	vol 7 part D
FTP	File Transfer Profile Specification	vol 8 part C
GAP	Generic Access Profile Specification	vol 3 part C
GAVDP	Generic A/V Distribution Profile Specification	vol 10 part B
GOEP	Generic Object Exchange Profile Specification	vol 8 part A
HCI (1)	Host Controller Interface Functional Specification	vol 2 part E
HCI (2)	Host Controller Interface Transport Layers Specification	vol 4 part A-C
HCRP	Hardcopy Cable Replacement Profile Specification	vol 11 part B
HFP	Hands-Free Profile Specification	vol 7 part E
HID	Human Interface Device Profile Specification	vol 11 part A
HSP	Headset Profile Specification	vol 7 part F
ICP	Intercom Profile Specification	vol 9 part C
L2CAP	Logical Link Control and Adaptation Protocol Specification	vol 3 part A
LAP	LAN Access Profile Specification	deprecated

Table 2.1: Abbreviations of the specification names.

Acronym or abbreviation	Writing out in full	Placement in Specification
LMP	Link Manager Protocol Specification	vol 2 part C
MSC	Message Sequence Charts	vol 2 part F
OPP	Object Push Profile Specification	vol 8 part B
PAN	Personal Area Networking Profile Specification	vol 6 part B
RF	Radio Specification	vol 2 part A
RFCOMM	RFCOMM with TS 07.10	vol 7 part A
SAP	SIM Access Profile Specification	vol 12 part C
SDAP	Service Discovery Application Profile Specification	vol 5 part B
SDP (1)	Service Discovery Protocol Specification (server)	vol 3 part B
SDP (2)	Service Discovery Protocol Specification (client)	vol 5 part A
SPP	Serial Port Profile Specification	vol 7 part B
Synch	Synchronization Profile Specification	vol 8 part D
TCI	Test Control Interface	vol 3 part D, section 2
TCP	Telephony Control Protocol Specification	vol 9 part A
UDI	Unrestricted Digital Information Profile Specification	vol 12 part B

Table 2.1: Abbreviations of the specification names.



Changes from Bluetooth Specification v 1.1



CONTENTS

Contents 69

1 New Features 71

2 Changes in Wording 73

2.1 IEEE Language Update 73

2.1.1 Shall 74

2.1.2 Must 74

2.1.3 Will 74

2.1.4 Should 74

2.1.5 May 75

2.1.6 Can 75

2.2 Nomenclature Changes 75

3 Structure Changes 77

4 Deprecated Specifications 79

4.1 Deprecated Features 79

4.2 Deprecated Profiles 79

1 NEW FEATURES

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

- Architectual 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 vol2 and vol3.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 148

2 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.

2.1 IEEE LANGUAGE UPDATE

The purpose of changing terms used to decribe attributes of the Specification is to make it easy for the reader to identify text that decribes 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 Nomative. The general term for language that provides background and context for normative text is "informative". These terms have been added to 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 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.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 149

2.1.1 Shall

The word *shall* is used to indicate mandatory requirements strictly 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.

2.1.2 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”.

2.1.3 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”

2.1.4 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*.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 150

2.1.5 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.

2.1.6 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.

2.2 NOMENCLATURE CHANGES

The nomenclature in Bluetooth 1.2 has also been updated due to new concepts that are introduced together with the new features and the new architecture (see [Part A] Section 1.2 on page 15).

Petitioner Kiosoft Exhibit 1011 Part 1
Page 151

Petitioner Kiosoft Exhibit 1011 Part 1
Page 152

3 STRUCTURE CHANGES

The Bluetooth Core Specification 1.2 has been 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 Part A on page 9)

In addition, new text and requirements have been added for the new features as well as many changes throughout the specification to update the text to use IEEE language (see Section 2 on page 73).

Petitioner Kiosoft Exhibit 1011 Part 1
Page 154

4 DEPRECATED SPECIFICATIONS

As the Bluetooth Specification continues to evolve, some features, protocols, and profiles are replaced with new ways of performing the same function. Often these changes reflect the evolution of the communications industry. Some of the changes merely reflect an evolved understanding of the Bluetooth environment itself.

Those functions no longer recommended are being 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.

Specifications that have been deprecated will not be included in the updated Bluetooth Specifications, but the earlier versions are still available and may be used according to the rules set forth in the Qualification Policy (PRD).

4.1 DEPRECATED FEATURES

Features deprecated in version 1.2 are:

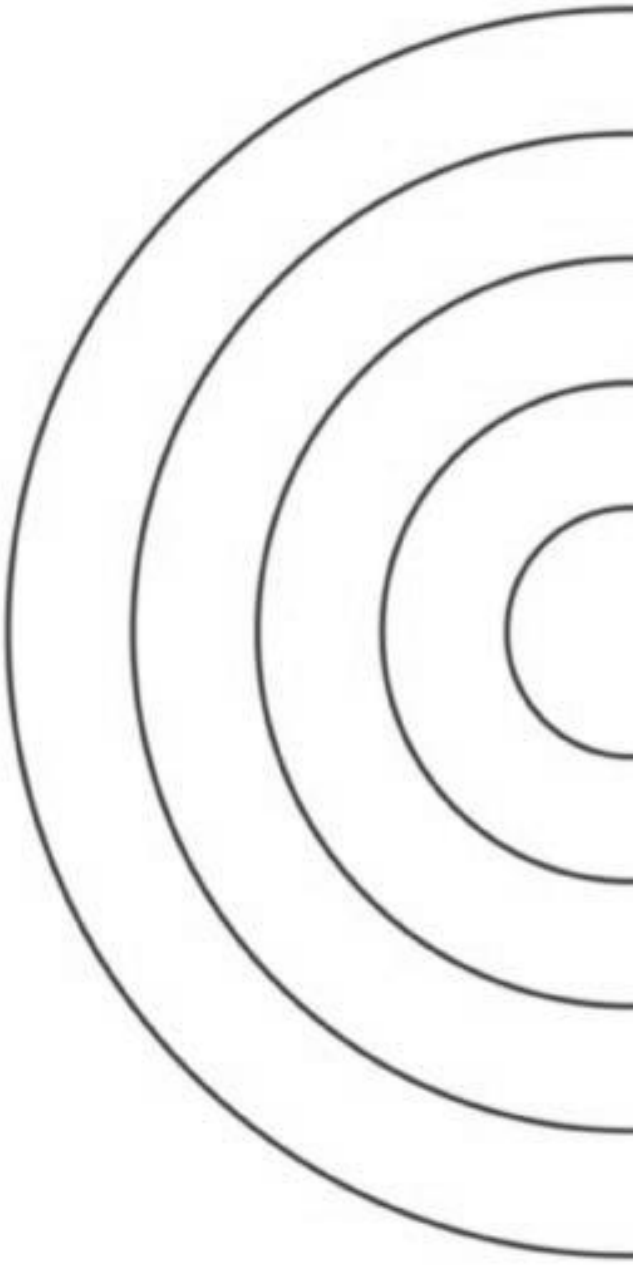
- The use of Unit Keys for security
- Optional Paging schemes
- 23 channel hopping sequence

4.2 DEPRECATED PROFILES

For the release of Bluetooth Specification 1.2, the deprecated profiles are:

- LAN Access Profile (LAP)
- Portions of WAP over LAP
- ESDP over L2CAP and LAP

Petitioner Kiosoft Exhibit 1011 Part 1
Page 156





Specification of the Bluetooth System

Wireless connections made easy

Core System Package
[Controller volume]

Version 1.2
05 November 2003

1.2

Bluetooth®

Revision History

The Revision History is shown in the “Appendix” on page 53[vol. 0].

Contributors

The persons who contributed to this specification are listed in the “Appendix” on page 53[vol. 0].

Web Site

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

Disclaimer and Copyright Notice

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

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 160

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

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

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

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

Copyright © 1999, 2000, 2001, 2002, 2003

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 162

TABLE OF CONTENTS

Part A
RADIO SPECIFICATION

Contents **27**

1 Scope **29**

2 Frequency Bands and Channel Arrangement **31**

3 Transmitter Characteristics **33**
3.1 Modulation Characteristics 34
3.2 Spurious Emissions 35
3.2.1 In-band spurious emission 35
3.3 Radio Frequency Tolerance 36

4 Receiver Characteristics **37**
4.1 Actual Sensitivity Level 37
4.2 Interference Performance 37
4.3 Out-of-Band Blocking 38
4.4 Intermodulation Characteristics 38
4.5 Maximum Usable Level 39
4.6 Receiver Signal Strength Indicator 39
4.7 Reference Signal Definition 39

5 Appendix A **41**

6 Appendix B **43**

Petitioner Kiosoft Exhibit 1011 Part 1
Page 163

Part B
BASEBAND SPECIFICATION

Contents

1 General Description 53
1.1 Bluetooth Clock 54
1.2 Bluetooth Device Addressing 55
1.2.1 Reserved addresses 55
1.3 Access Codes 56
2 Physical Channels 57
2.1 Physical Channel Definition 58
2.2 Basic Piconet Physical Channel 58
2.2.1 Master-slave definition 58
2.2.2 Hopping characteristics 59
2.2.3 Time slots 59
2.2.4 Piconet clocks 60
2.2.5 Transmit/receive timing 60
2.3 Adapted Piconet Physical Channel 63
2.3.1 Hopping characteristics 63
2.4 Page Scan Physical Channel 64
2.4.1 Clock estimate for paging 64
2.4.2 Hopping characteristics 64
2.4.3 Paging procedure timing 65
2.4.4 Page response timing 66
2.5 Inquiry Scan Physical Channel 68
2.5.1 Clock for inquiry 68
2.5.2 Hopping characteristics 68
2.5.3 Inquiry procedure timing 68
2.5.4 Inquiry response timing 68
2.6 Hop Selection 70
2.6.1 General selection scheme 70
2.6.2 Selection kernel 74
2.6.3 Adapted hop selection kernel 77
2.6.4 Control word 78
3 Physical Links 83
3.1 Link Supervision 83

4 Logical Transports 85
4.1 General 85
4.2 Logical Transport Address (LT_ADDR) 85
4.3 Synchronous Logical Transports 86
4.4 Asynchronous Logical Transport 86
4.5 Transmit/Receive Routines 87
4.5.1 TX Routine 87
4.5.2 RX routine 90
4.5.3 Flow control 91
4.6 Active Slave Broadcast Transport 92
4.7 Parked Slave Broadcast Transport 93
4.7.1 Parked member address (PM_ADDR) 93
4.7.2 Access request address (AR_ADDR) 93

5 Logical Links 95
5.1 Link Control Logical Link (LC) 95
5.2 ACL Control Logical Link (ACL-C) 95
5.3 User Asynchronous/Isochronous Logical Link (ACL-U) 95
5.3.1 Pausing the ACL-U logical link 96
5.4 User Synchronous Data Logical Link (SCO-S) 96
5.5 User Extended Synchronous Data Logical Link (eSCO-S) 96
5.6 Logical Link Priorities 96

6 Packets 97
6.1 General Format 97
6.2 Bit Ordering 97
6.3 Access Code 98
6.3.1 Access code types 98
6.3.2 Preamble 99
6.3.3 Sync word 99
6.3.4 Trailer 102
6.4 Packet Header 103
6.4.1 LT_ADDR 103
6.4.2 TYPE 103
6.4.3 FLOW 104
6.4.4 ARQN 104
6.4.5 SEQN 104
6.4.6 HEC 104

Petitioner Kiosoft Exhibit 1011 Part 1
Page 165

6.5 Packet Types 105
6.5.1 Common packet types 106
6.5.2 SCO packets 109
6.5.3 eSCO packets 110
6.5.4 ACL packets 111
6.6 Payload Format 112
6.6.1 Synchronous data field 112
6.6.2 Asynchronous data field 113
6.7 Packet Summary 116

7 Bitstream Processesing 117
7.1 Error Checking 118
7.1.1 HEC generation 118
7.1.2 CRC generation 119
7.2 Data Whitening 121
7.3 Error Correction 122
7.4 FEC Code: Rate 1/3 122
7.5 FEC Code: Rate 2/3 123
7.6 ARQ Scheme 124
7.6.1 Unnumbered ARQ 124
7.6.2 Retransmit filtering 127
7.6.3 Flushing payloads 130
7.6.4 Multi-slave considerations 130
7.6.5 Broadcast packets 130

8 Link Controller Operation 133
8.1 Overview of States 133
8.2 Standby State 134
8.3 Connection Establishment Substates 134
8.3.1 Page scan substate 134
8.3.2 Page substate 136
8.3.3 Page response substates 139
8.4 Device Discovery Substates 143
8.4.1 Inquiry scan substate 144
8.4.2 Inquiry substate 145
8.4.3 Inquiry response substate 146
8.5 Connection State 148

Petitioner Kiosoft Exhibit 1011 Part 1
Page 166

8.6 Active Mode 149
8.6.1 Polling in the active mode 150
8.6.2 SCO 150
8.6.3 eSCO 151
8.6.4 Broadcast scheme 154
8.6.5 Role switch 155
8.6.6 Scatternet 157
8.6.7 Hop sequence switching 158
8.6.8 Channel classificaition and channel map selection ...161
8.6.9 Power Management 162
8.7 Sniff Mode 163
8.7.1 Sniff Transition Mode 164
8.8 Hold Mode 165
8.9 Park State 165
8.9.1 Beacon train 166
8.9.2 Beacon access window 168
8.9.3 Parked slave synchronization 169
8.9.4 Parking 170
8.9.5 Master-initiated unparking 171
8.9.6 Slave-initiated unparking 171
8.9.7 Broadcast scan window 172
8.9.8 Polling in the park state 172

9 Audio 173
9.1 LOG PCM CODEC 173
9.2 CVSD CODEC 173
9.3 Error Handling 176
9.4 General Audio Requirements 176
9.4.1 Signal levels 176
9.4.2 CVSD audio quality 176

10 List of Figures 177

11 List of Tables 181

Appendix A:
General Audio Recommendations 182

Appendix B:
Timers 185

Appendix C: ....
Recommendations for AFH Operation in Park, Hold and Sniff 187

Petitioner Kiosoft Exhibit 1011 Part 1
Page 167

Part C
LINK MANAGER PROTOCOL

Contents **191**

1 Introduction **195**

2 General Rules **197**
2.1 Message Transport 197
2.2 Synchronization 197
2.3 Packet Format 198
2.4 Transactions 199
2.4.1 LMP Response Timeout 200
2.5 Error Handling 200
2.5.1 Transaction collision resolution 201
2.6 Procedure Rules 201
2.7 General Response Messages 202
2.8 Specification Writing Policies 202

3 Device Features **203**
3.1 General Description 203
3.2 Feature Definitions 203
3.3 Feature Mask Definition 207
3.4 Link Manager Interoperability policy 208

4 Procedure Rules **209**
4.1 Connection Control 209
4.1.1 Connection establishment 209
4.1.2 Detach 210
4.1.3 Power control 211
4.1.4 Adaptive frequency hopping 213
4.1.5 Channel classification 216
4.1.6 Link supervision 218
4.1.7 Channel quality driven data rate change (CQDDR) 219
4.1.8 Quality of service (QoS) 220
4.1.9 Paging scheme parameters 222
4.1.10 Control of multi-slot packets 223
4.2 Security 224
4.2.1 Authentication 224
4.2.2 Pairing 226
4.2.3 Change link key 229
4.2.4 Change current link key type 230
4.2.5 Encryption 232
4.2.6 Request supported encryption key size 236

Petitioner Kiosoft Exhibit 1011 Part 1
Page 168

4.3 Informational Requests 237
4.3.1 Timing accuracy 237
4.3.2 Clock offset....238
4.3.3 LMP version 238
4.3.4 Supported features....239
4.3.5 Name request....241
4.4 Role Switch 242
4.4.1 Slot offset 242
4.4.2 Role switch 243
4.5 Modes of Operation 245
4.5.1 Hold mode 245
4.5.2 Park state 247
4.5.3 Sniff mode 253
4.6 Logical Transports....256
4.6.1 SCO logical transport 256
4.6.2 eSCO logical transport 259
4.7 Test Mode 264
4.7.1 Activation and deactivation of test mode....264
4.7.2 Control of test mode 265
4.7.3 Summary of test mode PDUs....266
5 Summary 269
5.1 PDU Summary 269
5.2 Parameter Definitions 277
5.3 Default Values 284
6 List of Figures....285
7 List of Tables 289

Petitioner Kiosoft Exhibit 1011 Part 1
Page 169

Part D
ERROR CODES

Contents **293**

1 Overview of Error Codes **295**

1.1 Usage Descriptions 295
1.2 HCI Command Errors 295
1.3 List of Error Codes 296

2 Error Code Descriptions **299**

2.1 Unknown HCI Command (0X01) 299
2.2 Unknown Connection Identifier (0X02) 299
2.3 Hardware Failure (0X03) 299
2.4 Page Timeout (0X04) 299
2.5 Authentication Failure (0X05) 299
2.6 PIN Missing (0X06) 299
2.7 Memory Capacity Exceeded (0X07) 299
2.8 Connection Timeout (0X08) 300
2.9 Connection Limit Exceeded (0X09) 300
2.10 Synchronous Connection Limit to a Device Exceeded (0X0A) 300
2.11 ACL Connection Already Exists (0X0B) 300
2.12 Command Disallowed (0X0C) 300
2.13 Connection Rejected due to Limited Resources (0X0D) 300
2.14 Connection Rejected due to Security Reasons (0X0E) 300
2.15 Connection Rejected due to Unacceptable BD_ADDR (0X0F) 301
2.16 Connection Accept Timeout Exceeded (0X10) 301
2.17 Unsupported Feature or Parameter Value (0X11) 301
2.18 Invalid HCI Command Parameters (0X12) 301
2.19 Remote User Terminated Connection (0X13) 301
2.20 Remote Device Terminated Connection due to Low Resources (0X14) 302
2.21 Remote Device Terminated Connection due to Power Off (0X15) 302
2.22 Connection Terminated by Local Host (0X16) 302
2.23 Repeated Attempts (0X17) 302
2.24 Pairing not Allowed (0X18) 302
2.25 Unknown LMP PDU (0X19) 302
2.26 Unsupported Remote Feature (0X1A) 302
2.27 SCO Offset Rejected (0X1B) 302
2.28 SCO Interval Rejected (0X1C) 303
2.29 SCO Air Mode Rejected (0X1D) 303
2.30 INvalid LMP Parameters (0X1E) 303
2.31 Unspecified Error (0X1F) 303
2.32 Unsupported LMP Parameter Value (0X20) 303

Petitioner Kiosoft Exhibit 1011 Part 1
Page 170

2.33 Role Change Not Allowed (0X21) 303
2.34 LMP Response Timeout (0X22) 303
2.35 LMP Error Transaction Collision (0X23) 304
2.36 LMP PDU Not Allowed (0X24) 304
2.37 Encryption Mode Not Acceptable (0X25) 304
2.38 Link Key Can Not be Changed (0X26) 304
2.39 Requested Qos is Not Supported (0X27) 304
2.40 Instant Passed (0X28) 304
2.41 Pairing with Unit Key Not Supported (0X29) 304
2.42 DIFFERENT TRANSACTION COLLISION (0X2A) 304
2.43 QoS Unacceptable Parameter (0X2C) 304
2.44 QoS Rejected (0X2D) 305
2.45 Channel Classification Not Supported (0X2E) 305
2.46 Insufficient Security (0X2F) 305
2.47 Parameter out of Mandatory Range (0X30) 305
2.48 Role Switch Pending (0X32) 305
2.49 Reserved Slot Violation (0X34) 305
2.50 Role Switch Failed (0X35) 305

Part E
HOST CONTROLLER INTERFACE FUNCTIONAL SPECIFICATION

Contents **309**

1 Introduction **317**
1.1 Lower Layers of the Bluetooth Software Stack 317

2 Overview of Host Controller Transport Layer **319**

3 Overview of Commands and Events **321**
3.1 Generic Events 322
3.2 Device Setup 322
3.3 Controller Flow Control 323
3.4 Controller Information 323
3.5 Controller Configuration 324
3.6 Device Discovery 325
3.7 Connection Setup 327
3.8 Remote Information 329
3.9 Synchronous Connections 330
3.10 Connection State 331
3.11 Piconet Structure 332
3.12 Quality of Service 333
3.13 Physical Links 334
3.14 Host Flow Control 335
3.15 Link Information 336
3.16 Authentication and Encryption 337
3.17 Testing 339
3.18 Alphabetical List of Commands and Events 340

4 HCI Flow Control **345**
4.1 Host to Controller Data Flow Control 345
4.2 Controller to Host Data Flow Control 346
4.3 Disconnection Behaviour 347
4.4 Command Flow Control 347
4.5 Command Error Handling 348

Petitioner Kiosoft Exhibit 1011 Part 1
Page 172

5 HCI Data Formats 349
5.1 Introduction 349
5.2 Data and Parameter Formats.....349
5.3 Connection Handles.....350
5.4 Exchange of HCI-Specific Information 351
5.4.1 HCI Command Packet.....351
5.4.2 HCI ACL Data Packets.....353
5.4.3 HCI Synchronous Data Packets.....355
5.4.4 HCI Event Packet.....356
6 HCI Configuration Parameters.....357
6.1 Scan Enable.....357
6.2 Inquiry Scan Interval 357
6.3 Inquiry Scan Window 358
6.4 Inquiry Scan Type 358
6.5 Inquiry Mode 358
6.6 Page Timeout.....359
6.7 Connection Accept Timeout.....359
6.8 Page Scan Interval.....360
6.9 Page Scan Window.....360
6.10 Page Scan Period Mode 360
6.11 Page Scan Type.....361
6.12 Voice Setting 361
6.13 PIN Type 362
6.14 Link Key 362
6.15 Authentication Enable 362
6.16 Encryption Mode 363
6.17 Failed Contact Counter 364
6.18 Hold Mode Activity 364
6.19 Link Policy Settings.....365
6.20 Flush Timeout 366
6.21 Num Broadcast Retransmissions.....366
6.22 Link Supervision Timeout.....367
6.23 Synchronous Flow Control Enable.....367
6.24 Local Name.....368
6.25 Class Of Device 368
6.26 Supported Commands 369

7 HCI Commands and Events 373
7.1 Link Control Commands 373
7.1.1 Inquiry Command 373
7.1.2 Inquiry Cancel Command 375
7.1.3 Periodic Inquiry Mode Command 376
7.1.4 Exit Periodic Inquiry Mode Command 379
7.1.5 Create Connection Command 380
7.1.6 Disconnect Command 383
7.1.7 Create Connection Cancel Command 384
7.1.8 Accept Connection Request Command 386
7.1.9 Reject Connection Request Command 388
7.1.10 Link Key Request Reply Command 389
7.1.11 Link Key Request Negative Reply Command 391
7.1.12 PIN Code Request Reply Command 392
7.1.13 PIN Code Request Negative Reply Command 394
7.1.14 Change Connection Packet Type Command 395
7.1.15 Authentication Requested Command 398
7.1.16 Set Connection Encryption Command 399
7.1.17 Change Connection Link Key Command 400
7.1.18 Master Link Key Command 401
7.1.19 Remote Name Request Command 402
7.1.20 Remote Name Request Cancel Command 404
7.1.21 Read Remote Supported Features Command 405
7.1.22 Read Remote Extended Features Command 406
7.1.23 Read Remote Version Information Command 407
7.1.24 Read Clock Offset Command 408
7.1.25 Read LMP Handle Command 409
7.1.26 Setup Synchronous Connection Command 411
7.1.27 Accept Synchronous Connection Request Command 415
7.1.28 Reject Synchronous Connection Request Command . 419

Petitioner Kiosoft Exhibit 1011 Part 1
Page 174

7.2 Link Policy Commands....420
7.2.1 Hold Mode Command....420
7.2.2 Sniff Mode Command....422
7.2.3 Exit Sniff Mode Command....425
7.2.4 Park State Command....426
7.2.5 Exit Park State Command....428
7.2.6 QoS Setup Command....429
7.2.7 Role Discovery Command....431
7.2.8 Switch Role Command....432
7.2.9 Read Link Policy Settings Command....433
7.2.10 Write Link Policy Settings Command....434
7.2.11 Read Default Link Policy Settings Command....436
7.2.12 Write Default Link Policy Settings Command....437
7.2.13 Flow Specification Command....438
7.3 Controller & Baseband Commands....440
7.3.1 Set Event Mask Command....440
7.3.2 Reset Command....442
7.3.3 Set Event Filter Command....443
7.3.4 Flush Command....448
7.3.5 Read PIN Type Command....450
7.3.6 Write PIN Type Command....451
7.3.7 Create New Unit Key Command....452
7.3.8 Read Stored Link Key Command....453
7.3.9 Write Stored Link Key Command....454
7.3.10 Delete Stored Link Key Command....456
7.3.11 Write Local Name Command....457
7.3.12 Read Local Name Command....458
7.3.13 Read Connection Accept Timeout Command....459
7.3.14 Write Connection Accept Timeout Command....460
7.3.15 Read Page Timeout Command....461
7.3.16 Write Page Timeout Command....462
7.3.17 Read Scan Enable Command....463
7.3.18 Write Scan Enable Command....464
7.3.19 Read Page Scan Activity Command....465
7.3.20 Write Page Scan Activity Command....467
7.3.21 Read Inquiry Scan Activity Command....468
7.3.22 Write Inquiry Scan Activity Command....469
7.3.23 Read Authentication Enable Command....470

Petitioner Kiosoft Exhibit 1011 Part 1
Page 175

7.3.24 Write Authentication Enable Command 471
7.3.25 Read Encryption Mode Command 472
7.3.26 Write Encryption Mode Command 473
7.3.27 Read Class of Device Command 474
7.3.28 Write Class of Device Command 475
7.3.29 Read Voice Setting Command 476
7.3.30 Write Voice Setting Command 477
7.3.31 Read Automatic Flush Timeout Command 478
7.3.32 Write Automatic Flush Timeout Command 479
7.3.33 Read Num Broadcast Retransmissions Command 480
7.3.34 Write Num Broadcast Retransmissions Command 481
7.3.35 Read Hold Mode Activity Command 482
7.3.36 Write Hold Mode Activity Command 483
7.3.37 Read Transmit Power Level Command 484
7.3.38 Read Synchronous Flow Control Enable Command ... 486
7.3.39 Write Synchronous Flow Control Enable Command ... 487
7.3.40 Set Controller To Host Flow Control Command 488
7.3.41 Host Buffer Size Command 489
7.3.42 Host Number Of Completed Packets Command 491
7.3.43 Read Link Supervision Timeout Command 493
7.3.44 Write Link Supervision Timeout Command 494
7.3.45 Read Number Of Supported IAC Command 496
7.3.46 Read Current IAC LAP Command 497
7.3.47 Write Current IAC LAP Command 498
7.3.48 Read Page Scan Period Mode Command 500
7.3.49 Write Page Scan Period Mode Command 501
7.3.50 Set AFH Host Channel Classification Command 502
7.3.51 Read Inquiry Scan Type Command 503
7.3.52 Write Inquiry Scan Type Command 504
7.3.53 Read Inquiry Mode Command 505
7.3.54 Write Inquiry Mode Command 506
7.3.55 Read Page Scan Type Command 507
7.3.56 Write Page Scan Type Command 508
7.3.57 Read AFH Channel Assessment Mode Command 509
7.3.58 Write AFH Channel Assessment Mode Command 510

Petitioner Kiosoft Exhibit 1011 Part 1
Page 176

7.4 Informational Parameters....512
7.4.1 Read Local Version Information Command....512
7.4.2 Read Local Supported Commands Command....514
7.4.3 Read Local Supported Features Command....515
7.4.4 Read Local Extended Features Command 516
7.4.5 Read Buffer Size Command....518
7.4.6 Read BD_ADDR Command....520
7.5 Status Parameters....521
7.5.1 Read Failed Contact Counter Command....521
7.5.2 Reset Failed Contact Counter Command....523
7.5.3 Read Link Quality Command....524
7.5.4 Read RSSI Command....525
7.5.5 Read AFH Channel Map Command....527
7.5.6 Read Clock Command....529
7.6 Testing Commands....531
7.6.1 Read Loopback Mode Command....531
7.6.2 Write Loopback Mode Command....532
7.6.3 Enable Device Under Test Mode Command....535
7.7 Events....537
7.7.1 Inquiry Complete Event....537
7.7.2 Inquiry Result Event....538
7.7.3 Connection Complete Event....540
7.7.4 Connection Request Event....541
7.7.5 Disconnection Complete Event....543
7.7.6 Authentication Complete Event....544
7.7.7 Remote Name Request Complete Event....545
7.7.8 Encryption Change Event....546
7.7.9 Change Connection Link Key Complete Event....547
7.7.10 Master Link Key Complete Event....548
7.7.11 Read Remote Supported Features Complete Event...549
7.7.12 Read Remote Version Information Complete Event....550
7.7.13 QoS Setup Complete Event....551
7.7.14 Command Complete Event....553
7.7.15 Command Status Event....554
7.7.16 Hardware Error Event....555
7.7.17 Flush Occurred Event....555
7.7.18 Role Change Event....556
7.7.19 Number Of Completed Packets Event....557

Petitioner Kiosoft Exhibit 1011 Part 1
Page 177

7.7.20 Mode Change Event 558
7.7.21 Return Link Keys Event..... 560
7.7.22 PIN Code Request Event..... 561
7.7.23 Link Key Request Event..... 562
7.7.24 Link Key Notification Event 563
7.7.25 Loopback Command Event..... 564
7.7.26 Data Buffer Overflow Event..... 564
7.7.27 Max Slots Change Event..... 565
7.7.28 Read Clock Offset Complete Event 566
7.7.29 Connection Packet Type Changed Event 567
7.7.30 QoS Violation Event..... 569
7.7.31 Page Scan Repetition Mode Change Event..... 570
7.7.32 Flow Specification Complete Event..... 571
7.7.33 Inquiry Result with RSSI Event 573
7.7.34 Read Remote Extended Features Complete Event 575
7.7.35 Synchronous Connection Complete Event 576
7.7.36 Synchronous Connection Changed event..... 578

8 List of Figures 581

9 List of Tables 583

Appendix A:
Deprecated Commands, Events and Configuration Parameters 585

Part F
MESSAGE SEQUENCE CHARTS

Contents **595**

1 Introduction **599**
1.1 Notation 599
1.2 Flow of Control 600
1.3 Example MSC 600

2 Services Without Connection Request **601**
2.1 Remote Name Request 601
2.2 One-time Inquiry 602
2.3 Periodic Inquiry 604

3 ACL Connection Establishment and Detachment **607**
3.1 Connection Setup 608

4 Optional Activities After ACL Connection Establishment **615**
4.1 Authentication Requested 615
4.2 Set Connection Encryption 616
4.3 Change Connection Link Key 617
4.4 Master Link Key 618
4.5 Read Remote Supported Features 620
4.6 Read Remote Extended Features 620
4.7 Read Clock Offset 621
4.8 Read Remote Version Information 621
4.9 QOS Setup 622
4.10 Switch Role 622

5 Synchronous Connection Establishment and Detachment **625**
5.1 Synchronous Connection Setup 625

6 Sniff, Hold and Park **631**
6.1 Sniff Mode 631
6.2 Hold Mode 632
6.3 Park State 634

7 Buffer Management, Flow Control **637**

8 Loopback Mode **639**
8.1 Local Loopback Mode 639
8.2 Remote Loopback Mode 641

9 List of Figures **643**

Petitioner Kiosoft Exhibit 1011 Part 1
Page 179

Part G
SAMPLE DATA

Contents **647**

1 Encryption Sample Data **649**
- 1.1 Generating Kc' from Kc, ... 649
- 1.2 First Set of Sample Data 652
- 1.3 Second Set of Sample Data 660
- 1.4 Third Set of Samples 668
- 1.5 Fourth Set of Samples 676

2 Frequency Hopping Sample Data **685**
- 2.1 First set 686
- 2.2 Second set 692
- 2.3 Third set 698

3 Access Code Sample Data **705**

4 HEC and Packet Header Sample Data **709**

5 CRC Sample Data **711**

6 Complete Sample Packets **713**
- 6.1 Example of DH1 Packet 713
- 6.2 Example of DM1 Packet 714

7 Whitening Sequence Sample Data **715**

8 FEC Sample Data **719**

9 Encryption Key Sample Data **721**
- 9.1 Four Tests of E1 721
- 9.2 Four Tests of E21 726
- 9.3 Three Tests of E22 728
- 9.4 Tests of E22 With Pin Augmenting 730
- 9.5 Four Tests of E3 740

Petitioner Kiosoft Exhibit 1011 Part 1
Page 180

Part H
SECURITY SPECIFICATION

Contents **747**

1 Security Overview **749**

2 Random Number Generation **751**

3 Key Management **753**

3.1 Key Types 753

3.2 Key Generation and Initialization 755

3.2.1 Generation of initialization key, ... 756

3.2.2 Authentication 756

3.2.3 Generation of a unit key 756

3.2.4 Generation of a combination key 757

3.2.5 Generating the encryption key 758

3.2.6 Point-to-multipoint configuration 759

3.2.7 Modifying the link keys 760

3.2.8 Generating a master key 760

4 Encryption **763**

4.1 Encryption Key Size Negotiation 764

4.2 Encryption of Broadcast Messages 764

4.3 Encryption Concept 765

4.4 Encryption Algorithm 766

4.4.1 The operation of the cipher 768

4.5 LFSR Initialization 769

4.6 Key Stream Sequence 772

5 Authentication **773**

5.1 Repeated Attempts 775

6 The Authentication And Key-Generating Functions **777**

6.1 The Authentication Function E1 777

6.2 The Functions Ar and A'r 779

6.2.1 The round computations 779

6.2.2 The substitution boxes "e" and "l" 779

6.2.3 Key scheduling 780

6.3 E2-Key Generation Function for Authentication 781

6.4 E3-Key Generation Function for Encryption 783

7 List of Figures **785**

8 List of Tables **787**

Petitioner Kiosoft Exhibit 1011 Part 1
Page 182

Core System Package [Controller volume]
Part A



RADIO SPECIFICATION

Petitioner Kiosoft Exhibit 1011 Part 1
Page 184

CONTENTS

1 Scope 29

2 Frequency Bands and Channel Arrangement 31

3 Transmitter Characteristics 33

3.1 Modulation Characteristics 34

3.2 Spurious Emissions 35

3.2.1 In-band spurious emission 35

3.3 Radio Frequency Tolerance 36

4 Receiver Characteristics 37

4.1 Actual Sensitivity Level 37

4.2 Interference Performance 37

4.3 Out-of-Band Blocking 38

4.4 Intermodulation Characteristics 38

4.5 Maximum Usable Level 39

4.6 Receiver Signal Strength Indicator 39

4.7 Reference Signal Definition 39

5 Appendix A 41

6 Appendix B 43

Petitioner Kiosoft Exhibit 1011 Part 1
Page 186

1 SCOPE

Bluetooth devices operate in the unlicensed 2.4 GHz ISM (Industrial Scientific Medical) band. A frequency hop transceiver is applied to combat interference and fading. A shaped, binary FM modulation is applied to minimize transceiver complexity. The symbol rate is 1 Ms/s. For full duplex transmission, a Time-Division Duplex (TDD) scheme is used. This specification defines the requirements for a Bluetooth radio.

Requirements are defined for two reasons:

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

The Bluetooth radio shall fulfil the stated requirements under the operating conditions specified in Appendix A and Appendix B. The radio parameters shall be measured according to the methods described in the RF Test Specification.

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

Europe:

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

Japan:

Approval Standards: Association of Radio Industries and Businesses, ARIB
Documents: ARIB STD-T66
Approval Authority: Ministry of Post and Telecommunications, MPT.

North America:

Approval Standards: Federal Communications Commission, FCC, USA
Documents: CFR47, Part 15, Sections 15.205, 15.209, 15.247 and 15.249

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 187

Petitioner Kiosoft Exhibit 1011 Part 1
Page 188

2 FREQUENCY BANDS AND CHANNEL ARRANGEMENT

The Bluetooth system operates in the 2.4 GHz ISM band. This frequency band is 2400 - 2483.5 MHz.

Regulatory Range	RF Channels
2.400-2.4835 GHz	f=2402+k MHz, k=0,…,78

Table 2.1: Operating frequency bands

RF channels are spaced 1 MHz and are ordered in channel number k as shown in Table 2.1. In order to comply with out-of-band regulations in each country, a guard band is used at the lower and upper band edge.

Lower Guard Band	Upper Guard Band
2 MHz	3.5 MHz

Table 2.2: Guard Bands

Petitioner Kiosoft Exhibit 1011 Part 1
Page 190

3 TRANSMITTER CHARACTERISTICS

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

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

If transmitting antennas of directional gain greater than 0 dBi are used, the applicable paragraphs in EN 300 328, EN 301 489-17and FCC part 15 shall be compensated for.

The device is classified into three power classes.

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

Table 3.1: Power classes

1. Minimum output power at maximum power setting.
2. The lower power limit Pmin<-30dBm is suggested but is not mandatory, and may be chosen according to application needs.

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

Devices with power control capability optimizes the output power in a physical link with LMP commands (see Link Manager Protocol). It is done by measuring RSSI and reporting back if the transmission power shall be increased or decreased if possible.

In a connection, the output power shall not exceed the maximum output power of power class 2 for transmitting packets if the receiving device does not support the necessary messaging for sending the power control messages, see

Link Manager Protocol Section 4.1.3 on page 211. In this case, the transmitting device shall comply with the rules of a class 2 or class 3 device.

If a class 1 device is paging or inquiring very close to another device, the input power can be larger than the requirement in Section 4.5 on page 39. This can cause the receiving device to fail to respond. It may therefore be useful to page at Class 2 or 3 power in addition to paging at power class 1.

3.1 MODULATION CHARACTERISTICS

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

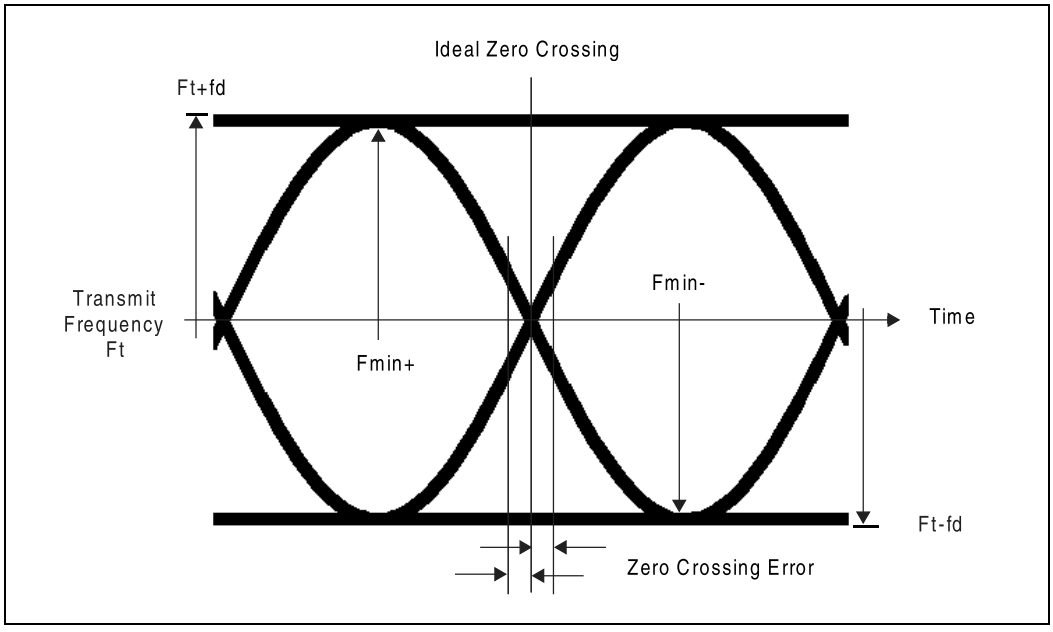


Figure 3.1: GFSK parameters definition.

For each transmission, the minimum frequency deviation, Fmin = min{|Fmin+|, Fmin-}, which corresponds to 1010 sequence shall be no smaller than ±80% of the frequency deviation (fd) with respect to the transmit frequency Ft, which corresponds to a 00001111 sequence.

In addition, the minimum frequency deviation shall never be smaller than 115 kHz. The data transmitted has a symbol rate of 1 Ms/s.

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

3.2 SPURIOUS EMISSIONS

In-band spurious emissions shall be measured with a frequency hopping radio transmitting on one RF channel and receiving on a second RF channel; this means that the synthesizer may change RF channels between reception and transmission, but always returns to the same transmit RF channel. There will be no reference in this document to out of ISM band spurious emissions; the equipment manufacturer is responsible for compliance in the intended country of use.

3.2.1 In-band spurious emission

Within the ISM band the transmitter shall pass a spectrum mask, given in Table 3.2. The spectrum shall comply with the 20dB bandwidth definition in FCC part 15.247 and shall be measured accordingly. In addition to the FCC requirement an adjacent channel power on adjacent channels with a difference in RF channel number of two or greater is defined. This adjacent channel power is defined as the sum of the measured power in a 1 MHz RF channel. The transmitted power shall be measured in a 100 kHz bandwidth using maximum hold. The device shall transmit on RF channel M and the adjacent channel power shall be measured on RF channel number N. The transmitter shall transmit a pseudo random data pattern in the payload throughout the test.

Frequency offset	Transmit Power
± 500 kHz	-20 dBc
2MHz (|M-N| = 2)	-20 dBm
3MHz or greater (|M-N| ≥ 3)	-40 dBm

Table 3.2: Transmit Spectrum mask.

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 193

3.3 RADIO FREQUENCY TOLERANCE

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

The limits on the transmitter center frequency drift within a packet are specified in Table 3.3. The different packets are defined in the Baseband Specification.

Duration of Packet	Frequency Drift
Max length one slot packet	±25 kHz
Max length three slot packet	±40 kHz
Max length five slot packet	±40 kHz
Maximum drift rate[1]	400 Hz/µs

Table 3.3: Maximum allowable frequency drifts in a packet.

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

4 RECEIVER CHARACTERISTICS

The receiver characteristics shall be measured using loopback as defined in “Test Methodology” on page 233[vol. 3].

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

4.1 ACTUAL SENSITIVITY LEVEL

The actual sensitivity level is defined as the input level for which a raw bit error rate (BER) of 0.1% is met. The receiver sensitivity shall be below or equal to –70 dBm with any Bluetooth transmitter compliant to the transmitter specification specified in Section 3 on page 33.

4.2 INTERFERENCE PERFORMANCE

The interference performance on Co-channel and adjacent 1 MHz and 2 MHz shall be measured with the wanted signal 10 dB over the reference sensitivity level. For interference performance on all other RF channels the wanted signal shall be 3 dB over the reference sensitivity level. If the frequency of an interfering signal is outside of the band 2400-2483,5 MHz, the out-of-band blocking specification (see Section 4.3 on page 38) shall apply. The interfering signal shall be Bluetooth-modulated (see section 4.7 on page 39). The BER shall be ≤0.1% for all the signal-to-interference ratios listed in Table 4.1:

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

Table 4.1: Interference performance

1. In-band image frequency
2. If the image frequency ≠ n*1 MHz, then the image reference frequency is defined as the closest n*1 MHz frequency.

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

These specifications are only to be tested at nominal temperature conditions with a device receiving on one RF channel and transmitting on a second RF channel; this means that the synthesizer may change RF channels between reception and transmission, but always returns to the same receive RF channel.

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

4.3 OUT-OF-BAND BLOCKING

The out-of-band suppression (or rejection) shall be measured with the wanted signal 3 dB over the reference sensitivity level. The interfering signal shall be a continuous wave signal. The BER shall be ≤ 0.1%. The out-of-band blocking shall fulfil the following requirements:

Interfering Signal Frequency	Interfering Signal Power Level
30 MHz - 2000 MHz	-10 dBm
2000 - 2399 MHz	-27 dBm
2484 – 3000 MHz	-27 dBm
3000 MHz – 12.75 GHz	-10 dBm

Table 4.2: Out-of-band suppression (or rejection) requirements.

24 exceptions are permitted which are dependent upon the given RF channel and are centered at a frequency which is an integer multiple of 1 MHz. For at least 19 of these spurious response frequencies, a reduced interference level of at least -50dBm is allowed in order to achieve the required BER=0.1% performance, whereas for a maximum of 5 of the spurious frequencies the interference level may be assumed arbitrarily lower.

4.4 INTERMODULATION CHARACTERISTICS

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

- The wanted signal shall be at frequency f_0 with a power level 6 dB over the reference sensitivity level.
- A static sine wave signal shall be at a frequency f_1 with a power level of –39 dBm.
- A Bluetooth modulated signal (see Section 4.7 on page 39) shall be at f_2 with a power level of -39 dBm.

Frequencies f0, f1 and f2 shall be chosen such that $f_0=2f_1-f_2$ and $|f_2-f_1|=n*1$ MHz, where n can be 3, 4, or 5. The system shall fulfill at least one of the three alternatives (n=3,4, or 5).

Petitioner Kiosoft Exhibit 1011 Part 1
Page 196

4.5 MAXIMUM USABLE LEVEL

The maximum usable input level that the receiver operates at shall be greater than -20 dBm. The BER shall be less than or equal to 0.1% at –20 dBm input power.

4.6 RECEIVER SIGNAL STRENGTH INDICATOR

If a device supports Receive Signal Strength Indicator (RSSI) the accuracy shall be +/- 6 dBm.

4.7 REFERENCE SIGNAL DEFINITION

A Bluetooth modulated interfering signal shall be defined as:

Modulation = GFSK

Modulation index = 0.32±1%

BT= 0.5±1%

Bit Rate = 1 Mbps ±1 ppm

Modulating Data for wanted signal = PRBS9

Modulating Data for interfering signal = PRBS 15

Frequency accuracy better than ±1 ppm.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 198

5 APPENDIX A

5.1 NOMINAL TEST CONDITIONS

5.1.1 Nominal temperature

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

5.1.2 Nominal power source

5.1.2.1 Mains voltage

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

5.1.2.2 Lead-acid battery power sources used in vehicles

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

5.1.2.3 Other power sources

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

5.2 EXTREME TEST CONDITIONS

5.2.1 Extreme temperatures

The extreme temperature range shall be the largest temperature range given by the combination of:

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

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

5.2.2 Extreme power source voltages

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

5.2.2.1 Mains voltage

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

5.2.2.2 Lead-acid battery power source used on vehicles

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

5.2.2.3 Power sources using other types of batteries

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

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

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

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

5.2.2.4 Other power sources

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 200

6 APPENDIX B

The radio parameters shall be tested in the following conditions

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

ETC = Extreme Test Conditions
NTC = Nominal Test Conditions

Petitioner Kiosoft Exhibit 1011 Part 1
Page 202

Core System Package [Controller volume]
Part B



BASEBAND SPECIFICATION

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

CONTENTS

1 General Description 53
1.1 Bluetooth Clock 54
1.2 Bluetooth Device Addressing 55
1.2.1 Reserved addresses 55
1.3 Access Codes 56
2 Physical Channels 57
2.1 Physical Channel Definition 58
2.2 Basic Piconet Physical Channel 58
2.2.1 Master-slave definition 58
2.2.2 Hopping characteristics 59
2.2.3 Time slots 59
2.2.4 Piconet clocks 60
2.2.5 Transmit/receive timing 60
2.2.5.1 Piconet physical channel timing 61
2.2.5.2 Piconet physical channel re-synchronization 62
2.3 Adapted Piconet Physical Channel 63
2.3.1 Hopping characteristics 63
2.4 Page Scan Physical Channel 64
2.4.1 Clock estimate for paging 64
2.4.2 Hopping characteristics 64
2.4.3 Paging procedure timing 65
2.4.4 Page response timing 66
2.5 Inquiry Scan Physical Channel 68
2.5.1 Clock for inquiry 68
2.5.2 Hopping characteristics 68
2.5.3 Inquiry procedure timing 68
2.5.4 Inquiry response timing 68
2.6 Hop Selection 70
2.6.1 General selection scheme 70
2.6.2 Selection kernel 74
2.6.2.1 First addition operation 74
2.6.2.2 XOR operation 75
2.6.2.3 Permutation operation 75
2.6.2.4 Second addition operation 76
2.6.2.5 Register bank 76
2.6.3 Adapted hop selection kernel 77
2.6.3.1 Channel re-mapping function 77

2.6.4 Control word 78
2.6.4.1 Page scan and inquiry scan hopping sequences 79
2.6.4.2 Page hopping sequence 80
2.6.4.3 Slave page response hopping sequence 80
2.6.4.4 Master page response hopping sequence 81
2.6.4.5 Inquiry hopping sequence 81
2.6.4.6 Inquiry response hopping sequence 82
2.6.4.7 Basic and adapted channel hopping sequence 82

3 Physical Links 83
3.1 Link Supervision 83

4 Logical Transports 85
4.1 General 85
4.2 Logical Transport Address (LT_ADDR) 85
4.3 Synchronous Logical Transports 86
4.4 Asynchronous Logical Transport 86
4.5 Transmit/Receive Routines 87
4.5.1 TX Routine 87
4.5.1.1 ACL traffic 88
4.5.1.2 SCO traffic 89
4.5.1.3 Mixed data/voice traffic 89
4.5.1.4 eSCO Traffic 90
4.5.1.5 Default packet types 90
4.5.2 RX routine 90
4.5.3 Flow control 91
4.5.3.1 Destination control 92
4.5.3.2 Source control 92
4.6 Active Slave Broadcast Transport 92
4.7 Parked Slave Broadcast Transport 93
4.7.1 Parked member address (PM_ADDR) 93
4.7.2 Access request address (AR_ADDR) 93

5 Logical Links 95
5.1 Link Control Logical Link (LC) 95
5.2 ACL Control Logical Link (ACL-C) 95
5.3 User Asynchronous/Isochronous Logical Link (ACL-U) 95
5.3.1 Pausing the ACL-U logical link 96
5.4 User Synchronous Data Logical Link (SCO-S) 96
5.5 User Extended Synchronous Data Logical Link (eSCO-S) 96
5.6 Logical Link Priorities 96

6 Packets 97
6.1 General Format 97
6.2 Bit Ordering 97
6.3 Access Code 98
6.3.1 Access code types 98
6.3.2 Preamble 99
6.3.3 Sync word 99
6.3.3.1 Synchronization word definition 99
6.3.3.2 Pseudo-random noise sequence generation 102
6.3.4 Trailer 102
6.4 Packet Header 103
6.4.1 LT_ADDR 103
6.4.2 TYPE 103
6.4.3 FLOW 104
6.4.4 ARQN 104
6.4.5 SEQN 104
6.4.6 HEC 104
6.5 Packet Types 105
6.5.1 Common packet types 106
6.5.1.1 ID packet 106
6.5.1.2 NULL packet 106
6.5.1.3 POLL packet 106
6.5.1.4 FHS packet 106
6.5.1.5 DM1 packet 108
6.5.2 SCO packets 109
6.5.2.1 HV1 packet 109
6.5.2.2 HV2 packet 109
6.5.2.3 HV3 packet 109
6.5.2.4 DV packet 109
6.5.3 eSCO packets 110
6.5.3.1 EV3 packet 110
6.5.3.2 EV4 packet 110
6.5.3.3 EV5 packet 110
6.5.4 ACL packets 111
6.5.4.1 DM1 packet 111
6.5.4.2 DH1 packet 111
6.5.4.3 DM3 packet 111
6.5.4.4 DH3 packet 111
6.5.4.5 DM5 packet 111
6.5.4.6 DH5 packet 112
6.5.4.7 AUX1 packet 112

6.6 Payload Format 112
6.6.1 Synchronous data field 112
6.6.2 Asynchronous data field 113
6.7 Packet Summary 116

7 Bitstream Processing 117
7.1 Error Checking 118
7.1.1 HEC generation 118
7.1.2 CRC generation 119
7.2 Data Whitening 121
7.3 Error Correction 122
7.4 FEC Code: Rate 1/3 122
7.5 FEC Code: Rate 2/3 123
7.6 ARQ Scheme 124
7.6.1 Unnumbered ARQ 124
7.6.2 Retransmit filtering 127
7.6.2.1 Initialization of SEQN at start of new connection 128
7.6.2.2 ACL and SCO retransmit filtering 128
7.6.2.3 eSCO retransmit filtering 129
7.6.2.4 FHS retransmit filtering 129
7.6.2.5 Packets without CRC retransmit filtering 129
7.6.3 Flushing payloads 130
7.6.4 Multi-slave considerations 130
7.6.5 Broadcast packets 130

8 Link Controller Operation 133
8.1 Overview of States 133
8.2 Standby State 134
8.3 Connection Establishment Substates 134
8.3.1 Page scan substate 134
8.3.2 Page substate 136
8.3.3 Page response substates 139
8.3.3.1 Slave response substate 140
8.3.3.2 Master response substate 142
8.4 Device Discovery Substates 143
8.4.1 Inquiry scan substate 144
8.4.2 Inquiry substate 145
8.4.3 Inquiry response substate 147
8.5 Connection State 148

8.6 Active Mode 149
8.6.1 Polling in the active mode 150
8.6.2 SCO 150
8.6.3 eSCO 151
8.6.4 Broadcast scheme 154
8.6.5 Role switch 155
8.6.6 Scatternet 157
8.6.6.1 Inter-piconet communications 157
8.6.7 Hop sequence switching 158
8.6.8 Channel classification and channel map selection 161
8.6.9 Power Management 162
8.6.9.1 Packet handling 162
8.6.9.2 Slot occupancy 162
8.6.9.3 Recommendations for low-power operation 162
8.7 Sniff Mode 163
8.7.1 Sniff Transition Mode 164
8.8 Hold Mode 165
8.9 Park State 165
8.9.1 Beacon train 166
8.9.2 Beacon access window 168
8.9.3 Parked slave synchronization 169
8.9.4 Parking 170
8.9.5 Master-initiated unparking 171
8.9.6 Slave-initiated unparking 171
8.9.7 Broadcast scan window 172
8.9.8 Polling in the park state 172

9 Audio 173
9.1 LOG PCM CODEC 173
9.2 CVSD CODEC 173
9.3 Error Handling 176
9.4 General Audio Requirements 176
9.4.1 Signal levels 176
9.4.2 CVSD audio quality 176

10 List of Figures 177

11 List of Tables 181

Appendix A:
General Audio Recommendations 182

Appendix B:
Timers 185

Appendix C:
Recommendations for AFH Operation in Park, Hold and Sniff 187

1 GENERAL DESCRIPTION

This part specifies the normal operation of a Bluetooth baseband.

The Bluetooth system provides a point-to-point connection or a point-to-multi-point connection, see (a) and (b) in Figure 1.1 on page 53. In a point-to-point connection the physical channel is shared between two Bluetooth devices. In a point-to-multipoint connection, the physical channel is shared among several Bluetooth devices. Two or more devices sharing the same physical channel form a *piconet*. One Bluetooth device acts as the master of the piconet, whereas the other device(s) act as slave(s). Up to seven slaves can be active in the piconet. Additionally, many more slaves can remain connected in a parked state. These parked slaves are not active on the channel, but remain synchronized to the master and can become active without using the connection establishment procedure. Both for active and parked slaves, the channel access is controlled by the master.

Piconets that have common devices are called a *scatternet,* see (c) in Figure 1.1 on page 53. Each piconet only has a single master, however, slaves can participate in different piconets on a time-division multiplex basis. In addition, a master in one piconet can be a slave in other piconets. Piconets shall not be frequency synchronized and each piconet has its own hopping sequence.

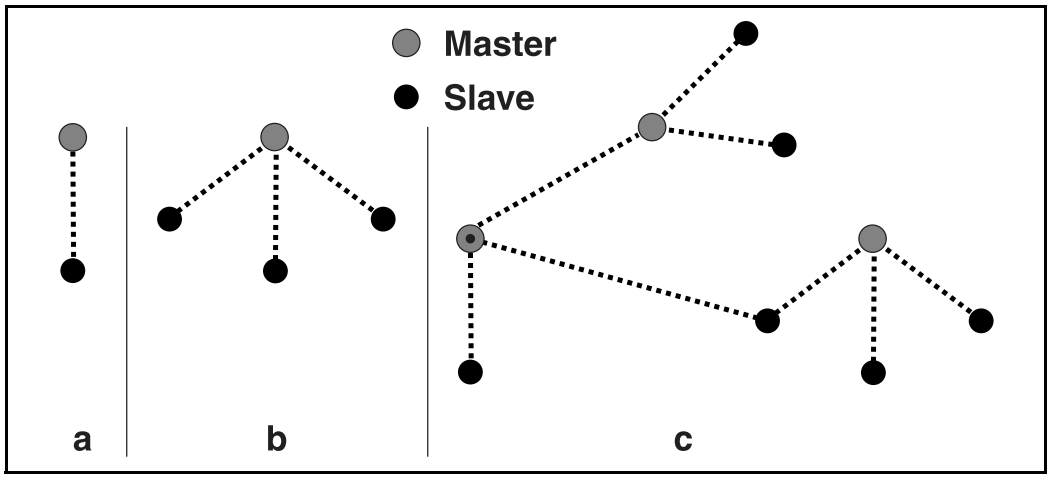


Figure 1.1: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c).

Data is transmitted over the air in packets. The general packet format is shown in Figure 1.2. Each packet consists of 3 entities: the access code, the header, and the payload.

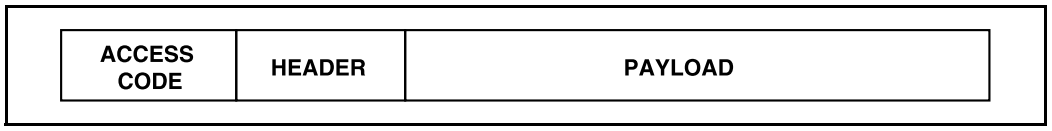


Figure 1.2: Standard packet format.

1.1 BLUETOOTH CLOCK

Every Bluetooth device shall have a native clock that shall be derived from a free running system clock. For synchronization with other devices, offsets are used that, when added to the native clock, provide temporary Bluetooth clocks that are mutually synchronized. It should be noted that the Bluetooth clock has no relation to the time of day; it may therefore be initialized to any value. The clock has a cycle of about a day. If the clock is implemented with a counter, a 28-bit counter is required that shall wrap around at 2^{28}-1. The least significant bit (LSB) shall tick in units of 312.5 μs (i.e. half a time slot), giving a clock rate of 3.2 kHz.

The clock determines critical periods and triggers the events in the device. Four periods are important in the Bluetooth system: 312.5 μs, 625 μs, 1.25 ms, and 1.28 s; these periods correspond to the timer bits CLK_0, CLK_1, CLK_2, and CLK_{12}, respectively, see Figure 1.3 on page 54.

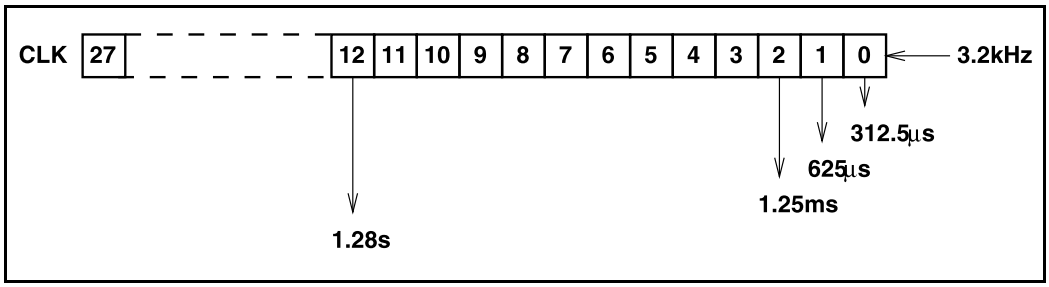


Figure 1.3: Bluetooth clock.

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

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

CLKN is the native clock and shall be the reference to all other clock appearances. In STANDBY and in Park, Hold and Sniff mode the native clock may be driven by a low power oscillator (LPO) with worst case accuracy (+/-250ppm). Otherwise, the native clock shall be driven by the reference crystal oscillator with worst case accuracy of +/-20ppm.

See Section 2.2.4 on page 60 for the definition of CLK and Section 2.4.1 on page 64 for the definition of CLKE.

The master shall never adjust its native clock during the existence of the piconet.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 212

1.2 BLUETOOTH DEVICE ADDRESSING

Each Bluetooth device shall be allocated a unique 48-bit Bluetooth device address (BD_ADDR). This address shall be obtained from the IEEE Registration Authority. The address is divided into the following three fields:

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

The LAP and UAP form the significant part of the BD_ADDR. The bit pattern in Figure 1.4 is an example BD_ADDR.

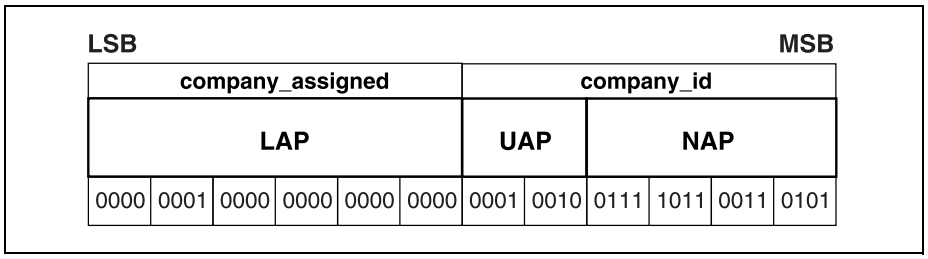


Figure 1.4: Format of BD_ADDR.

The BD_ADDR may take any values except the 64 reserved LAP values for general and dedicated inquiries (see Section 1.2.1 on page 55).

1.2.1 Reserved addresses

A block of 64 contiguous LAPs is reserved for inquiry operations; one LAP common to all devices is reserved for general inquiry, the remaining 63 LAPs are reserved for dedicated inquiry of specific classes of devices (see Assigned Numbers on the website[1]). The same LAP values are used regardless of the contents of UAP and NAP. Consequently, none of these LAPs can be part of a user BD_ADDR.

The reserved LAP addresses are 0x9E8B00-0x9E8B3F. The general inquiry LAP is 0x9E8B33. All addresses have the LSB at the rightmost position, hexadecimal notation. The default check initialization (DCI) is used as the UAP whenever one of the reserved LAP addresses is used. The DCI is defined to be 0x00 (hexadecimal).

1. https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers

1.3 ACCESS CODES

In the Bluetooth system all transmissions over the physical channel begin with an access code. Three different access codes are defined, see also Section 6.3.1 on page 98:

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

All access codes are derived from the LAP of a device address or an inquiry address. The device access code is used during **page**, **page scan** and **page response** substates and shall be derived from the paged device's BD_ADDR. The channel access code is used in the **CONNECTION** state and forms the beginning of all packets exchanged on the piconet physical channel. The channel access code shall be derived from the LAP of the master's BD_ADDR. Finally, the inquiry access code shall be used in the **inquiry** substate. There is one general IAC (GIAC) for general inquiry operations and there are 63 dedicated IACs (DIACs) for dedicated inquiry operations.

The access code also indicates to the receiver the arrival of a packet. It is used for timing synchronization and offset compensation. The receiver correlates against the entire synchronization word in the access code, providing very robust signalling.

2 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 the combination of a pseudo-random frequency hopping sequence, the specific slot timing of the transmissions, the access code and packet header encoding. These aspects, together with the range of the transmitters, define the signature of the physical channel. For the basic and adapted piconet physical channels frequency hopping is used to change frequency periodically to reduce the effects of interference and to satisfy local regulatory requirements.

Two devices that wish to communicate use a shared physical channel for this communication. To achieve this, their transceivers must be tuned to the same RF frequency at the same time, and they must 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 always present at the start of every transmitted packet.

Four Bluetooth physical 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 physical channels are used for discovering (the inquiry scan channel) and connecting (the page scan channel) Bluetooth devices.

A Bluetooth device can only use one of these 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 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.) At a minimum, a device need only be capable of connection to one physical channel at a time, however, advanced devices may be capable of connecting simultaneously to more than one physical channel, but the specification does not assume that this is possible.

2.1 PHYSICAL CHANNEL DEFINITION

Physical channels are defined by a pseudo-random RF channel hopping sequence, the packet (slot) timing and an access code. The hopping sequence is determined by the UAP and LAP of a Bluetooth device address and the selected hopping sequence. The phase in the hopping sequence is determined by the Bluetooth clock. All physical channels are subdivided into time slots whose length is different depending on the physical channel. Within the physical channel, each reception or transmission event is associated with a time slot or time slots. For each reception or transmission event an RF channel is selected by the hop selection kernel (see Section 2.6 on page 70).The maximum hop rate is 1600 hops/s in the **CONNECTION** state and the maximum is 3200 hops/s in the **inquiry** and **page** substates.

The following physical channels are defined:

- basic piconet physical channel
- adapted piconet physical channel
- page scan physical channel
- inquiry scan physical channel

2.2 BASIC PICONET PHYSICAL CHANNEL

During the **CONNECTION** state the basic piconet physical channel is used by default. The adapted piconet physical channel may also be used. The adapted piconet physical channel is identical to the basic piconet physical channel except for the differences listed in Section 2.3 on page 63.

2.2.1 Master-slave definition

The basic piconet physical channel is defined by the master of the piconet. The master controls the traffic on the piconet physical channel by a polling scheme. (see Section 8.5 on page 148)

By definition, the device that initiates a connection by paging is the master. Once a piconet has been established, master-slave roles may be exchanged. This is described in Section 8.6.5 on page 155.

2.2.2 Hopping characteristics

The basic piconet physical channel is characterized by a pseudo-random hopping through all 79 RF channels. The frequency hopping in the piconet physical channel is determined by the Bluetooth clock and BD_ADDR of the master. When the piconet is established, the master clock is communicated to the slaves. Each slave shall add an offset to its native clock to synchronize with the master clock. Since the clocks are independent, the offsets must be updated regularly. All devices participating in the piconet are time-synchronized and hop-synchronized to the channel.

The basic piconet physical channel uses the basic channel hopping sequence and is described in Section 2.6 on page 70.

2.2.3 Time slots

The basic piconet physical channel is divided into time slots, each 625 µs in length. The time slots are numbered according to the most significant 27 bits of the Bluetooth clock CLK_{28-1} of the piconet master. The slot numbering ranges from 0 to 2^{27}-1 and is cyclic with a cycle length of 2^{27}. The time slot number is denoted as k.

A TDD scheme is used where master and slave alternatively transmit, see Figure 2.1 on page 59. The packet start shall be aligned with the slot start. Packets may extend over up to five time slots.

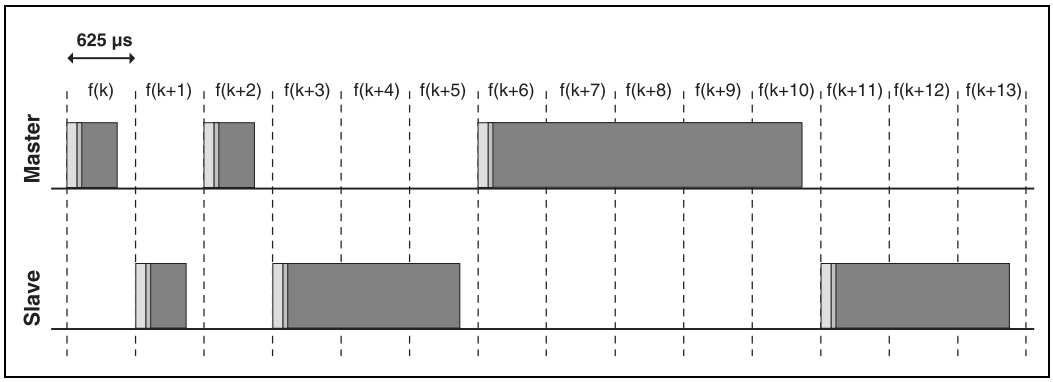


Figure 2.1: Multi-slot packets

The term *slot pairs* is used to indicate two adjacent time slots starting with a master-to-slave transmission slot.

2.2.4 Piconet clocks

CLK is the master clock of the piconet. It shall be used for all timing and scheduling activities in the piconet. All devices shall use the CLK to schedule their transmission and reception. The CLK shall be derived from the native clock CLKN (see Section 1.1 on page 54) by adding an offset, see Figure 2.2 on page 60. The offset shall be zero for the master since CLK is identical to its own native clock CLKN. Each slave shall add an appropriate offset to its CLKN such that the CLK corresponds to the CLKN of the master. Although all CLKNs in the devices run at the same nominal rate, mutual drift causes inaccuracies in CLK. Therefore, the offsets in the slaves must be regularly updated such that CLK is approximately CLKN of the master.

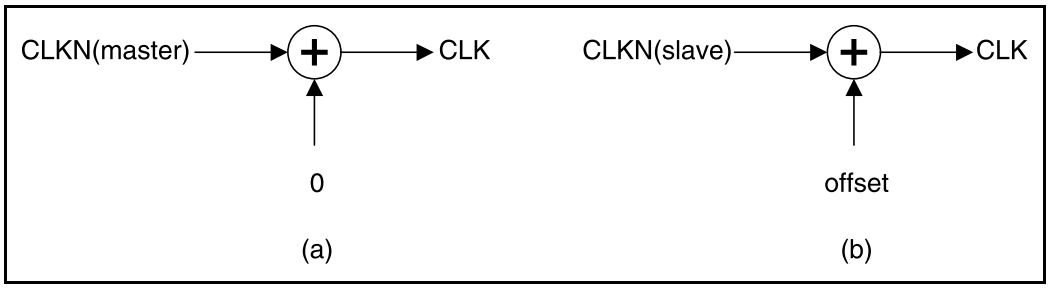


Figure 2.2: Derivation of CLK in master (a) and in slave (b).

2.2.5 Transmit/receive timing

The master transmission shall always start at even numbered time slots (CLK_1=0) and the slave transmission shall always start at odd numbered time slots (CLK_1=1). Due to packet types that cover more than a single slot, master transmission may continue in odd numbered slots and slave transmission may continue in even numbered slots, see Figure 2.1 on page 59.

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

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

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

where T_N is the nominal slot length (625 µs), j_k denotes jitter ($|j_k| \leq 1$ µs) at the start of slot k, and, d_k, denotes the drift ($|d_k| \leq 20$ ppm) within slot k. The jitter and drift may vary arbitrarily within the given limits for every slot, while *offset* is an arbitrary but fixed constant. For hold, park and sniff the drift and jitter parameters specified in Link Manager Protocol [Part C] Section 4.3.1 on page 237 apply.

2.2.5.1 Piconet physical channel timing

In the figures, only single-slot packets are shown as an example.

The master TX/RX timing is shown in Figure 2.3 on page 61. In Figure 2.3 and Figure 2.4 the channel hopping frequencies are indicated by *f(k)* where k is the time slot number. After transmission, a return packet is expected N × 625 µs after the start of the TX packet where N is an odd, integer larger than 0. N depends on the type of the transmitted packet.

To allow for some time slipping, an uncertainty window is defined around the exact receive timing. During normal operation, the window length shall be 20 µs, which allows the RX packet to arrive up to 10 µs too early or 10 µs too late. It is recommended that slaves implement variable sized windows or time tracking to accomodate a master's absence of more than 250ms.

During the beginning of the RX cycle, the access correlator shall search for the correct channel access code over the uncertainty window. If an event trigger does not occur the receiver may go to sleep until the next RX event. If in the course of the search, it becomes apparent that the correlation output will never exceed the final threshold, the receiver may go to sleep earlier. If a trigger event occurs, the receiver shall remain open to receive the rest of the packet unless the packet is for another device, a non-recoverable header error is detected, or a non-recoverable payload error is detected.

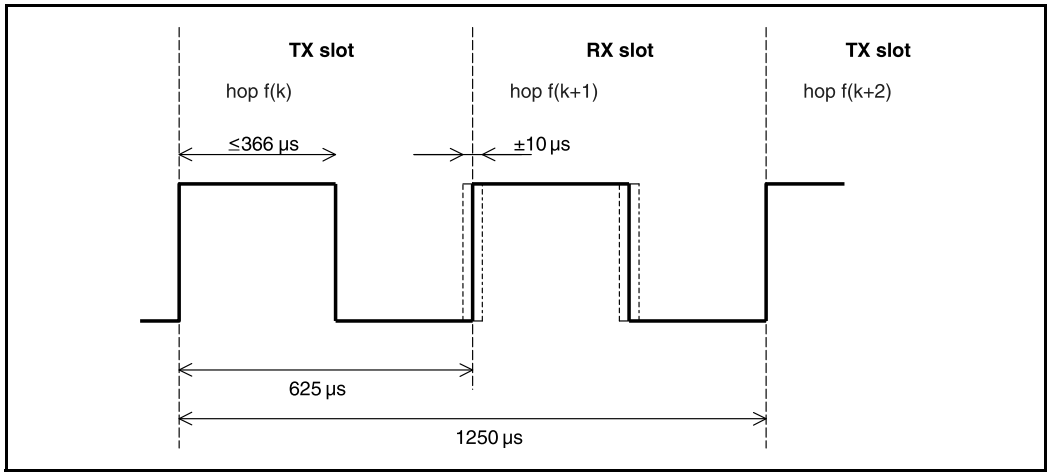


Figure 2.3: RX/TX cycle of master transceiver in normal mode for single-slot packets.

Each master transmission shall be derived from bit 2 of the Master's native Bluetooth clock, thus the current transmission will be scheduled Mx1250µs after the start of the previous master TX burst where M depends on the transmitted and received packet type and is an even, integer larger than 0. The master TX timing shall be derived from the master's native Bluetooth clock, and thus it will not be affected by time drifts in the slave(s).

Slaves maintain an estimate of the master's native clock by adding a timing offset to the slave's native clock (see Section 2.2.4 on page 60). This offset shall be updated each time a packet is received from the master. By comparing the exact RX timing of the received packet with the estimated RX timing, slaves shall correct the offset for any timing misalignments. Since only the channel access code is required to synchronize the slave, slave RX timing can be corrected with any packet sent in the master-to-slave transmission slot.

The slave's TX/RX timing is shown in Figure 2.4 on page 62. The slave's transmission shall be scheduled N × 625µs after the start of the slave's RX packet where N is an odd, positive integer larger than 0. If the slave's RX timing drifts, so will its TX timing. During periods when a slave is in the active mode (see Section 8.6 on page 149) and is not able to receive any valid channel access codes from the master, the slave may increase its receive uncertainty window and/or use predicted timing drift to increase the probability of receiving the master's bursts when reception resumes.

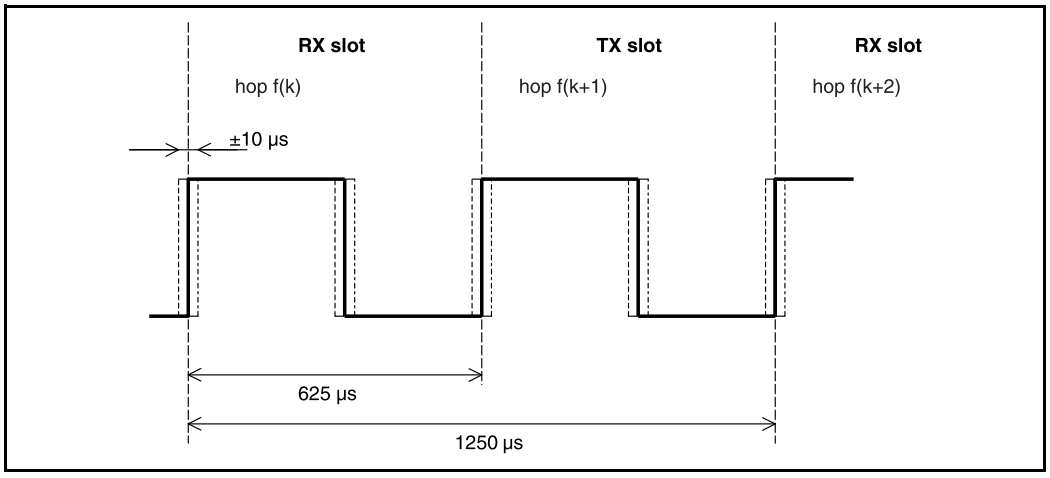


Figure 2.4: RX/TX cycle of slave transceiver in normal mode for single-slot packets.

2.2.5.2 Piconet physical channel re-synchronization

In the piconet physical channel, a slave may loose synchronization if it does not receive a packet from the master at least every 250ms (or less if the low power clock is used). This may occur in sniff, hold, park, in a scatternet or due to interference. When re-synchronizing to the piconet physical channel a slave device shall listen for the master before it may send information. In this case, the length of the search window in the slave device may be increased from 20 µs to a larger value X µs as illustrated in Figure 2.5 on page 63. Note that only RX hop frequencies are used. The hop frequency used in the master-to-slave (RX) slot shall also be used in the uncertainty window, even when it is extended into the preceding time interval normally used for the slave-to-master (TX) slot.

If the length of search window, X, exceeds 1250 µs, consecutive windows shall avoid overlapping search windows. Consecutive windows should instead be centered at *f(k), f(k+4), ... f(k+4i)* (where 'i' is an integer), which gives a maximum value X=2500 µs, or even at *f(k), f(k+6), ...f(k+6i)* which gives a maximum value X=3750 µs. The RX hop frequencies used shall correspond to the master-to-slave transmission slots.

It is recommended that single slot packets are transmitted by the master during slave re-synchronization.

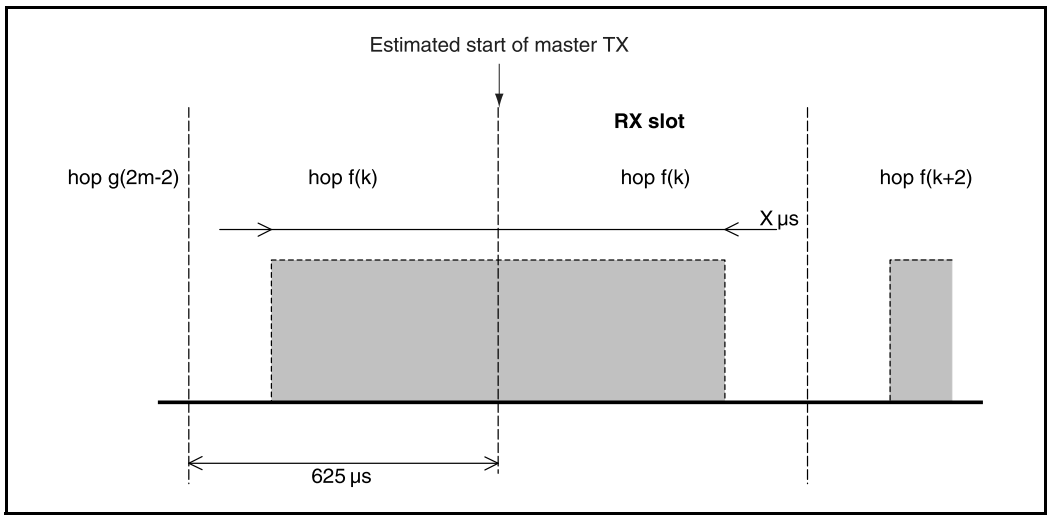


Figure 2.5: RX timing of slave returning from hold mode.

2.3 ADAPTED PICONET PHYSICAL CHANNEL

2.3.1 Hopping characteristics

The adapted piconet physical channel shall use at least N_{min} RF channels (where N_{min} is 20).

The adapted piconet physical channel uses the adapted channel hopping sequence described in Section 2.6 on page 70.

Adapted piconet physical channels can be used for connected devices that have adaptive frequency hopping (AFH) enabled. There are two distinctions between basic and adapted piconet physical channels. The first is that the same channel mechanism that makes the slave frequency the same as the preceding master transmission. The second aspect is that the adapted piconet physical channel may be based on less than the full 79 frequencies of the basic piconet physical channel.

2.4 PAGE SCAN PHYSICAL CHANNEL

Although master and slave roles are not defined prior to a connection, the term *master* is used for the paging device (that becomes a master in the **CONNECTION** state) and *slave* is used for the page scanning device (that becomes a slave in the **CONNECTION** state).

2.4.1 Clock estimate for paging

A paging device uses an estimate of the native clock of the page scanning device, CLKE; i.e. an offset shall be added to the CLKN of the pager to approximate the CLKN of the recipient, see Figure 2.6 on page 64. CLKE shall be derived from the reference CLKN by adding an offset. By using the CLKN of the recipient, the pager might be able to speed up the connection establishment.

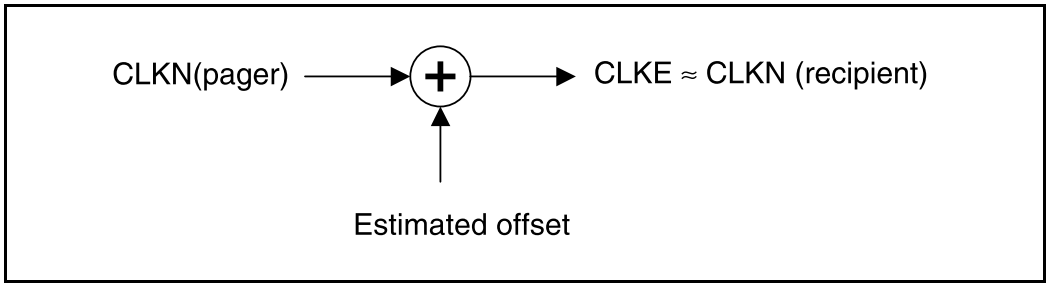


Figure 2.6: Derivation of CLKE.

2.4.2 Hopping characteristics

The page scan physical channel follows a slower hopping pattern than the basic piconet physical channel and is a short pseudo-random hopping sequence through the RF channels. The timing of the page scan channel shall be determined by the native Bluetooth clock of the scanning device. The frequency hopping sequence is determined by the Bluetooth address of the scanning device.

The page scan physical channel uses the page, master page response, slave page response, and page scan hopping sequences specified in Section 2.6 on page 70.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 222

2.4.3 Paging procedure timing

During the paging procedure, the master shall transmit paging messages (see Table 8.3 on page 139) corresponding to the slave to be connected. Since the paging message is a very short packet, the hop rate is 3200 hops/s. In a single TX slot interval, the paging device shall transmit on two different hop frequencies. In Figure 2.7 through Figure 2.11, *f(k)* is used for the frequencies of the page hopping sequence and *f'(k)* denotes the corresponding page response sequence frequencies. The first transmission starts where $CLK_0 = 0$ and the second transmission starts where $CLK_0 = 1$.

In a single RX slot interval, the paging device shall listen for the slave page response message on two different hop frequencies. Similar to transmission, the nominal reception starts where $CLK_0 = 0$ and the second reception nominally starts where $CLK_0 = 1$; see Figure 2.7 on page 65. During the TX slot, the paging device shall send the paging message at the TX hop frequencies *f(k)* and *f(k+1)*. In the RX slot, it shall listen for a response on the corresponding RX hop frequencies *f'(k)* and *f'(k+1)*. The listening periods shall be exactly timed 625 µs after the corresponding paging packets, and shall include a ±10 µs uncertainty window.

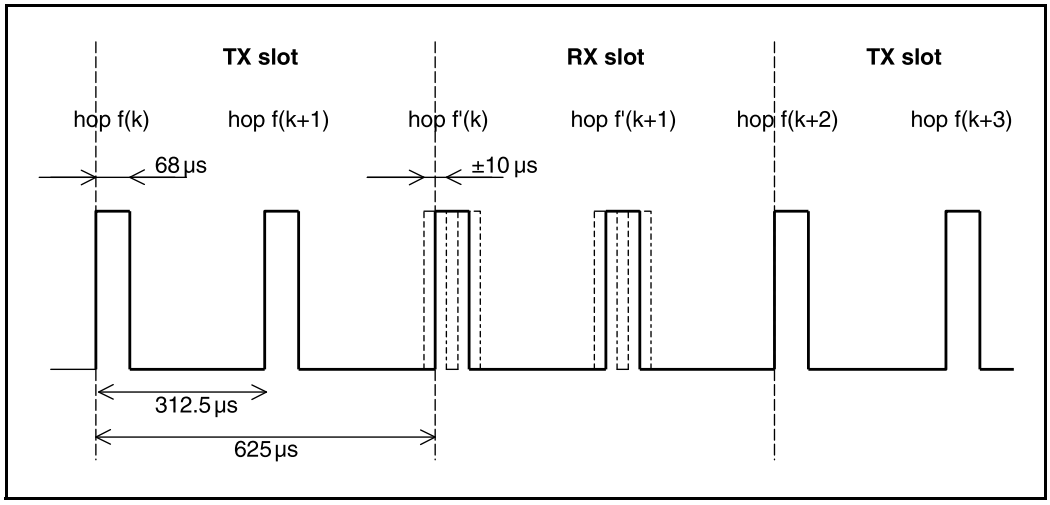


Figure 2.7: RX/TX cycle of transceiver in PAGE mode.

2.4.4 Page response timing

At connection setup a master page response packet is transmitted from the master to the slave (see Table 8.3 on page 139). This packet establishes the timing and frequency synchronization. After the slave device has received the page message, it shall return a response message that consists of the slave page response packet and shall follow 625 µs after the receipt of the page message. The master shall send the master page response packet in the TX slot following the RX slot in which it received the slave response, according to the RX/TX timing of the master. The time difference between the slave page response and master page response message will depend on the timing of the page message the slave received. In Figure 2.8 on page 66, the slave receives the paging message sent **first** in the master-to-slave slot. It then responds with a first slave page response packet in the first half of the slave-to-master slot. The timing of the master page response packet is based on the timing of the page message sent first in the preceding master-to-slave slot: there is an exact 1250 µs delay between the first page message and the master page response packet. The packet is sent at the hop frequency *f(k*+1*)* which is the hop frequency following the hop frequency *f(k)* the page message was received in.

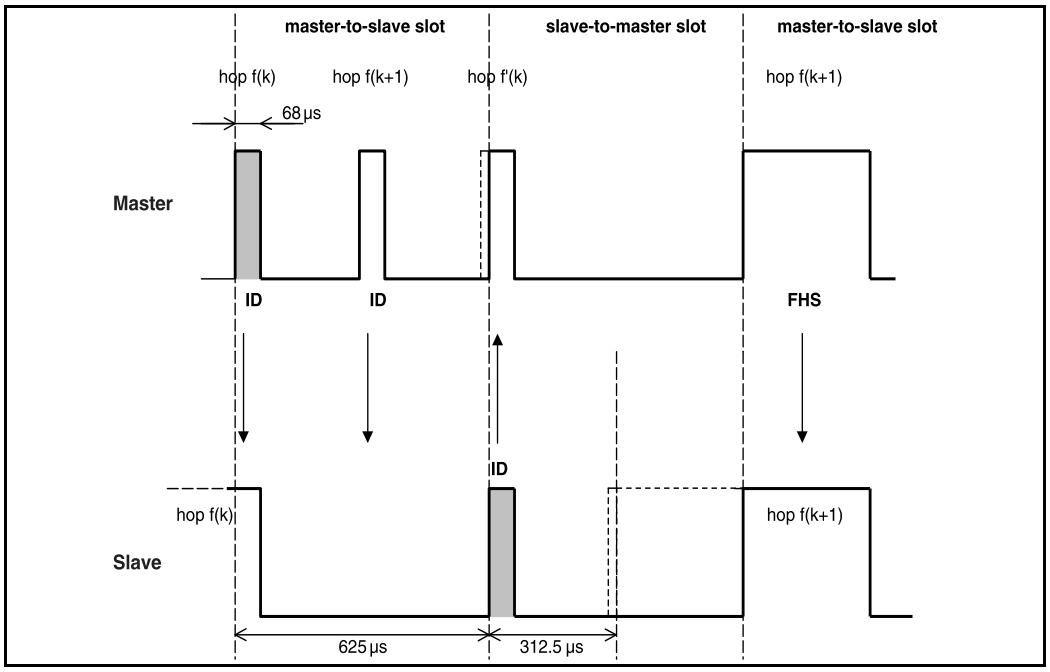


Figure 2.8: Timing of page response packets on successful page in first half slot

In Figure 2.9 on page 67, the slave receives the paging message sent **second** in the master-to-slave slot. It then responds with a slave page response packet in the second half of the slave-to-master slot exactly 625 µs after the receipt of the page message. The timing of the master page response packet is still based on the timing of the page message sent **first** in the preceding master-to-slave slot: there is an exact 1250 µs delay between the **first** page message and the master page response packet. The packet is sent at the hop frequency *f(k+2)* which is the hop frequency following the hop frequency *f(k+1)* the page message was received in.

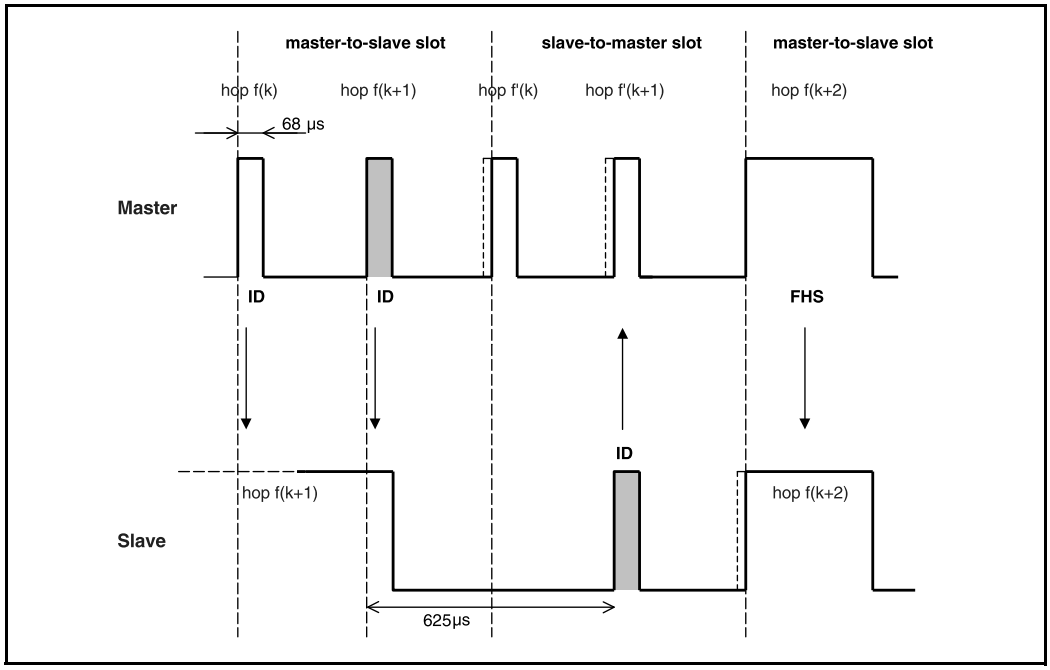


Figure 2.9: Timing of page response packets on successful page in second half slot

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

2.5 INQUIRY SCAN PHYSICAL CHANNEL

Although master and slave roles are not defined prior to a connection, the term *master* is used for the inquiring device and *slave* is used for the inquiry scanning device.

2.5.1 Clock for inquiry

The clock used for inquiry and inquiry scan shall be the device's native clock.

2.5.2 Hopping characteristics

The inquiry scan channel follows a slower hopping pattern than the piconet physical channel and is a short pseudo-random hopping sequence through the RF channels. The timing of the inquiry scan channel is determined by the native Bluetooth clock of the scanning device while the frequency hopping sequence is determined by the general inquiry access code.

The inquiry scan physical channel uses the inquiry, inquiry response, and inquiry scan hopping sequences described in Section 2.6 on page 70.

2.5.3 Inquiry procedure timing

During the inquiry procedure, the master shall transmit inquiry messages with the general or dedicated inquiry access code. The timing for inquiry is the same as for paging (see Section 2.4.3 on page 65).

2.5.4 Inquiry response timing

An inquiry response packet is transmitted from the slave to the master after the slave has received an inquiry message (see Table 8.5 on page 147). This packet contains information necessary for the inquiring master to page the slave (see definition of the FHS packet in Section 6.5.1.4 on page 106) and follows 625µs after the receipt of the inquiry message. In figures Figure 2.10 and Figure 2.11, *f(k)* is used for the frequencies of the inquiry hopping sequence and f'(k) denotes the corresponding inquiry response sequence frequency. The packet is received by the master at the hop frequency *f'(k)* when the inquiry message received by the slave was first in the master-to-slave slot.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 226

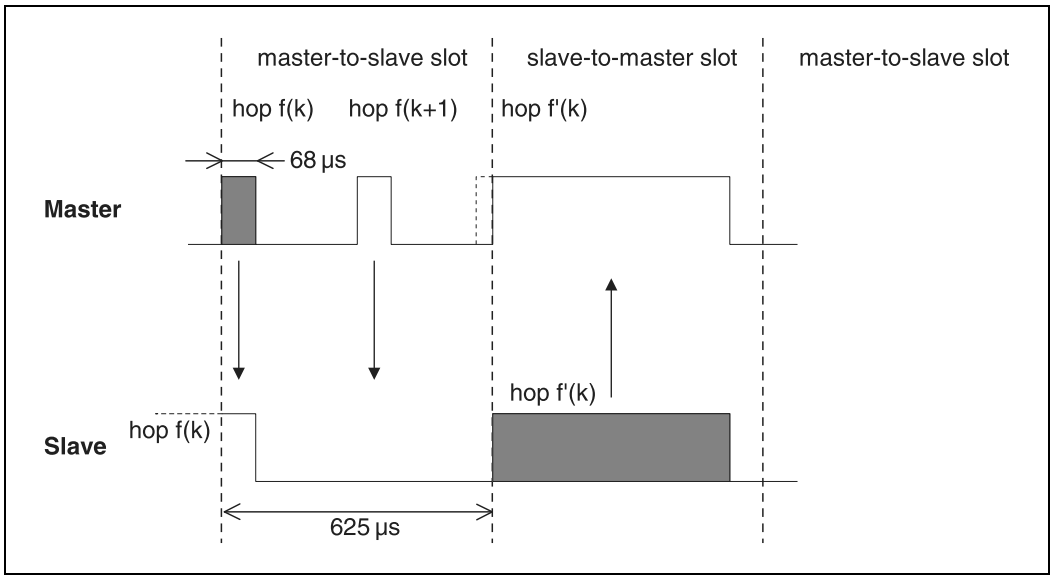


Figure 2.10: Timing of inquiry response packet on successful inquiry in first half slot

When the inquiry message received by the slave was the second in the master-to-slave slot the packet is received by the master at the hop frequency *f'(k+1)*.

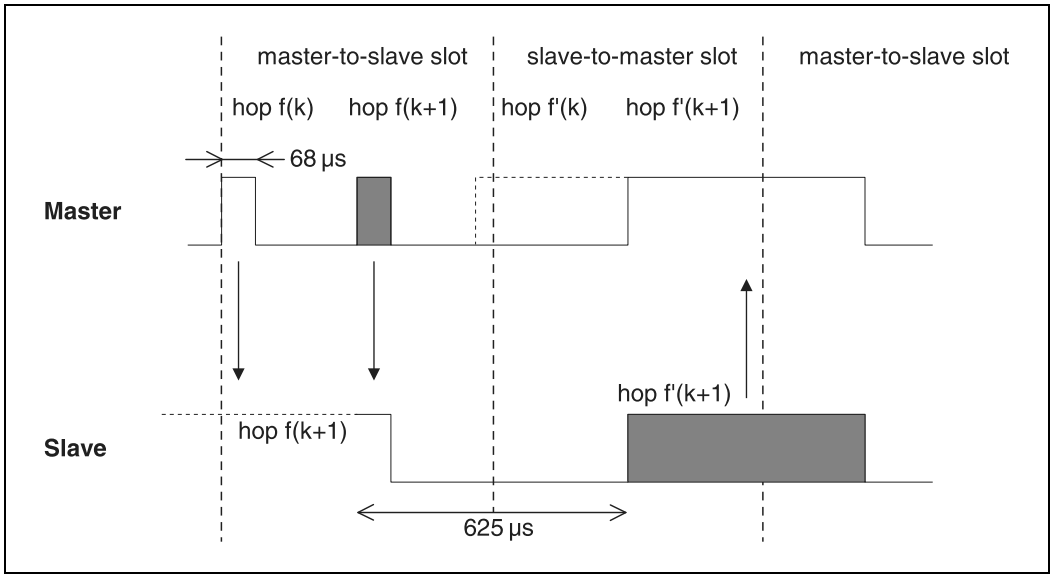


Figure 2.11: Timing of inquiry response packet on successful inquiry in second half slot

2.6 HOP SELECTION

Bluetooth devices shall use the hopping kernel as defined in the following sections.

In total, six types of hopping sequence are defined – five for the basic hop system and one for an adapted set of hop locations used by adaptive frequency hopping (AFH). These sequences are:

- A **page hopping sequence** with 32 wake-up frequencies distributed equally over the 79 MHz, with a period length of 32;
- A **page response hopping sequence** covering 32 response frequencies that are in a one-to-one correspondence to the current page hopping sequence. The master and slave use different rules to obtain the same sequence;
- An **inquiry hopping sequence** with 32 wake-up frequencies distributed equally over the 79 MHz, with a period length of 32;
- An **inquiry response hopping sequence** covering 32 response frequencies that are in a one-to-one correspondence to the current inquiry hopping sequence.
- A **basic channel hopping sequence** which has a very long period length, which does not show repetitive patterns over a short time interval, and which distributes the hop frequencies equally over the 79 MHz during a short time interval.
- An **adapted channel hopping sequence** derived from the basic channel hopping sequence which uses the same channel mechanism and may use fewer than 79 frequencies. The adapted channel hopping sequence is only used in place of the basic channel hopping sequence. All other hopping sequences are not affected by hop sequence adaptation.

2.6.1 General selection scheme

The selection scheme consists of two parts:

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

The general block diagram of the hop selection scheme is shown in Figure 2.12 on page 71. The mapping from the input to a particular RF channel index is performed in the selection box.

The inputs to the selection box are the selected clock, frozen clock, N, k_{offset}, address, sequence selection and AFH_channel_map. The source of the clock input depends on the hopping sequence selected. Additionally, each hopping sequence uses different bits of the clock (see Table 2.2 on page 79). N and k_{offset} are defined in Section 2.6.4 on page 78.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 228

The *sequence selection* input can be set to the following values:

- page scan
- inquiry scan
- page
- inquiry
- master page response
- slave page response
- inquiry response
- basic channel
- adapted channel

The address input consists of 28 bits including the entire LAP and the 4 LSBs of the UAP. This is designated as the UAP/LAP. When the basic or adapted channel hopping sequence is selected, the Bluetooth device address of the master (BD_ADDR) shall be used. When the page, master page response, slave page response, or page scan hopping sequences are selected the BD_ADDR given by the Host of the paged device shall be used (see HCI Create Connection Command [Part E] Section 7.1.5 on page 380). When the inquiry, inquiry response, or inquiry scan hopping sequences are selected, the UAP/LAP corresponding to the GIAC shall be used even if it concerns a DIAC. Whenever one of the reserved BD_ADDRs (see Section 1.2.1 on page 55) is used for generating a frequency hop sequence, the UAP shall be replaced by the default check initialization (DCI, see Section 7.1 on page 118). The hopping sequence is selected by the sequence selection input to the selection box.

When the adapted channel hopping sequence is selected, the *AFH_channel_map* is an additional input to the selection box. The *AFH_channel_map* indicates which channels shall be *used* and which shall be *unused*. These terms are defined in Section 2.6.3 on page 77.

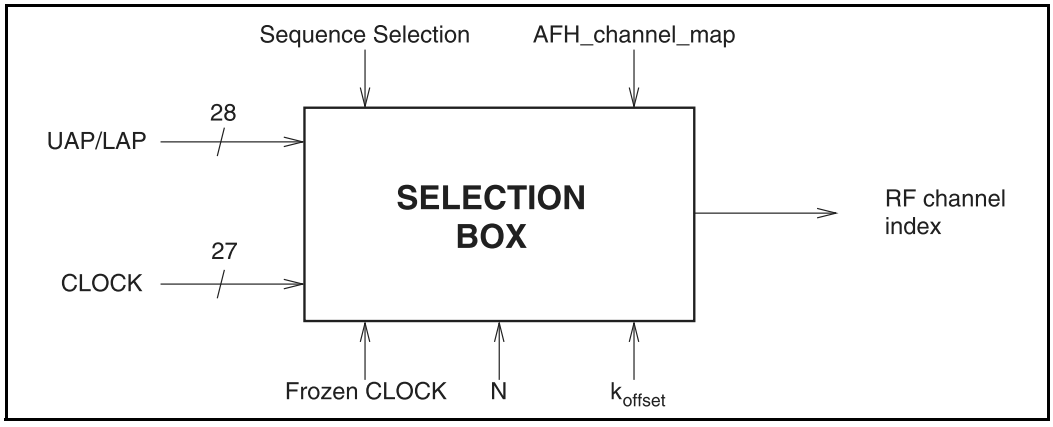


Figure 2.12: General block diagram of hop selection scheme.

The output, *RF channel index*, constitutes a pseudo-random sequence. The RF channel index is mapped to RF channel frequencies using the equation in Table 2.1 on page 31 in the Radio Specification.

The selection scheme chooses a segment of 32 hop frequencies spanning about 64 MHz and visits these hops in a pseudo-random order. Next, a different 32-hop segment is chosen, etc. In the page, master page response, slave page response, page scan, inquiry, inquiry response and inquiry scan hopping sequences, the same 32-hop segment is used all the time (the segment is selected by the address; different devices will have different paging segments). When the basic channel hopping sequence is selected, the output constitutes a pseudo-random sequence that slides through the 79 hops. The principle is depicted in Figure 2.13 on page 72.

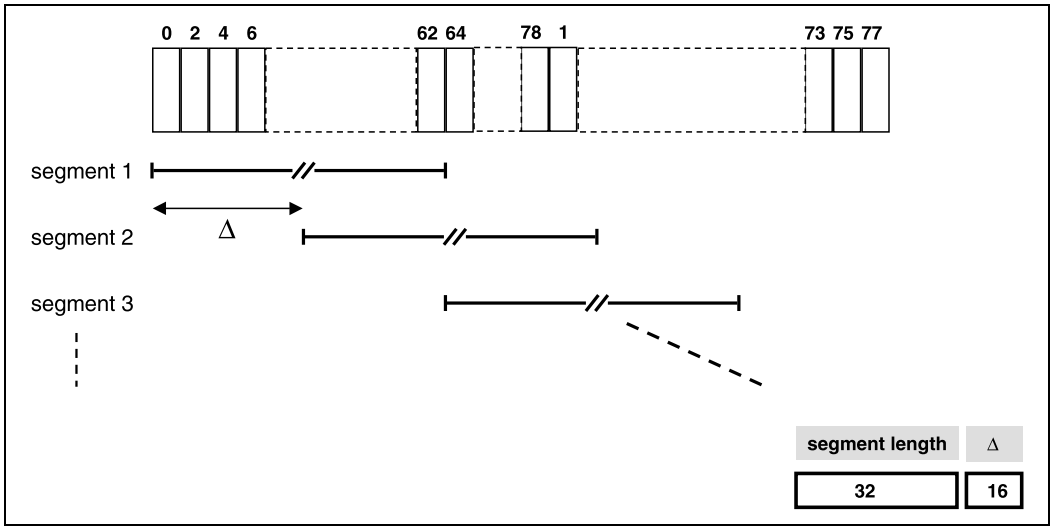


Figure 2.13: Hop selection scheme in CONNECTION state.

The RF frequency shall remain fixed for the duration of the packet. The RF frequency for the packet shall be derived from the Bluetooth clock value in the first slot of the packet. The RF frequency in the first slot after a multi-slot packet shall use the frequency as determined by the Bluetooth clock value for that slot. Figure 2.14 on page 73 illustrates the hop definition on single- and multi-slot packets.

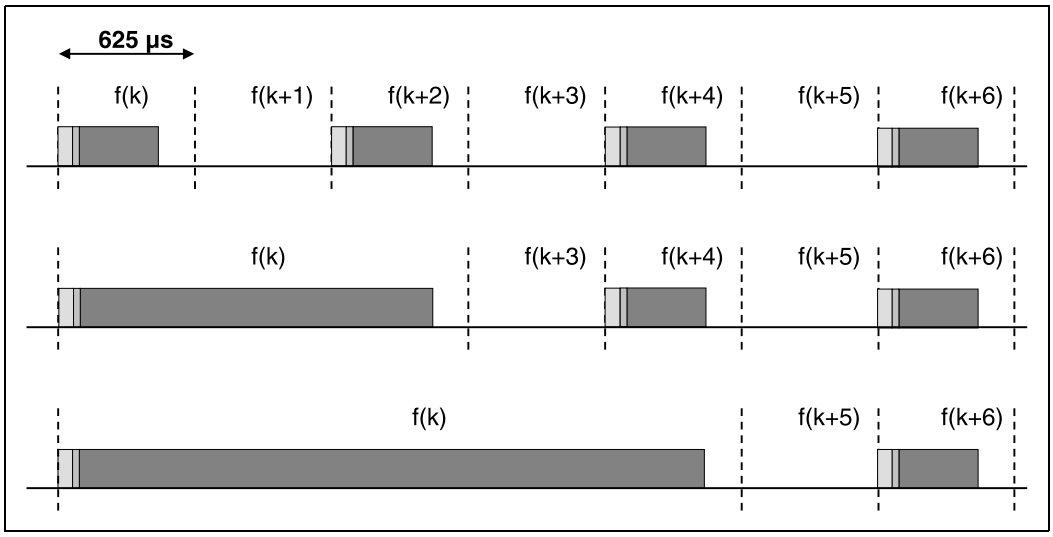


Figure 2.14: Single- and multi-slot packets.

When the adapted channel hopping sequence is used, the pseudo-random sequence contains only frequencies that are in the RF channel set defined by the *AFH_channel_map* input. The adapted sequence has similar statistical properties to the non-adapted hop sequence. In addition, the slave responds with its packet on the same RF channel that was used by the master to address that slave (or would have been in the case of a synchronous reserved slot without a validly received master-to-slave transmission). This is called the *same channel mechanism* of AFH. Thus, the RF channel used for the master to slave packet is also used for the immediately following slave to master packet. An example of the same channel mechanism is illustrated in Figure 2.15 on page 73. The same channel mechanism shall be used whenever the adapted channel hopping sequence is selected.

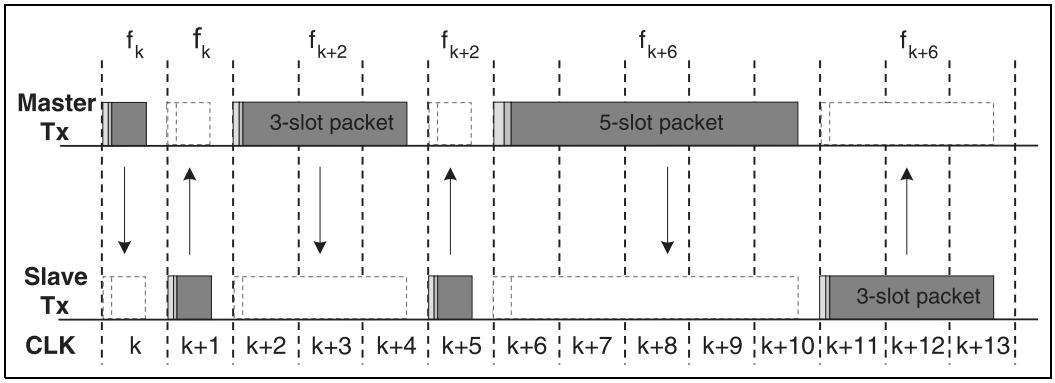


Figure 2.15: Example of the same channel mechanism.

2.6.2 Selection kernel

The basic hop selection kernel shall be as shown in Figure 2.16 on page 74 and is used for the page, page response, inquiry, inquiry response and basic channel hopping selection kernels. In these substates the AFH_channel_map input is unused. The adapted channel hopping selection kernel is described in Section 2.6.3 on page 77. The X input determines the phase in the 32-hop segment, whereas Y1 and Y2 selects between master-to-slave and slave-to-master. The inputs A to D determine the ordering within the segment, the inputs E and F determine the mapping onto the hop frequencies. The kernel addresses a register containing the RF channel indices. This list is ordered so that first all even RF channel indices are listed and then all odd hop frequencies. In this way, a 32-hop segment spans about 64 MHz.

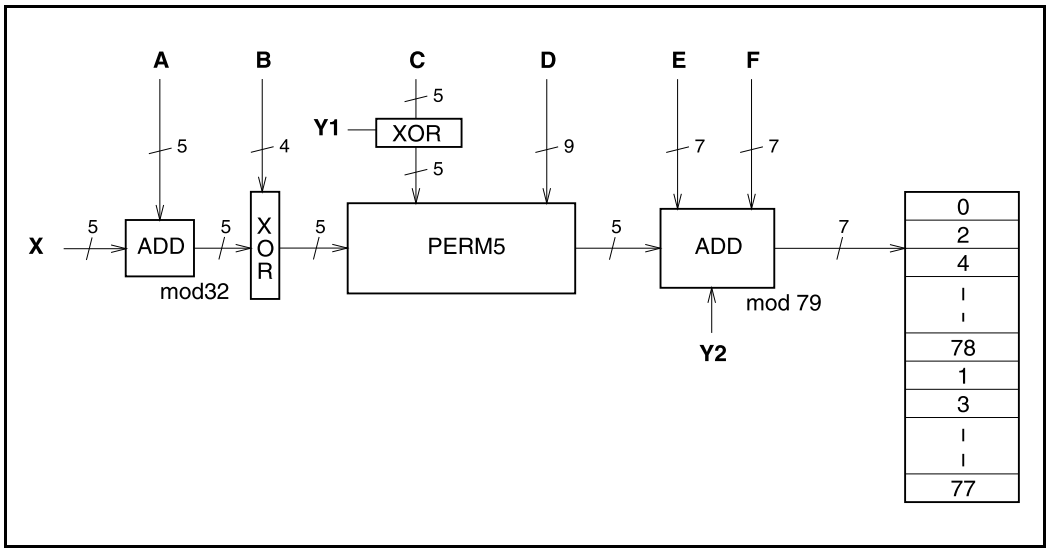


Figure 2.16: Block diagram of the basic hop selection kernel for the hop system.

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

2.6.2.1 First addition operation

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 232

2.6.2.2 XOR operation

Let Z' denote the output of the first addition. In the XOR operation, the four LSBs of Z' are modulo-2 added to the address bits $A_{22\text{-}19}$. The operation is illustrated in Figure 2.17 on page 75.

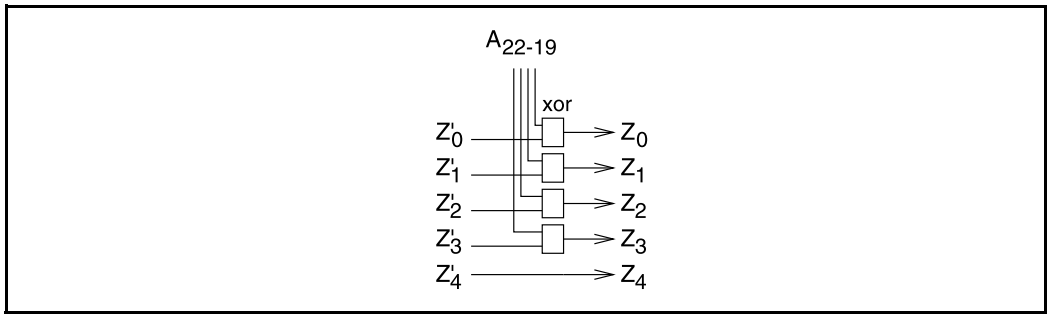


Figure 2.17: XOR operation for the hop system.

2.6.2.3 Permutation operation

The permutation operation involves the switching from 5 inputs to 5 outputs for the hop system, controlled by the control word. The permutation or switching box shall be as shown in Figure 2.18 on page 76. It consists of 7 stages of butterfly operations. The control of the butterflies by the control signals P is shown in Table 2.1. $P_{0\text{-}8}$ corresponds to $D_{0\text{-}8}$, and, P_{i+9} corresponds to $C_i \oplus Y1$ for $i = 0 \ldots 4$ in Figure 2.16 .

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

Table 2.1: Control of the butterflies for the hop system

The Z input is the output of the XOR operation as described in the previous section. The butterfly operation can be implemented with multiplexers as depicted in Figure 2.19 on page 76.

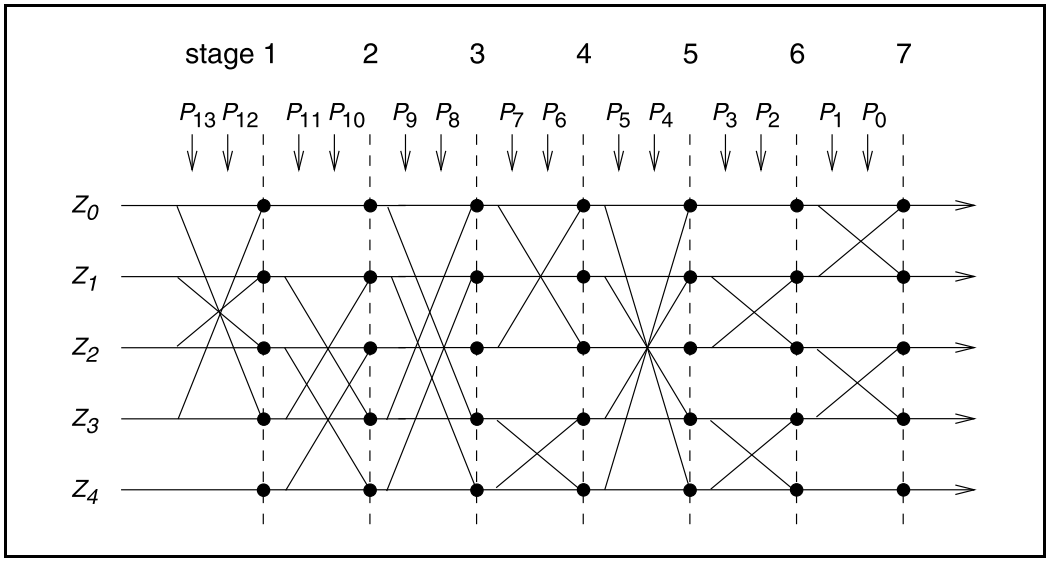


Figure 2.18: Permutation operation for the hop system.

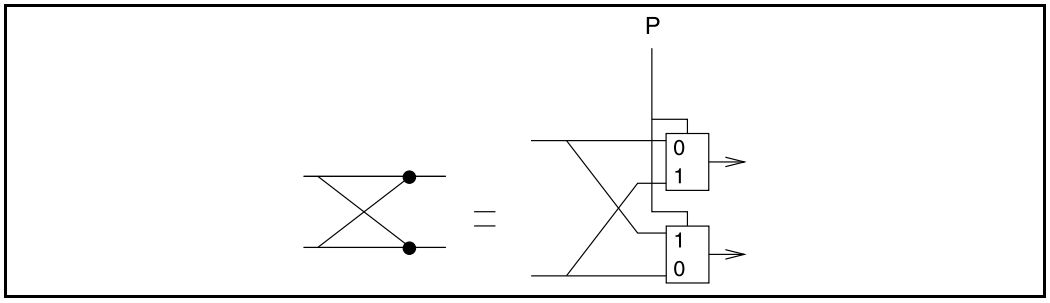


Figure 2.19: Butterfly implementation.

2.6.2.4 Second addition operation

The addition operation only adds a constant to the output of the permutation operation. The addition is applied modulo 79.

2.6.2.5 Register bank

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 234

2.6.3 Adapted hop selection kernel

The adapted hop selection kernel is based on the basic hop selection kernel defined in the preceding sections.

The inputs to the adapted hop selection kernel are the same as for the basic hop system kernel except that the input AFH_channel_map (defined in Link Manager Protocol [Part C] Section 5.2 on page 277) is used. The *AFH_channel_map* indicates which RF channels shall be *used* and which shall be *unused*. When hop sequence adaptation is enabled, the number of *used* RF channels may be reduced from 79 to some smaller value N. All devices shall be capable of operating on an adapted hop sequence (AHS) with $N_{min} \leq N \leq 79$, with any combination of *used* RF channels within the *AFH_channel_map* that meets this constraint. N_{min} is defined in Section 2.3.1 on page 63.

Adaptation of the hopping sequence is achieved through two additions to the basic channel hopping sequence according to Figure 2.16 on page 74:

- *Unused* RF channels are re-mapped uniformly onto *used* RF channels. That is, if the hop selection kernel of the basic system generates an *unused* RF channel, an alternative RF channel out of the set of *used* RF channels is selected pseudo-randomly.
- The *used* RF channel generated for the master-to-slave packet is also used for the immediately following slave-to-master packet (see Section 2.6.1 on page 70).

2.6.3.1 Channel re-mapping function

When the adapted hop selection kernel is selected, the basic hop selection kernel according to Figure 2.16 on page 74 is initially used to determine an RF channel. If this RF channel is *unused* according to the *AFH_channel_map*, the *unused* RF channel is re-mapped by the re-mapping function to one of the *used* RF channels. If the RF channel determined by the basic hop selection kernel is already in the set of *used* RF channels, no adjustment is made. The hop sequence of the (non-adapted) basic hop equals the sequence of the adapted selection kernel on all locations where *used* RF channels are generated by the basic hop. This property facilitates non-AFH slaves remaining synchronized while other slaves in the piconet are using the adapted hopping sequence.

A block diagram of the re-mapping mechanism is shown in Figure 2.20 on page 78. The re-mapping function is a post-processing step to the selection kernel from Figure 2.16 on page 74, denoted as ‘Hop selection of the basic hop’. The output f_k of the basic hop selection kernel is an RF channel number that ranges between 0 and 78. This RF channel will either be in the set of *used* RF channels or in the set of *unused* RF channels.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 235

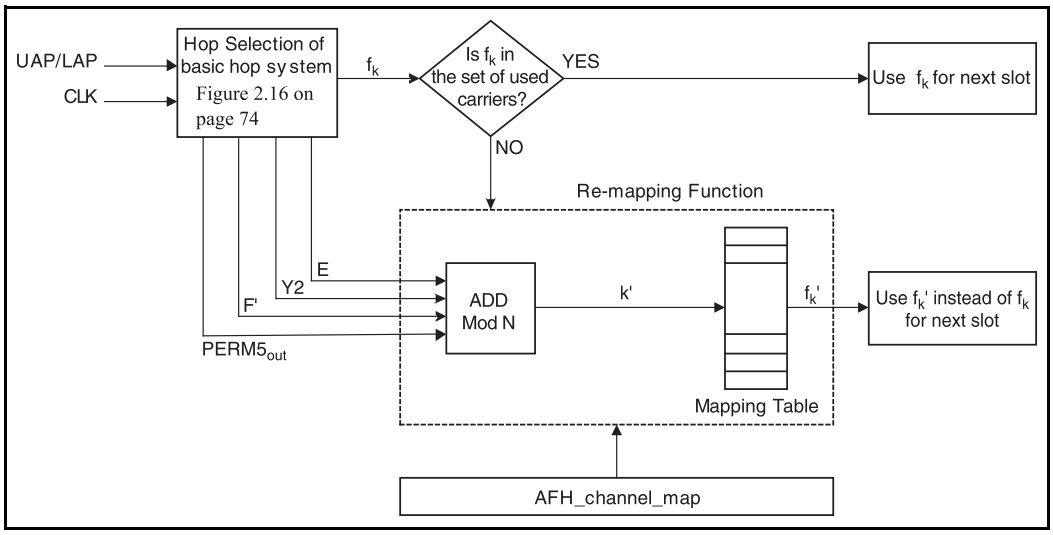


Figure 2.20: Block diagram of adaptive hop selection mechanism

When an unused RF channel is generated by the basic hop selection mechanism, it is re-mapped to the set of *used* RF channels as follows. A new index *k'* ∈ {0, 1, ..., *N*-1} is calculated using some of the parameters from the basic hop selection kernel:

$$k' = (PERM5_{out} + E + F' + Y2) \bmod N$$

where *F'* is defined in Table 2.2 on page 79. The index *k'* is then used to select the re-mapped channel from a mapping table that contains all of the even *used* RF channels in ascending order followed by all the odd *used* RF channels in ascending order (i.e., the mapping table of Figure 2.16 on page 74 with all the *unused* RF channels removed).

2.6.4 Control word

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

The control word of the kernel is controlled by the overall control signals X, Y1, Y2, A to F, and F' as illustrated in Figure 2.16 on page 74 and Figure 2.20 on page 78. During paging and inquiry, the inputs A to E use the address values as given in the corresponding columns of Table 2.2 on page 79. In addition, the inputs X, Y1 and Y2 are used. The F and F' inputs are unused. The clock bits CLK_{6-2} (i.e., input X) specifies the phase within the length 32 sequence. CLK_1 (i.e., inputs Y1 and Y2) is used to select between TX and RX. The address inputs determine the sequence order within segments. The final mapping onto the hop frequencies is determined by the register contents.

During the **CONNECTION** state (see Section 8.5 on page 148), the inputs A, C and D shall be derived from the address bits being bit-wise XORed with the

Petitioner Kiosoft Exhibit 1011 Part 1
Page 236

clock bits as shown in the “Connection state” column of Table 2.2 on page 79 (the two most significant bits, MSBs, are XORed together, the two second MSBs are XORed together, etc.).

	Page scan / Interlaced Page Scan / Inquiry scan / Interlaced Inquiry Scan	**Page/Inquiry**	**Master/Slave page response and Inquiry response**	**Connection state**
X	CLKN_{16-12} / $(\mathrm{CLKN}_{16-12}+16) mod 32$ / Xir_{4-0} / $Xir_{4-0}+16) mod 32$	Xp_{4-0}/Xi_{4-0}	$Xprm_{4-0}$ / $Xprs_{4-0}$ / Xir_{4-0}	CLK_{6-2}
Y1	0	$\mathrm{CLKE}_1/\mathrm{CLKN}_1$	$\mathrm{CLKE}_1/\mathrm{CLKN}_1/1$	CLK_1
Y2	0	$32\times\mathrm{CLKE}_1/$ $32\times\mathrm{CLKN}_1$	$32\times\mathrm{CLKE}_1$ / $32\times\mathrm{CLKN}_1$ / 32×1	$32\times\mathrm{CLK}_1$
A	A_{27-23}	A_{27-23}	A_{27-23}	$A_{27-23}\oplus\mathrm{CLK}_{25-21}$
B	A_{22-19}	A_{22-19}	A_{22-19}	A_{22-19}
C	$A_{8,6,4,2,0}$	$A_{8,6,4,2,0}$	$A_{8,6,4,2,0}$	$A_{8,6,4,2,0}\oplus\mathrm{CLK}_{20-16}$
D	A_{18-10}	A_{18-10}	A_{18-10}	$A_{18-10}\oplus\mathrm{CLK}_{15-7}$
E	$A_{13,11,9,7,5,3,1}$	$A_{13,11,9,7,5,3,1}$	$A_{13,11,9,7,5,3,1}$	$A_{13,11,9,7,5,3,1}$
F	0	0	0	$16\times\mathrm{CLK}_{27-7} \bmod 79$
F’	n/a	n/a	n/a	$16\times\mathrm{CLK}_{27-7} \bmod \mathrm{N}$

Table 2.2: Control for hop system.

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

2.6.4.1 Page scan and inquiry scan hopping sequences

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

2.6.4.2 Page hopping sequence

When the sequence selection input is set to page, the paging device shall start using the **A**-train, i.e., $\{f(k-8), \ldots, f(k), \ldots, f(k+7)\}$, where $f(k)$ is the source's estimate of the current receiver frequency in the paged device. The index k is a function of all the inputs in Figure 2.16. There are 32 possible paging frequencies within each 1.28 second interval. Half of these frequencies belong to the **A**-train, the rest (i.e., $\{f(k+8), \ldots, f(k+15), f(k-16), \ldots, f(k-9)\}$) belong to the **B**-train. In order to achieve the -8 offset of the **A**-train, a constant of 24 shall be added to the clock bits (which is equivalent to -8 due to the modulo 32 operation). The **B**-train is obtained by setting the offset to 8. A cyclic shift of the order within the trains is also necessary in order to avoid a possible repetitive mismatch between the paging and scanning devices. Thus,

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

where

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

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

2.6.4.3 Slave page response hopping sequence

When the sequence selection input is set to *slave page response,* in order to eliminate the possibility of losing the link due to discrepancies of the native clock CLKN and the master's clock estimate CLKE, the four bits CLKN_{16-12} shall be frozen at their current value. The value shall be frozen at the content it has in the slot where the recipient's access code is detected. The native clock shall *not* be stopped; it is merely the values of the bits used for creating the X-input that are kept fixed for a while. A frozen value is denoted by an asterisk (*) in the discussion below.

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

$$Xprs = [\mathrm{CLKN}^*_{16-12} + N] \mod 32, \qquad \text{(EQ 4)}$$

Petitioner Kiosoft Exhibit 1011 Part 1
Page 238

The counter N shall be set to zero in the slot where the slave acknowledges the page (see Figure 8.3 on page 140 and Figure 8.4 on page 140). Then, the value of N shall be increased by one each time CLKN_1 is set to zero, which corresponds to the start of a master TX slot. The X-input shall be constructed this way until the first **FHS** packet is received *and* the immediately following response packet has been transmitted. After this the slave shall enter the **CONNECTION** state using the parameters received in the **FHS** packet.

2.6.4.4 Master page response hopping sequence

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

$$Xprm = [\mathrm{CLKE}^*_{16-12} + k_{offset}{}^* + (\mathrm{CLKE}^*_{4-2,0} - \mathrm{CLKE}^*_{16-12}) \bmod 16 + N] \bmod 32, \quad \text{(EQ 5)}$$

The value of N shall be increased each time CLKE_1 is set to zero, which corresponds to the start of a master TX slot.

2.6.4.5 Inquiry hopping sequence

When the sequence selection input is set to *inquiry*, the X-input is similar to that used in the *page hopping sequence*. Since no particular device is addressed, the native clock CLKN of the inquirer shall be used. Moreover, which of the two train offsets to start with is of no real concern in this state. Consequently,

$$Xi = [\mathrm{CLKN}_{16-12} + k_{offset} + (\mathrm{CLKN}_{4-2,0} - \mathrm{CLKN}_{16-12}) \bmod 16] \bmod 32, \quad \text{(EQ 6)}$$

where k_{offset} is defined by (EQ 3) on page 80. The initial choice of the offset is arbitrary.

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 239

2.6.4.6 Inquiry response hopping sequence

The *inquiry response* hopping sequence is similar to the *slave page response* hopping sequence with respect to the X-input. The clock input shall not be frozen, thus the following equation apply:

$$Xir = [\mathrm{CLKN}_{16-12} + N] \mod 32, \quad \text{(EQ 7)}$$

Furthermore, the counter N is increased not on CLKN_1 basis, but rather after each **FHS** packet has been transmitted in response to the inquiry. There is no restriction on the initial value of N as it is independent of the corresponding value in the inquiring unit.

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

2.6.4.7 Basic and adapted channel hopping sequence

In the *basic* and *adapted channel hopping sequences*, the clock bits to use in the basic or adapted hopping sequence generation shall always be derived from the master clock, CLK. The address bits shall be derived from the Bluetooth device address of the master.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 240

3 PHYSICAL LINKS

A physical link represents a baseband connection between devices. A physical link is always associated with exactly one physical channel. Physical links have common properties that apply to all logical transports on the physical link.

The common properties of physical links are:

- Power control (see Link Manager Protocol Section 4.1.3 on page 211)
- Link supervision (see Section 3.1 on page 83 and Link Manager Protocol Section 4.1.6 on page 218)
- Encryption (see Security [Part H] Section 4 on page 763 and Link Manager Protocol [Part C] Section 4.2.5 on page 232)
- Channel quality-driven data rate change (see Link Manager Protocol Section 4.1.7 on page 219)
- Multi-slot packet control (see Link Manager Protocol Section 4.1.10 on page 223)

3.1 LINK SUPERVISION

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 to monitor the link on both the master and the slave side to avoid possible collisions when the logical transport address (see Section 4.2 on page 85) or parked member address (see Section 4.7.1 on page 93) is reassigned to another slave.

To be able to detect link loss, both the master and the slave shall use a link supervision timer, $T_{supervision}$. Upon reception of a valid packet header with one of the slave's addresses (see Section 4.2 on page 85) on the physical link, the timer shall be reset. If at any time in **CONNECTION** state, the timer reaches the *supervisionTO* value, the connection shall be considered disconnected. The same link supervision timer shall be used for SCO, eSCO, and ACL logical transports.

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 242

4 LOGICAL TRANSPORTS

4.1 GENERAL

Between master and slave(s), different types of logical transports may be established. Five logical transports have been defined:

- Synchronous Connection-Oriented (SCO) logical transport
- Extended Synchronous Connection-Oriented (eSCO) logical transport
- Asynchronous Connection-Oriented (ACL) logical transport
- Active Slave Broadcast (ASB) logical transport
- Parked Slave Broadcast (PSB) logical transport

The synchronous logical transports are point-to-point logical transports between a master and a single slave in the piconet. The synchronous logical transports typically support time-bounded information like voice or general synchronous data. The master maintains the synchronous logical transports by using reserved slots at regular intervals. In addition to the reserved slots the eSCO logical transport may have a retransmission window after the reserved slots.

The ACL logical transport is also a point-to-point logical transport between the master and a slave. In the slots not reserved for synchronous logical transport(s), the master can establish an ACL logical transport on a per-slot basis to any slave, including the slave(s) already engaged in a synchronous logical transport.

The ASB logical transport is used by a master to communicate with active slaves. The PSB logical transport is used by a master to communicate with parked slaves.

4.2 LOGICAL TRANSPORT ADDRESS (LT_ADDR)

Each slave active in a piconet is assigned a primary 3-bit logical transport address (LT_ADDR). The all-zero LT_ADDR is reserved for broadcast messages. The master does not have an LT_ADDR. A master's timing relative to the slaves distinguishes it from the slaves. A secondary LT_ADDR is assigned to the slave for each eSCO logical transport in use in the piconet. Only eSCO traffic (i.e. NULL, POLL, and one of the EV packet types as negotiated at eSCO logical transport setup) may be sent on these LT_ADDRs. ACL traffic (including LMP) shall always be sent on the primary LT_ADDR. A slave shall only accept packets with matching primary or secondary LT_ADDR and broadcast packets. The LT_ADDR is carried in the packet header (see Section 6.4 on page 103). The LT_ADDR shall only be valid for as long as a slave is in the active mode. As soon as it is disconnected or parked, the slave shall lose all of its LT_ADDRs.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 243

The primary LT_ADDR shall be assigned by the master to the slave when the slave is activated. This is either at connection establishment, at role switch, or when the slave is unparked. At connection establishment and at role switch, the primary LT_ADDR is carried in the **FHS** payload. When unparking, the primary LT_ADDR is carried in the unpark message.

4.3 SYNCHRONOUS LOGICAL TRANSPORTS

The first type of synchronous logical transport, the SCO logical transport is a symmetric, point-to-point link between the master and a specific slave. The SCO logical transport reserves slots and can therefore be considered as a circuit-switched connection between the master and the slave. The master may support up to three SCO links to the same slave or to different slaves. A slave may support up to three SCO links from the same master, or two SCO links if the links originate from different masters. SCO packets are never retransmitted.

The second type of synchronous logical transport, the eSCO logical transport, is a point-to-point logical transport between the master and a specific slave. eSCO logical transports may be symmetric or asymmetric. Similar to SCO, eSCO reserves slots and can therefore be considered a circuit-switched connection between the master and the slave. In addition to the reserved slots, eSCO supports a retransmission window immediately following the reserved slots. Together, the reserved slots and the retransmission window form the complete eSCO window.

4.4 ASYNCHRONOUS LOGICAL TRANSPORT

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 244

4.5 TRANSMIT/RECEIVE ROUTINES

This section describes the way to use the packets as defined in Section 6 on page 97 in order to support the traffic on the ACL, SCO and eSCO links. Both single-slave and multi-slave configurations are considered. In addition, the use of buffers for the TX and RX routines are described.

The TX and RX routines described in sections 4.5.1 and 4.5.2 are informative only.

4.5.1 TX Routine

The TX routine is carried out separately for each asynchronous and synchronous link. Figure 4.1 on page 87 shows the asynchronous and synchronous buffers as used in the TX routine. In this figure, only a single TX asynchronous buffer and a single TX synchronous buffer are shown. In the master, there is a separate TX asynchronous buffer for each slave. In addition there may be one or more TX synchronous buffers for each synchronous slave (different SCO or eSCO logical transports may either reuse the same TX synchronous buffer, or each have their own TX synchronous buffer). Each TX buffer consists of two FIFO registers: one **current** register which can be accessed and read by the Link Controller in order to compose the packets, and one **next** register that can be accessed by the Baseband Resource Manager to load new information. The positions of the switches S1 and S2 determine which register is current and which register is next; the switches are controlled by the Link Controller. The switches at the input and the output of the FIFO registers can never be connected to the same register simultaneously.

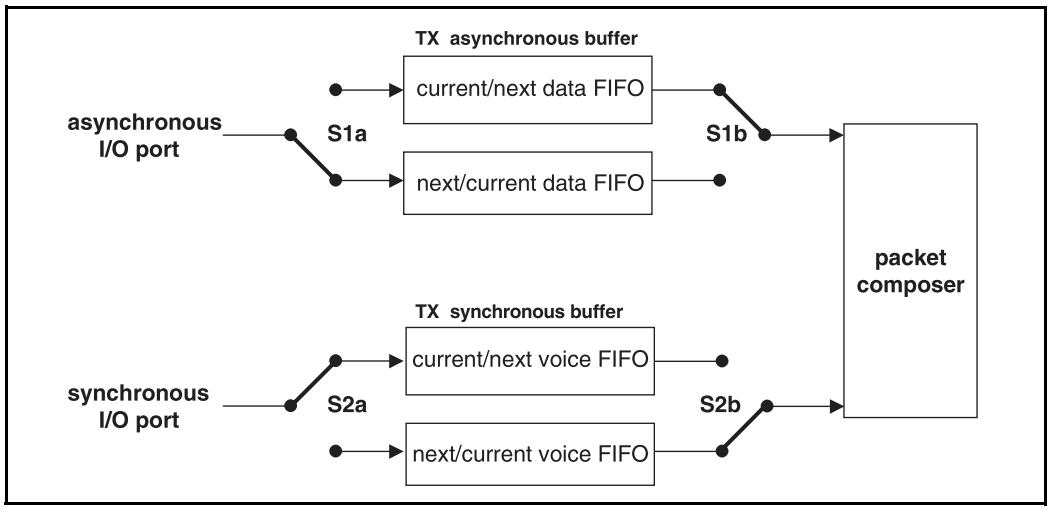


Figure 4.1: Functional diagram of TX buffering.

Of the packets common on the ACL and SCO logical transports (**NULL**, **POLL and DM1**) only the **DM1** packet carries a payload that is exchanged between the Link Controller and the Link Manager; this common packet makes use of the asynchronous buffer. All ACL packets make use of the asynchronous

buffer. All SCO and eSCO packets make use of the synchronous buffer except for the **DV** packet where the synchronous data part is handled by the synchronous buffer and the data part is handled by the asynchronous buffer. In the next sections, the operation for ACL traffic, SCO traffic, eSCO traffic, and combined data-voice traffic on the SCO logical transport are described.

4.5.1.1 ACL traffic

In the case of asynchronous data only the TX ACL buffer in Figure 4.1 on page 87 has to be considered. In this case, only packet types **DM** or **DH** are used, and these can have different lengths. The length is indicated in the payload header. The selection of **DM** or **DH** packets should depend on the quality of the link. See [Part C] Section 4.1.7 on page 219.

The default packet type in pure data traffic is **NULL** (see Section 6.5.1.2 on page 106). This means that, if there is no data to be sent (the data traffic is asynchronous, and therefore pauses occur in which no data is available) or no slaves need to be polled, **NULL** packets are sent instead – in order to send link control information to the other device (e.g. ACK/STOP information for received data). When no link control information is available (either no need to acknowledge and/or no need to stop the RX flow) no packet is sent at all.

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 246

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

4.5.1.2 SCO traffic

On the SCO logical transport only **HV** and **DV** packet types are used, See Section 6.5.2 on page 109. The synchronous port may continuously load the **next** register in the synchronous buffer. The S2 switches are changed according to the T_{sco} interval. This T_{sco} interval is negotiated between the master and the slave at the time the SCO logical transport is established.

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

4.5.1.3 Mixed data/voice traffic

In Section 6.5.2 on page 109, a **DV** packet has been defined that can support both data and voice simultaneously on a single SCO logical transport. When the TYPE is **DV**, the Link Controller reads the data register to fill the data field and the voice register to fill the voice field. Thereafter, the switch S2 is changed. However, the position of S1 depends on the result of the transmission as on the ACL logical transport: only if an ACK has been received will the S1 switch change its position. In each **DV** packet, the voice information is new, but the data information might be retransmitted if the previous transmission failed. If there is no data to be sent, the SCO logical transport will automatically change from **DV** packet type to the current **HV** packet type used before the mixed data/voice transmission. Note that a **flush** command is necessary when the data stream has been interrupted and new data has arrived.

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 247

4.5.1.4 eSCO Traffic

On the eSCO logical transport only **EV**, **POLL** and **NULL** packet types are used, see Section 6.5.3 on page 110. The synchronous port may continuously load the next register in the synchronous buffer. The S2 switches are changed according to the T_{eSCO} interval. This T_{eSCO} interval is negotiated between the master and the slave at the time the eSCO logical transport is established.

For each new eSCO slot, the packet composer reads the current register after which the S2 switch is changed. If the eSCO slot has to be used to send control information with high priority concerning a control packet between the master and the eSCO slave, or an ACL packet between the master and any other slave, the packet composer will discard the eSCO information and use the control information instead. Control information to the eSCO slave is sent in a DM1 packet on the primary LT_ADDR.

4.5.1.5 Default packet types

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

The SCO and eSCO packet types are negotiated at the LM level when the SCO or eSCO logical transport is established. The agreed packet type is also the default packet type for the reserved SCO or eSCO slots.

4.5.2 RX routine

The RX routine is carried out separately for the ACL logical transport and the synchronous logical transports. However, in contrast to the master TX asynchronous buffer, a single RX buffer is shared among all slaves. For the synchronous buffer, how the different synchronous logical transports are distinguished depends on whether extra synchronous buffers are required or not. Figure 4.2 on page 91 shows the asynchronous and synchronous buffers as used in the RX routine. The RX asynchronous buffer consists of two FIFO registers: one register that can be accessed and loaded by the Link Controller with the payload of the latest RX packet, and one register that can be accessed by the Baseband Resource Manager to read the previous payload. The RX synchronous buffer also consists of two FIFO registers: one register which is filled with newly arrived voice information, and one register which can be read by the voice processing unit.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 248

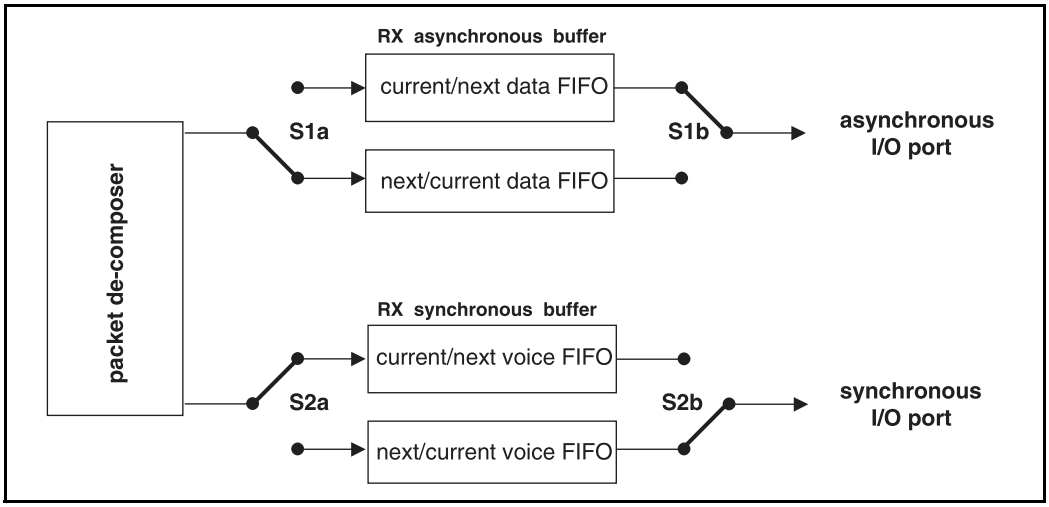


Figure 4.2: Functional diagram of RX buffering

Since the TYPE indication in the header (see Section 6.4.2 on page 103) of the received packet indicates whether the payload contains data and/or voice, the packet de-composer can automatically direct the traffic to the proper buffers. The switch S1 changes every time the Baseband Resource Manager reads the old register. If the next payload arrives before the RX register is emptied, a STOP indication is included in the packet header of the next TX packet that is returned. The STOP indication is removed again as soon as the RX register is emptied. The SEQN field is checked before a new ACL payload is stored into the asynchronous register (flush indication in LLID and broadcast messages influence the interpretation of the SEQN field see Section 7.6 on page 124).

The S2 switch is changed every T_{SCO} or T_{eSCO} for SCO and eSCO respectively. If, due to errors in the header, no new synchronous payload arrives, the switch still changes. The synchronous data processing unit then processes the synchronous data to account for the missing parts.

4.5.3 Flow control

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

4.5.3.1 Destination control

As long as data can not be received, a STOP indication shall be transmitted which is automatically inserted by the Link Controller into the header of the return packet. STOP shall be returned as long as the RX ACL buffer is not emptied by the Baseband Resource Manager. When new data can be accepted again, the GO indication shall be returned. GO shall be the default value. All packet types not including data can still be received. Voice communication for example is not affected by the flow control. Although a device can not receive new information, it may still continue to transmit information: the flow control shall be separate for each direction.

4.5.3.2 Source control

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

In a multi-slave configuration, only the transmission to the slave that issued the STOP signal shall be stalled. This means that the master shall only stop transmission from the TX ACL buffer corresponding to the slave that momentarily cannot accept data.

4.6 ACTIVE SLAVE BROADCAST TRANSPORT

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 piconet 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 logical transport may only be used for L2CAP group traffic and shall never be used for L2CAP connection-oriented channels, L2CAP control signalling or LMP control signalling.

The ASB logical transport is unreliable. To improve reliability somewhat 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 the reserved, all-zero, LT_ADDR. Packets on the ASB logical transport may be sent by the master at any time.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 250

4.7 PARKED SLAVE BROADCAST TRANSPORT

The parked slave broadcast logical transport is used for communication from the master to the slaves that are parked. 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 signalling).

The PSB logical transport is identified by the reserved, all-zero, LT_ADDR.

4.7.1 Parked member address (PM_ADDR)

A slave in the **PARK** state can be identified by its BD_ADDR or by a dedicated parked member address (PM_ADDR). This latter address is an 8-bit member address that separates the parked slaves. The PM_ADDR shall only be valid as long as the slave is parked. When the slave is activated it shall be assigned an LT_ADDR but shall lose the PM_ADDR. The PM_ADDR is assigned to the slave by the master during the parking procedure (see [Part C] Section 4.5.2 on page 247).

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

4.7.2 Access request address (AR_ADDR)

The access request address (AR_ADDR) is used by the parked slave to determine the slave-to-master half slot in the access window where it is allowed to send access request messages, see also Section 8.9.6 on page 171. The AR_ADDR shall be assigned to the slave when it enters the **PARK** state and shall only be valid as long as the slave is parked. The AR_ADDR is not necessarily unique; i.e. different parked slaves may have the same AR_ADDR.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 251

Petitioner Kiosoft Exhibit 1011 Part 1
Page 252

5 LOGICAL LINKS

Five logical links are defined:

- Link Control (LC)
- ACL Control (ACL-C)
- User Asynchronous/Isochronous (ACL-U)
- User Synchronous (SCO-S)
- User Extended Synchronous (eSCO-S)

The control logical links LC and ACL-C are used at the link control level and link manager level, respectively. The ACL-U logical link is used to carry either asynchronous or isochronous user information. The SCO-S, and eSCO-S logical links are used to carry synchronous user information. The LC logical link is carried in the packet header, all other logical links are carried in the packet payload. The ACL-C and ACL-U logical links are indicated in the logical link ID, LLID, field in the payload header. The SCO-S and eSCO-S logical links are carried by the synchronous logical transports only; the ACL-U link is normally carried by the ACL logical transport; however, they may also be carried by the data in the DV packet on the SCO logical transport. The ACL-C link may be carried either by the SCO or the ACL logical transport.

5.1 LINK CONTROL LOGICAL LINK (LC)

The LC control logical link shall be mapped onto the packet header. This logical link carries low level link control information like ARQ, flow control, and payload characterization. The LC logical link is carried in every packet except in the **ID** packet which does not have packet header.

5.2 ACL CONTROL LOGICAL LINK (ACL-C)

The ACL-C logical link shall carry control information exchanged between the link managers of the master and the slave(s). The ACL-C logical link shall use DM1 packets. The ACL-C logical link is indicated by the LLID code 11 in the payload header.

5.3 USER ASYNCHRONOUS/ISOCHRONOUS LOGICAL LINK (ACL-U)

The ACL-U logical link shall carry L2CAP asynchronous and isochronous user data. These messages may be transmitted in one or more baseband packets. For fragmented messages, the start packet shall use an LLID code of 10 in the payload header. Remaining continuation packets shall use LLID code 01. If there is no fragmentation, all packets shall use the LLID start code 10.

5.3.1 Pausing the ACL-U logical link

When paused by LM, the Link Controller transmits the current packet with ACL-U information, if any, until an ACK is received or, optionally, until an explicit NACK is received. While the ACL-U logical link is paused, the Link Controller shall not transmit any packets with ACL-U logical link information.

If the ACL-U was paused after an ACK, the next sequence number shall be used on the next packet. If the ACL-U was paused after a NAK, the same sequence number shall be used on the next packet and the un-acknowledged packet shall be transmitted once the ACL-U logical link is un-paused.

When the ACL-U logical link is un-paused by LM, the Link Controller may resume transmitting packets with ACL-U information.

5.4 USER SYNCHRONOUS DATA LOGICAL LINK (SCO-S)

The SCO-S logical link carries transparent synchronous user data. This logical link is carried over the synchronous logical transport SCO.

5.5 USER EXTENDED SYNCHRONOUS DATA LOGICAL LINK (eSCO-S)

The eSCO-S logical link also carries transparent synchronous user data. This logical link is carried over the extended synchronous logical transport eSCO.

5.6 LOGICAL LINK PRIORITIES

The ACL-C logical link shall have a higher priority than the ACL-U logical link when scheduling traffic on the shared ACL logical transport, except in the case when retransmissions of unacknowledged ACL packets shall be given priority over traffic on the ACL-C logical link. The ACL-C logical link should also have priority over traffic on the SCO-S and eSCO-S logical links but opportunities for interleaving the logical links should be taken.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 254

6 PACKETS

Bluetooth devices shall use the packets as defined in the following sections.

6.1 GENERAL FORMAT

The general packet format is shown in Figure 6.1 on page 97. Each packet consists of 3 entities: the access code, the header, and the payload. In the figure, the number of bits per entity is indicated.

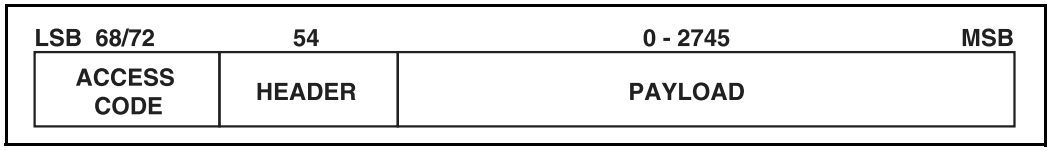


Figure 6.1: General packet format.

The access code is 72 or 68 bits and the header is 54 bits. The payload ranges from zero to a maximum of 2745 bits. Different packet types have been defined. Packet may consist of:

- the shortened access code only (see ID packet on page 116)
- the access code and the packet header
- the access code, the packet header and the payload.

6.2 BIT ORDERING

The bit ordering when defining packets and messages in the *Baseband 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 generated internally at baseband level, such as the packet header fields and payload header length, shall be transmitted with the LSB first. For instance, a 3-bit parameter X=3 is sent as:

$$b_0 b_1 b_2 = 110$$

over the air where 1 is sent first and 0 is sent last.

6.3 ACCESS CODE

Every packet starts with an access code. If a packet header follows, the access code is 72 bits long, otherwise the access code is 68 bits long and is known as a shortened access code. The shortened access code does not contain a trailer. This access code is used for synchronization, DC offset compensation and identification. The access code identifies all packets exchanged on a physical channel: all packets sent in the same physical channel are preceded by the same access code. In the receiver of the device, a sliding correlator correlates against the access code and triggers when a threshold is exceeded. This trigger signal is used to determine the receive timing.

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

The access code consists of a preamble, a sync word, and possibly a trailer, see Figure 6.2 on page 98. For details see Section 6.3.1 on page 98.

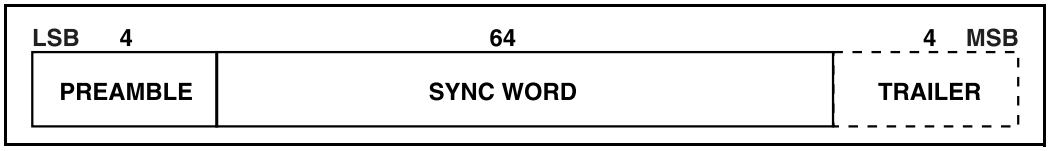


Figure 6.2: Access code format

6.3.1 Access code types

The different access code types use different Lower Address Parts (LAPs) to construct the sync word. The LAP field of the BD_ADDR is explained in Section 1.2 on page 55. A summary of the different access code types is in Table 6.1 on page 98.

Code type	LAP	Code length	Comments
CAC	Master	72	See also Section 1.3 on page 56
DAC	Paged device	68/72[1]	
GIAC	Reserved	68/72*	
DIAC	Dedicated	68/72*	

Table 6.1: Summary of access code types.

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

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

6.3.2 Preamble

The preamble is a fixed zero-one pattern of 4 symbols used to facilitate DC compensation. The sequence is either 1010 or 0101, depending on whether the LSB of the following sync word is 1 or 0, respectively. The preamble is shown in Figure 6.3 on page 99.



Figure 6.3: Preamble

6.3.3 Sync word

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

6.3.3.1 Synchronization word definition

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

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

The information sequence is generated by appending 6 bits to the 24 bit LAP (step 1). The appended bits are 001101 if the MSB of the LAP equals 0. If the MSB of the LAP is 1 the appended bits are 110010. The LAP MSB together with the appended bits constitute a length-seven Barker sequence. The purpose of including a Barker sequence is to further improve the auto correlation properties. In step 2 the information is pre-scrambled by XORing it with the bits $p_{34} \ldots p_{63}$ of the PN sequence (defined in section 6.3.3.2 on page 102). After generating the codeword (step 3), the complete PN sequence is XORed to the

Petitioner Kiosoft Exhibit 1011 Part 1
Page 257

codeword (step 4). This step de-scrambles the information part of the codeword. At the same time the parity bits of the codeword are scrambled. Consequently, the original LAP and Barker sequence are ensured a role as a part of the access code sync word, and the cyclic properties of the underlying code is removed. The principle is depicted in Figure 6.4 on page 100

In the following discussion, binary sequences will be denoted by their corresponding D-transform (in which D^i represents a delay of *i* time units). Let $p'(D) = p'_0 + p'_1 D + \ldots + p'_{62} D^{62}$ be the 63 bit PN sequence, where p'_0 is the first bit (LSB) leaving the PRNG (see Figure 6.5 on page 102), and, p'_{62} is the last bit (MSB). To obtain 64 bits, an extra zero is appended at the *end* of this sequence (thus, $p'(D)$ is unchanged). For notational convenience, the reciprocal of this extended polynomial, $p(D) = D^{63} p'(1/D)$, will be used in the following discussion. This is the sequence $p'(D)$ in reverse order. We denote the 24 bit lower address part (LAP) of the Bluetooth device address by $a(D) = a_0 + a_1 D + \ldots + a_{23} D^{23}$ (a_0 is the LSB of the Bluetooth device address).

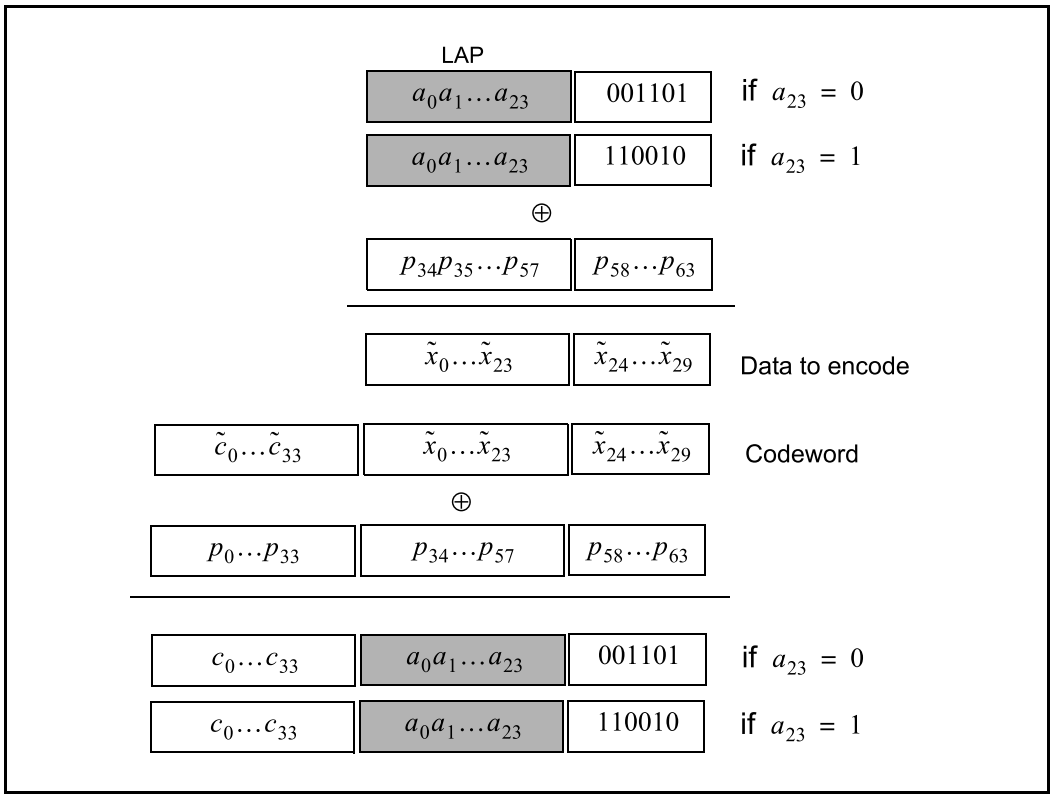


Figure 6.4: Construction of the sync word.

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 258

$$g(D) = 260534236651, \quad \text{(EQ 8)}$$

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

$$\begin{cases} F_0(D) = D + D^3, \\ F_1(D) = 1 + D^2, \end{cases} \quad \text{(EQ 9)}$$

respectively. Furthermore,

$$\begin{cases} B_0(D) = D^2 + D^3 + D^5, \\ B_1(D) = 1 + D + D^4, \end{cases} \quad \text{(EQ 10)}$$

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

1. Format the 30 information bits to encode:

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

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

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

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

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

4. Create the codeword:

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

5. Add the PN sequence:

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

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

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

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

6.3.3.2 Pseudo-random noise sequence generation

To generate the PN sequence the primitive polynomial $h(D) = 1 + D + D^3 + D^4 + D^6$ shall be used. The LFSR and its starting state are shown in Figure 6.5 on page 102. The PN sequence generated (including the extra terminating zero) becomes (hexadecimal notation) 83848D96BBCC54FC. The LFSR output starts with the left-most bit of this PN sequence. This corresponds to $p'(D)$ of the previous section. Thus, using the reciprocal $p(D)$ as overlay gives the 64 bit sequence:

$$\boldsymbol{p} = 3F2A33DD69B121C1, \quad \text{(EQ 11)}$$

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

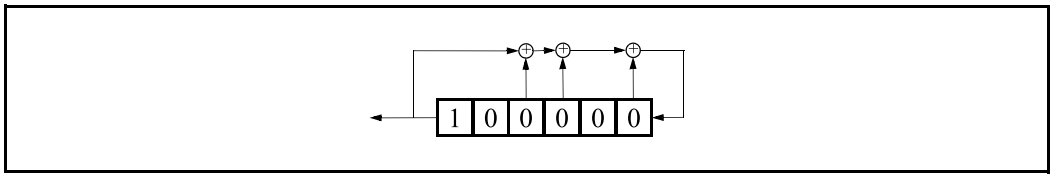


Figure 6.5: LFSR and the starting state to generate $p'(D)$.

6.3.4 Trailer

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

The trailer is a fixed zero-one pattern of four symbols. The trailer together with the three MSBs of the syncword form a 7-bit pattern of alternating ones and zeroes which may be used for extended DC compensation. The trailer sequence is either 1010 or 0101 depending on whether the MSB of the sync word is 0 or 1, respectively. The choice of trailer is illustrated in Figure 6.6 on page 102.



Figure 6.6: Trailer in CAC when MSB of sync word is 0 (a), and when MSB of sync word is 1 (b).

6.4 PACKET HEADER

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

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

The total header, including the HEC, consists of 18 bits, see Figure 6.7 on page 103, and is encoded with a rate 1/3 FEC (not shown but described in Section 7.4 on page 122) resulting in a 54-bit header. The LT_ADDR and TYPE fields shall be sent LSB first.

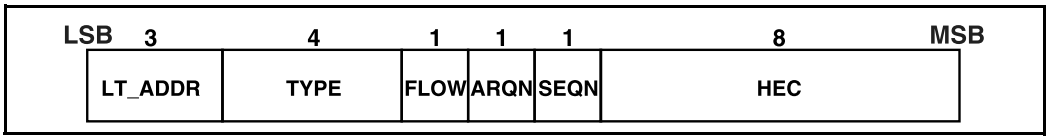


Figure 6.7: Header format.

6.4.1 LT_ADDR

The 3-bit LT_ADDR field contains the logical transport address for the packet (see Section 4.2 on page 85). This field indicates the destination slave for a packet in a master-to-slave transmission slot and indicates the source slave for a slave-to-master transmission slot.

6.4.2 TYPE

Sixteen different types of packets can be distinguished. The 4-bit TYPE code specifies which packet type is used. The interpretation of the TYPE code depends on the logical transport address in the packet. First, it shall be determined whether the packet is sent on an SCO logical transport, an eSCO logical transport, or an ACL logical transport. Then it can be determined which type of SCO packet, eSCO packet, or ACL packet has been received. The TYPE code determines how many slots the current packet will occupy (see the slot occupancy column in Table 6.2 on page 105). This allows the non-addressed receivers to refrain from listening to the channel for the duration of the remaining slots. In Section 6.5 on page 105, each packet type is described in more detail.

6.4.3 FLOW

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

The FLOW bit is not used on the eSCO logical transport or the ACL-C logical link and shall be set to one on transmission and ignored upon receipt.

6.4.4 ARQN

The 1-bit acknowledgment indication ARQN is used to inform the source of a successful transfer of payload data with CRC, and can be positive acknowledge ACK or negative acknowledge NAK. See Section 7.6 on page 124 for initialization and usage of this bit.

6.4.5 SEQN

The SEQN bit provides a sequential numbering scheme to order the data packet stream. See section 7.6.2 on page 127 for initialization and usage of the SEQN bit. For broadcast packets, a modified sequencing method is used, see Section 7.6.5 on page 130.

6.4.6 HEC

Each header has a header-error-check to check the header integrity. The HEC is an 8-bit word (generation of the HEC is specified in Section 7.1.1 on page 118). Before generating the HEC, the HEC generator is initialized with an 8-bit value. For FHS packets sent in **master response** substate, the slave upper address part (UAP) shall be used. For FHS packets sent in **inquiry response**, the default check initialization (DCI, see Section 1.2.1 on page 55) shall be used. In all other cases, the UAP of the master device shall be used.

After the initialization, a HEC shall be calculated for the 10 header bits. Before checking the HEC, the receiver shall initialize the HEC check circuitry with the proper 8-bit UAP (or DCI). If the HEC does not check, the entire packet shall be discarded. More information can be found in Section 7.1 on page 118.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 262

6.5 PACKET TYPES

The packets used on the piconet are related to the logical transports they are used in. Three logical transports with distinct packet types are defined (see Section 4 on page 85): the SCO logical transport, the eSCO logical transport, and the ACL logical transport. For each of these logical transports, 15 different packet types can be defined.

To indicate the different packets on a logical transport, the 4-bit TYPE code is used. The packet types are divided into four segments. The first segment is reserved for control packets. All control packets occupy a single time slot. The second segment is reserved for packets occupying a single time slot. The third segment is reserved for packets occupying three time slots. The fourth segment is reserved for packets occupying five time slots. The slot occupancy is reflected in the segmentation and can directly be derived from the type code. Table 6.2 on page 105 summarizes the packets defined for the SCO, eSCO, and ACL logical transport types.

Segment	TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO logical transport	eSCO logical transport	ACL logical transport
1	0000	1	NULL	NULL	NULL
	0001	1	POLL	POLL	POLL
	0010	1	FHS	reserved	FHS
	0011	1	DM1	reserved	DM1
2	0100	1	undefined	undefined	DH1
	0101	1	HV1	undefined	undefined
	0110	1	HV2	undefined	undefined
	0111	1	HV3	EV3	undefined
	1000	1	DV	undefined	undefined
	1001	1	undefined	undefined	AUX1
3	1010	3	undefined	undefined	DM3
	1011	3	undefined	undefined	DH3
	1100	3	undefined	EV4	undefined
	1101	3	undefined	EV5	undefined
4	1110	5	undefined	undefined	DM5
	1111	5	undefined	undefined	DH5

Table 6.2: Packets defined for synchronous and asynchronous logical transport types.

6.5.1 Common packet types

There are five common kinds of packets. In addition to the types listed in segment 1 of the previous table, the ID packet is also a common packet type but is not listed in segment 1 because it does not have a packet header.

6.5.1.1 ID packet

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

6.5.1.2 NULL packet

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

6.5.1.3 POLL packet

The POLL packet is very similar to the NULL packet. It does not have a payload. In contrast to the NULL packet, it requires a confirmation from the recipient. It is not a part of the ARQ scheme. The POLL packet does not affect the ARQN and SEQN fields. Upon reception of a POLL packet the slave shall respond with a packet even when the slave does not have any information to send unless the slave has scatternet commitments in that timeslot. This return packet is an implicit acknowledgement of the POLL packet. This packet can be used by the master in a piconet to poll the slaves. Slaves shall not transmit the POLL packet.

6.5.1.4 FHS packet

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

Figure 6.8 on page 107 illustrates the format and contents of the FHS payload. The payload consists of eleven fields. The FHS packet is used in page master response, inquiry response and in role switch.

The FHS packet contains real-time clock information. This clock information shall be updated before each retransmission. The retransmission of the FHS payload is different than retransmissions of ordinary data payloads where the same payload is used for each retransmission. The FHS packet is used for fre-

Petitioner Kiosoft Exhibit 1011 Part 1
Page 264

quency hop synchronization before the piconet channel has been established, or when an existing piconet changes to a new piconet.

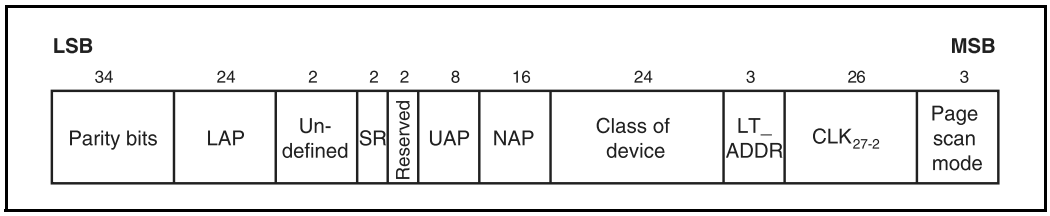


Figure 6.8: Format of the FHS payload.

Each field is described in more detail below:

Parity bits	This 34-bit field contains the parity bits that form the first part of the sync word of the access code of the device that sends the FHS packet. These bits are derived from the LAP as described in Section 1.2 on page 55.
LAP	This 24-bit field shall contain the lower address part of the device that sends the FHS packet.
Undefined	This 2-bit field is reserved for future use and shall be set to zero.
SR	This 2-bit field is the scan repetition field and indicates the interval between two consecutive page scan windows, see also Table 6.4 and Table 8.1 on page 135
Reserved	This 2-bit field shall be set to 10.
UAP	This 8-bit field shall contain the upper address part of the device that sends the FHS packet.
NAP	This 16-bit field shall contain the non-significant address part of the device that sends the FHS packet (see also Section 1.2 on page 55 for LAP, UAP, and NAP).
Class of device	This 24-bit field shall contain the class of device of the device that sends the FHS packet. The field is defined in Bluetooth Assigned Numbers (https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers).
LT_ADDR	This 3-bit field shall contain the logical transport address the recipient shall use if the FHS packet is used at connection setup or role switch. A slave responding to a master or a device responding to an inquiry request message shall include an all-zero LT_ADDR field if it sends the FHS packet.
CLK_{27-2}	This 26-bit field shall contain the value of the native clock of the device that sends the FHS packet, sampled at the beginning of the transmission of the access code of this FHS packet. This clock value has a resolution of 1.25ms (two-slot interval). For each new transmission, this field is updated so that it accurately reflects the real-time clock value.
Page scan mode	This 3-bit field shall indicate which scan mode is used by default by the sender of the FHS packet. The interpretation of the page scan mode is illustrated in Table 6.5.

Table 6.3: Description of the FHS payload

The device sending the FHS shall set the SR bits according to Table 6.4.

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

Table 6.4: Contents of SR field

The device sending the FHS shall set the page scan mode bits according to Table 6.5.

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

Table 6.5: Contents of page scan mode field

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

When initializing the HEC and CRC for the FHS packet of inquiry response, the UAP shall be the DCI.

6.5.1.5 DM1 packet

DM1 is part of segment 1 in order to support control messages in any logical transport that allows the DM1 packet (see Table 6.2 on page 105). However, it may also carry regular user data. Since the DM1 packet can be regarded as an ACL packet, it will be discussed in Section 6.5.4 on page 111.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 266

6.5.2 SCO packets

HV and DV packets are used on the synchronous SCO logical transport. The HV packets do not include a CRC and shall not be retransmitted. DV packets include a CRC on the data section, but not on the synchronous data section. The data section of DV packets shall be retransmitted. SCO packets may be routed to the synchronous I/O port. Four packets are allowed on the SCO logical transport: HV1, HV2, HV3 and DV. These packets are typically used for 64kb/s speech transmission but may be used for transparent synchronous data.

6.5.2.1 HV1 packet

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

6.5.2.2 HV2 packet

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

6.5.2.3 HV3 packet

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

6.5.2.4 DV packet

The **DV** packet is a combined data - voice packet.The DV packet shall only be used in place of an HV1 packet. The payload is divided into a voice field of 80 bits and a data field containing up to 150 bits, see Figure 6.9. The voice field is not protected by FEC. The data field has between 1 and 10 information bytes (including the 1-byte payload header) and includes a 16-bit CRC. The data field is encoded with a rate 2/3 FEC. Since the **DV** packet has to be sent at regular intervals due to its synchronous contents, it is listed under the SCO packet types. The voice and data fields shall be treated separately. The voice field shall be handled in the same way as normal SCO data and shall never be retransmitted; that is, the voice field is always new. The data field is checked for errors and shall be retransmitted if necessary. When the asynchronous data field in the DV packet has not be acknowledged before the SCO logical transport is terminated, the asynchronous data field shall be retransmitted in a DM1 packet.

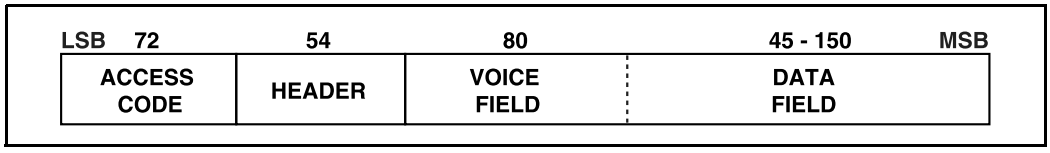


Figure 6.9: DV packet format

6.5.3 eSCO packets

EV packets are used on the synchronous eSCO logical transport. The packets include a CRC and retransmission may be applied if no acknowledgement of proper reception is received within the retransmission window. eSCO packets may be routed to the synchronous I/O port. Three eSCO packets have been defined. The eSCO packets may be used for 64kb/s speech transmission as well as transparent data at 64kb/s and other rates.

6.5.3.1 EV3 packet

The **EV3** packet has between 1 and 30 information bytes plus a 16-bit CRC code. The bytes are not protected by FEC. The EV3 packet may cover up to a single time slot. There is no payload header present. The payload length is set during the LMP eSCO setup and remains fixed until the link is removed or re-negotiated.

6.5.3.2 EV4 packet

The **EV4** packet has between 1 and 120 information bytes plus a 16-bit CRC code. The EV4 packet may cover to up three time slots. The information plus CRC bits are coded with a rate 2/3 FEC. There is no payload header present. The payload length is set during the LMP eSCO setup and remains fixed until the link is removed or re-negotiated.

6.5.3.3 EV5 packet

The **EV5** packet has between 1 and 180 information bytes plus a 16-bit CRC code. The bytes are not protected by FEC. The EV5 packet may cover up to three time slots. There is no payload header present. The payload length is set during the LMP eSCO setup and remains fixed until the link is removed or re-negotiated.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 268

6.5.4 ACL packets

ACL packets are used on the asynchronous logical transport. The information carried may be user data or control data.

6.5.4.1 DM1 packet

The DM1 packet carries data information only. The payload has between 1 and 18 information bytes (including the 1-byte payload header) plus a 16-bit CRC code. The DM1 packet occupies a single time slot. The information plus CRC bits are coded with a rate 2/3 FEC. The payload header in the DM1 packet is 1 byte long, see Figure 6.10 on page 113. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).

6.5.4.2 DH1 packet

This packet is similar to the DM1 packet, except that the information in the payload is not FEC encoded. As a result, the DH1 packet has between 1 and 28 information bytes (including the 1-byte payload header) plus a 16-bit CRC code. The DH1 packet occupies a single time slot.

6.5.4.3 DM3 packet

The DM3 packet may occupy up to three time slots. The payload has between 2 and 123 information bytes (including the 2-byte payload header) plus a 16-bit CRC code. The information plus CRC bits are coded with a rate 2/3 FEC. The payload header in the DM3 packet is 2 bytes long, see Figure 6.11 on page 113. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).

6.5.4.4 DH3 packet

This packet is similar to the DM3 packet, except that the information in the payload is not FEC encoded. As a result, the DH3 packet has between 2 and 185 information bytes (including the 2-byte payload header) plus a 16-bit CRC code. The DH3 packet may occupy up to three time slots.

6.5.4.5 DM5 packet

The DM5 packet may occupy up to five time slots. The payload has between 2 and 226 information bytes (including the 2-byte payload header) plus a 16-bit CRC code. The payload header in the DM5 packet is 2 bytes long. The information plus CRC bits are coded with a rate 2/3 FEC. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).

6.5.4.6 DH5 packet

This packet is similar to the DM5 packet, except that the information in the payload is not FEC encoded. As a result, the DH5 packet has between 2 and 341 information bytes (including the2-byte payload header) plus a 16-bit CRC code. The DH5 packet may occupy up to five time slots.

6.5.4.7 AUX1 packet

This packet resembles a DH1 packet but has no CRC code. The AUX1 packet has between 1 and 30 information bytes (including the 1-byte payload header). The AUX1 packet occupies a single time slot. The AUX1 packet shall not be used for the ACL-U or ACL-C logical links. An AUX1 packet may be discarded.

6.6 PAYLOAD FORMAT

In the payload, two fields are distinguished: the synchronous data field and the asynchronous data field. The ACL packets only have the asynchronous data field and the SCO and eSCOpackets only have the synchronous data field – with the exception of the DV packets which have both.

6.6.1 Synchronous data field

In SCO, the synchronous data field has a fixed length and consists only of the synchronous data body portion. No payload header is present.

In eSCO, the synchronous data field consists of two segments: a synchronous data body and a CRC code. No payload header is present.

1. **Synchronous data body**

   For HV and DV packets, the synchronous data body length is fixed. For EV packets, the synchronous data body length is negotiated during the LMP eSCO setup. Once negotiated, the synchronous data body length remains constant unless re-negotiated. The synchronous data body length may be different for each direction of the eSCO logical transport.

2. **CRC code**

   The 16-bit CRC in the payload is generated as specified in Section 7.1 on page 118. The 8-bit UAP of the master is used to initialize the CRC generator.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 270

6.6.2 Asynchronous data field

ACL packets have an asynchronous data field consisting of two or three segments: a payload header, a payload body, and possibly a CRC code (the AUX1 packet does not carry a CRC code).

1. Payload header

The payload header is one or two bytes long. Packets in segments one and two have a 1-byte payload header; packets in segments three and four have a 2-byte payload header. The payload header specifies the logical link (2-bit LLID indication), controls the flow on the logical channels (1-bit FLOW indication), and has a payload length indicator (5 bits and 9 bits for 1-byte and 2-byte payload headers, respectively). In the case of a 2-byte payload header, the length indicator is extended by four bits into the next byte. The remaining four bits of the second byte are reserved for future use and shall be set to zero. The formats of the 1-byte and 2-byte payload headers are shown in Figure 6.10 on page 113 and Figure 6.11 on page 113.

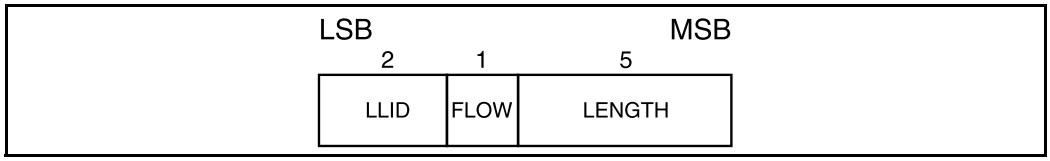


Figure 6.10: Payload header format for single-slot ACL packets.

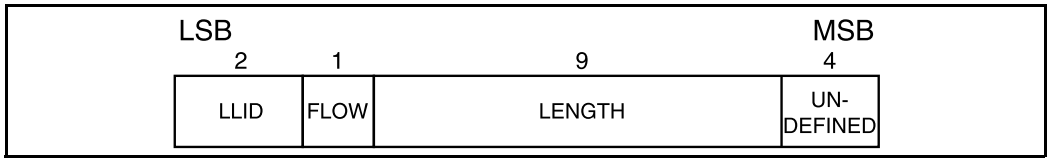


Figure 6.11: Payload header format for multi-slot ACL packets.

The LLID field shall be transmitted first, the length field last. In Table 6.6 on page 113, more details about the contents of the LLID field are listed.

LLID code b_1b_0	**Logical Link**	**Information**
00	NA	undefined
01	ACL-U	Continuation fragment of an L2CAP message
10	ACL-U	Start of an L2CAP message or no fragmentation
11	ACL-C	LMP message

Table 6.6: Logical link LLID field contents

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

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 272

The Baseband Resource Manager is responsible for setting and processing the flow bit in the payload header. Real-time flow control shall be carried out at the packet level by the link controller via the flow bit in the packet header (see Section 6.4.3 on page 104). With the payload flow bit, traffic from the remote end can be controlled. It is allowed to generate and send an ACL packet with payload length zero irrespective of flow status. L2CAP start-fragment and continue-fragment indications (LLID=10 and LLID=01) also retain their meaning when the payload length is equal to zero (i.e. an empty start-fragment shall not be sent in the middle of an on-going ACL-U packet transmission). It is always safe to send an ACL packet with length=0 and LLID=01. The payload flow bit has its own meaning for each logical link (ACL-U or ACL-C), Table 6.7 on page 115. On the ACL-C logical link, no flow control is applied and the payload FLOW bit shall always be set to one.

LLID code b_1b_0	**Usage and semantics of the ACL payload header FLOW bit**
00	Not defined, reserved for future use.
01 or 10	Flow control of the ACL-U channel (L2CAP messages)
11	Always set FLOW=1 on transmission and ignore the bit on reception

Table 6.7: Use of payload header flow bit on the logical links.

The length indicator shall be set to the number of bytes (i.e. 8-bit words) in the payload excluding the payload header and the CRC code; i.e. the payload body only. With reference to Figure 6.10 and Figure 6.11, the MSB of the length field in a 1-byte header is the last (right-most) bit in the payload header; the MSB of the length field in a 2-byte header is the fourth bit (from left) of the second byte in the payload header.

2. Payload body

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

3. CRC code generation

The 16-bit cyclic redundancy check code in the payload is generated as specified in Section 7.1 on page 118. Before determining the CRC code, an 8-bit value is used to initialize the CRC generator. For the CRC code in the FHS packets sent in **master response** substate, the UAP of the slave is used. For the FHS packet sent in **inquiry response** substate, the DCI (see Section 1.2.1 on page 55) is used. For all other packets, the UAP of the master is used.

6.7 PACKET SUMMARY

A summary of the packets and their characteristics is shown in Table 6.8, Table 6.9 and Table 6.10. The payload represents the packet payload excluding FEC, CRC, and payload header.

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

Table 6.8: Link control packets

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

Table 6.9: ACL packets

Type	Payload Header (bytes)	Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)
HV1	na	10	1/3	no	64.0
HV2	na	20	2/3	no	64.0
HV3	na	30	no	no	64.0
DV[1]	1 D	10+(0-9) D	2/3 D	yes D	64.0+57.6 D
EV3	na	1-30	No	Yes	96
EV4	na	1-120	2/3	Yes	192
EV5	na	1-180	No	Yes	288

Table 6.10: Synchronous packets

1. Items followed by 'D' relate to data field only.

7 BITSTREAM PROCESSING

Bluetooth devices shall use the bitstream processing schemes as defined in the following sections.

Before the payload is sent over the air interface, several bit manipulations are performed in the transmitter to increase reliability and security. An HEC is added to the packet header, the header bits are scrambled with a whitening word, and FEC coding is applied. In the receiver, the inverse processes are carried out. Figure 7.1 on page 117 shows the processes carried out for the packet header both at the transmit and the receive side. All header bit processes are mandatory.

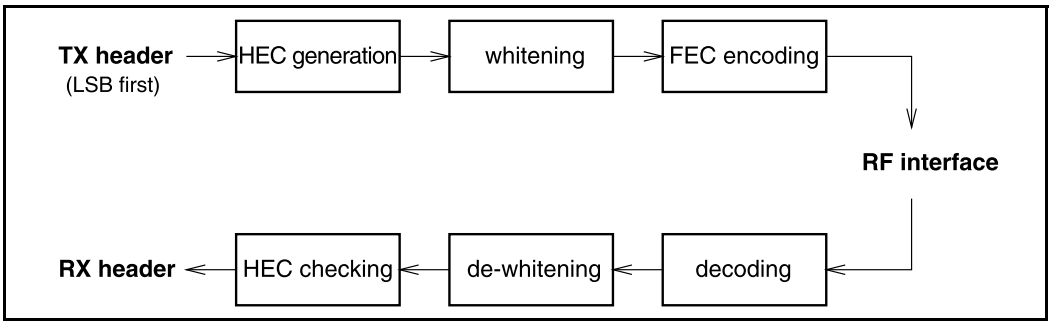


Figure 7.1: Header bit processes.

Figure 7.2 on page 117 shows the processes that may be carried out on the payload. In addition to the processes defined for the packet header, encryption can be applied on the payload. Only whitening and de-whitening, as explained in Section 7.2 on page 121, are mandatory for every payload; all other processes are optional and depend on the packet type (see Section 6.6 on page 112) and whether encryption is enabled. In Figure 7.2 on page 117, optional processes are indicated by dashed blocks.

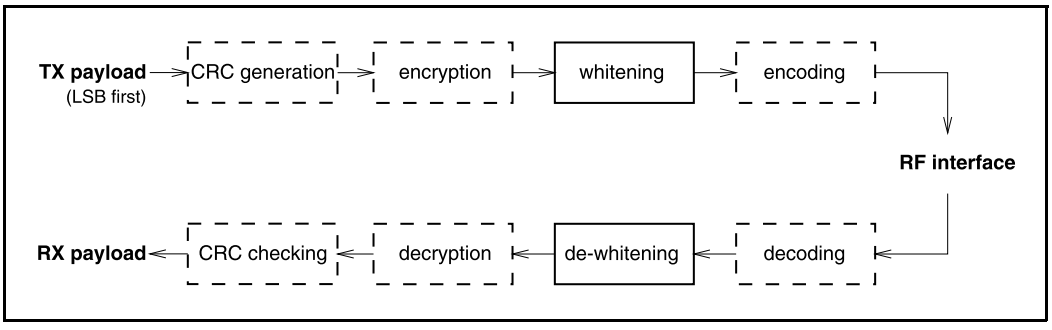


Figure 7.2: Payload bit processes.

7.1 ERROR CHECKING

The packet can be checked for errors or wrong delivery using the channel access code, the HEC in the header, and the CRC in the payload. At packet reception, the access code is checked first. Since the 64-bit sync word in the channel access code is derived from the 24-bit master LAP, this checks if the LAP is correct, and prevents the receiver from accepting a packet of another piconet (provided the LAP field of the master's BD_ADDR is different).

The HEC and CRC computations are normally initialized with the UAP of the master. Even though the access code may be the same for two piconets the different UAP values will typically cause the HEC and CRC to fail. However, there is an exception where no common UAP is available in the transmitter and receiver. This is the case when the HEC and CRC are generated for the FHS packet in **inquiry response** substate. In this case the DCI value shall be used.

The generation and check of the HEC and CRC are summarized in Figure 7.5 on page 119 and Figure 7.8 on page 120. Before calculating the HEC or CRC, the shift registers in the HEC/CRC generators shall be initialized with the 8-bit UAP (or DCI) value. Then the header and payload information shall be shifted into the HEC and CRC generators, respectively (with the LSB first).

7.1.1 HEC generation

The HEC generating LFSR is depicted in Figure 7.3 on page 118. The generator polynomial is $g(D) = (D+1)(D^7+D^4+D^3+D^2+1) = D^8+D^7+D^5+D^2+D+1$. Initially this circuit shall be pre-loaded with the 8-bit UAP such that the LSB of the UAP (denoted UAP_0) goes to the left-most shift register element, and, UAP_7 goes to the right-most element. The initial state of the HEC LFSR is depicted in Figure 7.4 on page 119. Then the data shall be shifted in with the switch S set in position 1. When the last data bit has been clocked into the LFSR, the switch S shall be set in position 2, and, the HEC can be read out from the register. The LFSR bits shall be read out from right to left (i.e., the bit in position 7 is the first to be transmitted, followed by the bit in position 6, etc.).

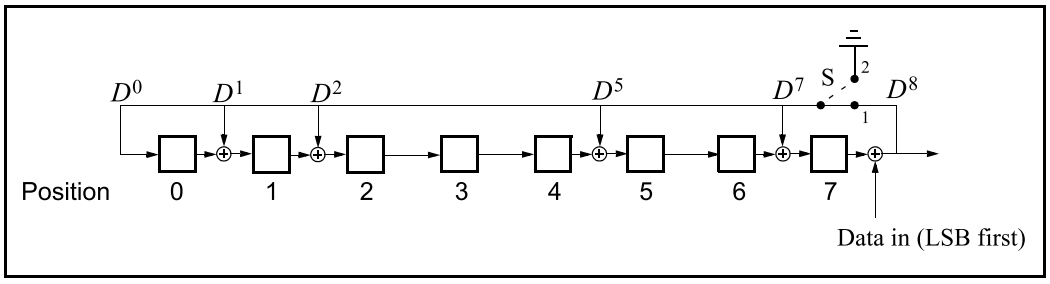


Figure 7.3: The LFSR circuit generating the HEC.

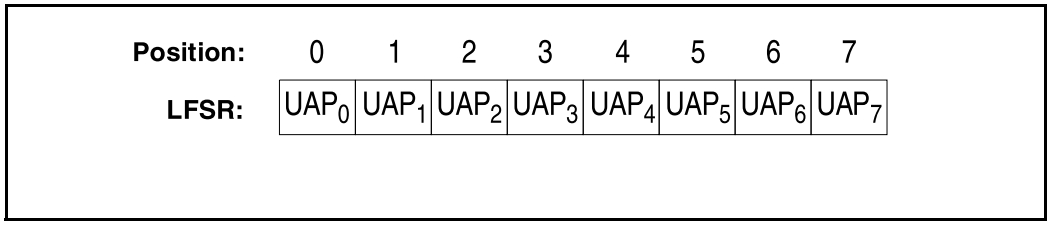


Figure 7.4: Initial state of the HEC generating circuit.

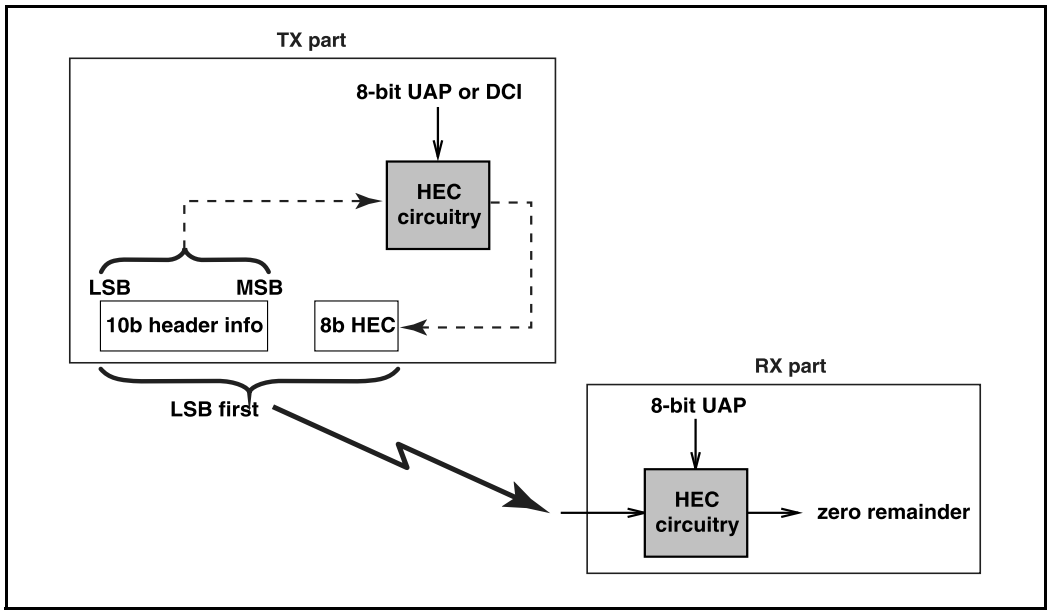


Figure 7.5: HEC generation and checking.

7.1.2 CRC generation

The 16 bit LFSR for the CRC is constructed similarly to the HEC using the CRC-CCITT generator polynomial $g(D) = D^{16} + D^{12} + D^5 + 1$ (i.e. 210041 in octal representation) (see Figure 7.6 on page 120). For this case, the 8 left-most bits shall be initially loaded with the 8-bit UAP (UAP_0 to the left and UAP_7 to the right) while the 8 right-most bits shall be reset to zero. The initial state of the 16 bit LFSR is specified in Figure 7.7 on page 120. The switch S shall be set in position 1 while the data is shifted in. After the last bit has entered the LFSR, the switch shall be set in position 2, and, the register's contents shall be transmitted, from right to left (i.e., starting with position 15, then position 14, etc.).

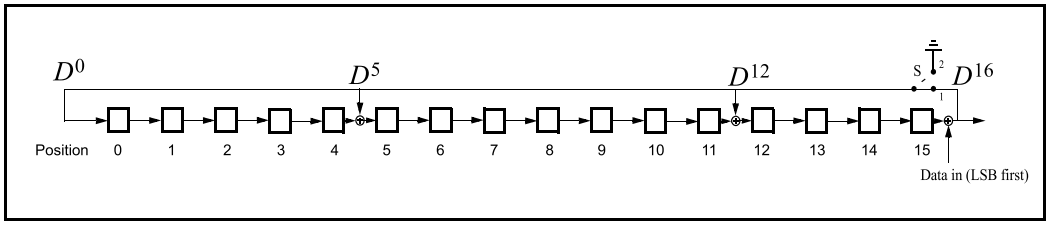


Figure 7.6: The LFSR circuit generating the CRC.

Position:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LFSR:	UAP_0	UAP_1	UAP_2	UAP_3	UAP_4	UAP_5	UAP_6	UAP_7	0	0	0	0	0	0	0	0

Figure 7.7: Initial state of the CRC generating circuit.

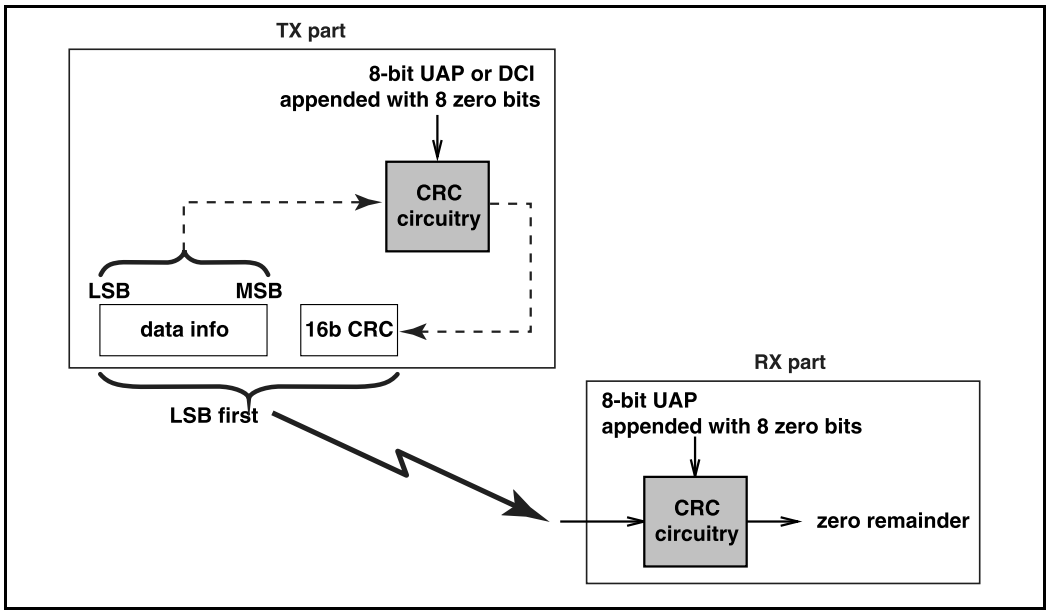


Figure 7.8: CRC generation and checking.

7.2 DATA WHITENING

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

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

The whitening word is generated with the polynomial $g(D) = D^7 + D^4 + 1$ (i.e., 221 in octal representation) and shall be subsequently XORed with the header and the payload. The whitening word is generated with the linear feedback shift register shown in Figure 7.9 on page 121. Before each transmission, the shift register shall be initialized with a portion of the master Bluetooth clock, CLK_{6-1}, extended with an MSB of value one. This initialization shall be carried out with CLK_1 written to position 0, CLK_2 written to position 1, etc. An exception is the FHS packet sent during page response or inquiry, where initialization of the whitening register shall be carried out differently. Instead of the master clock, the X-input used in the **inquiry** or **page response** (depending on current state) routine shall be used, see Table 2.2. The 5-bit value shall be extended with two MSBs of value 1. During register initialization, the LSB of X (i.e., X_0) shall be written to position 0, X_1 shall be written to position 1, etc.

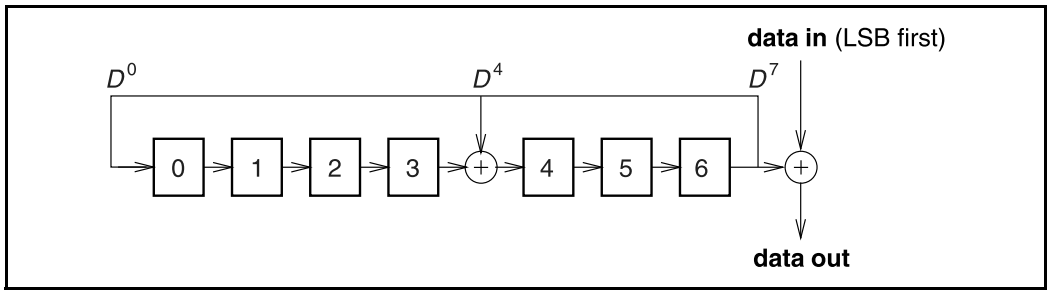


Figure 7.9: Data whitening LFSR.

After initialization, the packet header and the payload (including the CRC) are whitened. The payload whitening shall continue from the state the whitening LFSR had at the end of HEC. There shall be no re-initialization of the shift register between packet header and payload. The first bit of the “data in” sequence shall be the LSB of the packet header.

7.3 ERROR CORRECTION

There are three error correction schemes defined for Bluetooth:

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

The purpose of the FEC scheme on the data payload is to reduce the number of retransmissions. However, in a reasonable error-free environment, FEC gives unnecessary overhead that reduces the throughput. Therefore, the packet definitions given in Section 6 on page 97 have been kept flexible to use FEC in the payload or not, resulting in the **DM** and **DH** packets for the ACL logical transport, **HV** packets for the SCO logical transport, and **EV** packets for the eSCO logical transport. The packet header is always protected by a 1/3 rate FEC since it contains valuable link information and is designed to withstand more bit errors.

Correction measures to mask errors in the voice decoder are not included in this section. This matter is discussed in Section 9.3 on page 176.

7.4 FEC CODE: RATE 1/3

A simple 3-times repetition FEC code is used for the header. The repetition code is implemented by repeating each bit three times, see the illustration in Figure 7.10 on page 122. The 3-times repetition code is used for the entire header, as well as for the synchronous data field in the **HV1** packet.

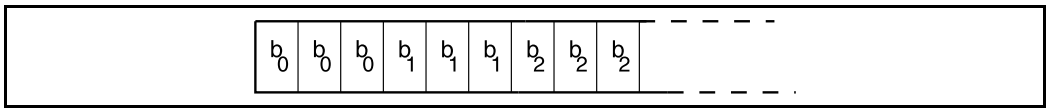


Figure 7.10: Bit-repetition encoding scheme.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 280

7.5 FEC CODE: RATE 2/3

The other FEC scheme is a (15,10) shortened Hamming code. The generator polynomial is $g(D) = (D+1)(D^4+D+1)$. This corresponds to 65 in octal notation. The LFSR generating this code is depicted in Figure 7.11 on page 123. Initially all register elements are set to zero. The 10 information bits are sequentially fed into the LFSR with the switches S1 and S2 set in position 1. Then, after the final input bit, the switches S1 and S2 are set in position 2, and the five parity bits are shifted out. The parity bits are appended to the information bits. Subsequently, each block of 10 information bits is encoded into a 15 bit codeword. This code can correct all single errors and detect all double errors in each codeword. This 2/3 rate FEC is used in the **DM** packets, in the data field of the **DV** packet, in the **FHS** packet, in the **HV2** packet, and in the **EV4** packet. Since the encoder operates with information segments of length 10, tail bits with value zero shall be appended after the CRC bits to bring the total number of bits equal to a multiple of 10. The number of tail bits to append shall be the least possible that achieves this (i.e., in the interval 0...9). These tail bits are not included in the payload length indicator for ACL packets or in the payload length field of the eSCO setup LMP command.

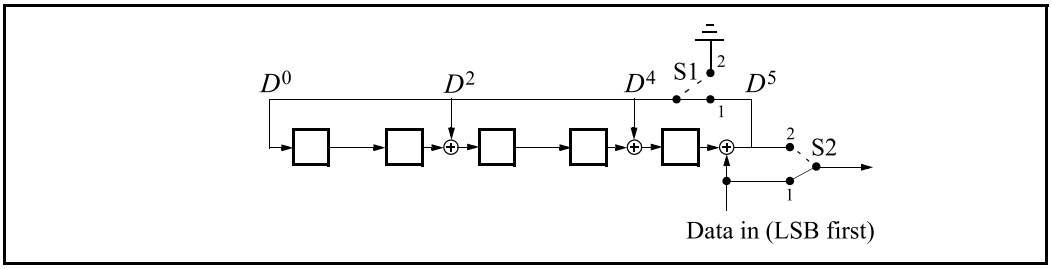


Figure 7.11: LFSR generating the (15,10) shortened Hamming code.

7.6 ARQ SCHEME

With an automatic repeat request scheme, **DM, DH** the data field of **DV** packets, and **EV** packets shall be transmitted until acknowledgement of a successful reception is returned by the destination (or timeout is exceeded). The acknowledgement information shall be included in the header of the return packet. The ARQ scheme is only used on the payload in the packet and only on packets that have a CRC. The packet header and the synchronous data payload of HV and DV packets are not protected by the ARQ scheme.

7.6.1 Unnumbered ARQ

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

In the first POLL packet at the start of a new connection (as a result of a page, page scan, role switch or unpark) the master shall initialize the ARQN bit to NAK. The response packet sent by the slave shall also have the ARQN bit set to NAK. The subsequent packets shall use the following rules. The initial value of the master's eSCO ARQN at link set-up shall be NAK.

The ARQ bit shall only be affected by data packets containing CRC and empty slots. As shown in Figure 7.12 on page 125, upon successful reception of a CRC packet, the ARQN bit shall be set to ACK. If, in any receive slot in the slave, or, in a receive slot in the master following transmission of a packet, one of these events applies:

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

then the ARQN bit shall be set to NAK. In eSCO the ARQN bit may be set to ACK even when the CRC on an EV packet has failed thus enabling delivery of erroneous packets.

Packets that have correct HEC but that are addressed to other slaves, or packets other than DH, DM, DV or EV packets, shall not affect the ARQN bit, except as noted in Section 7.6.2.2 on page 128. In these cases the ARQN bit shall be left as it was prior to reception of the packet. For ACL packets, if a CRC packet with a correct header has the same SEQN as the previously received CRC packet, the ARQN bit shall be set to ACK and the payload shall be ignored without checking the CRC. For eSCO packets, the SEQN shall not be used

Petitioner Kiosoft Exhibit 1011 Part 1
Page 282

when determining the ARQN. If an eSCO packet has been received successfully within the eSCO window subsequent receptions within the eSCO window shall be ignored. At the end of the eSCO window, the master's ARQN shall be retained for the first master-to-slave transmission in the next eSCO window.

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

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

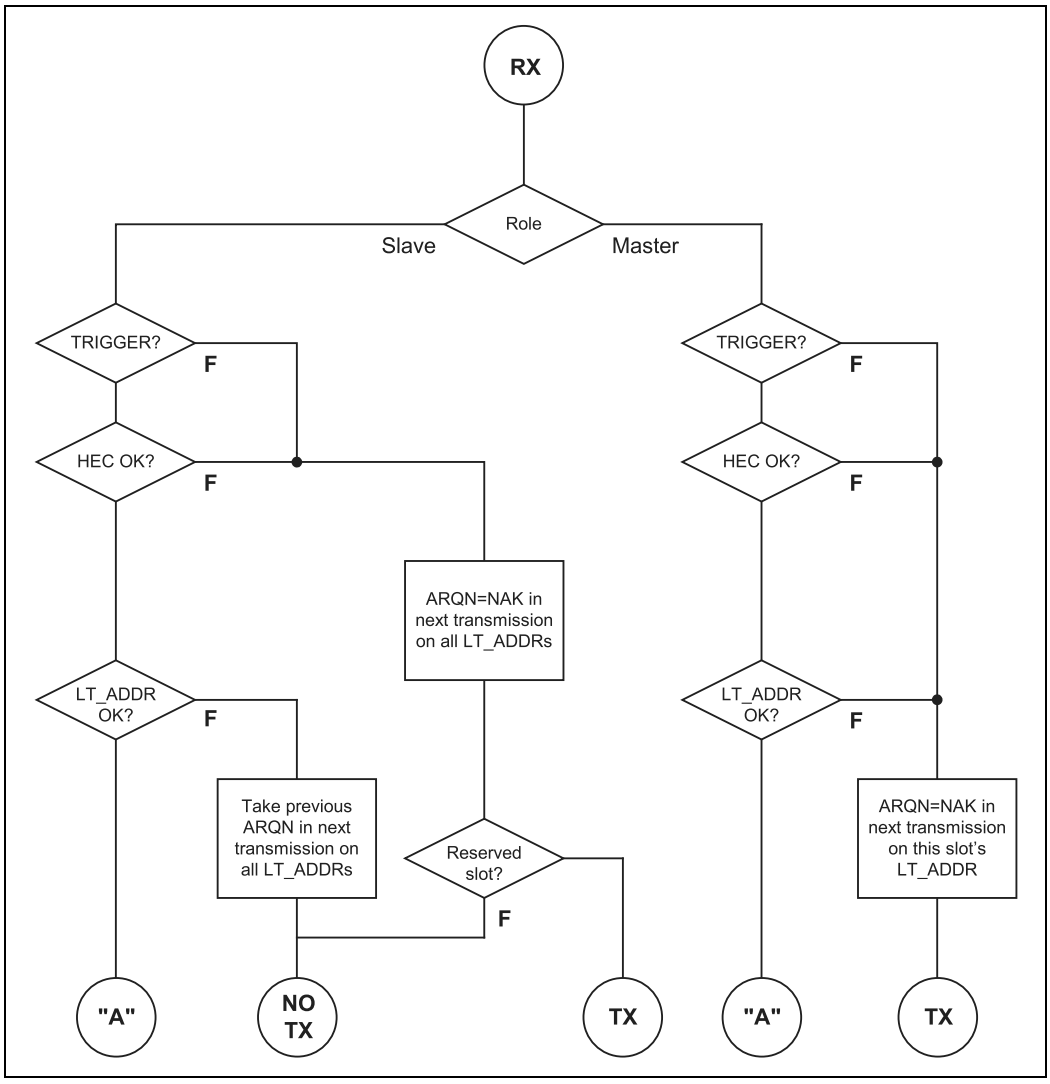


Figure 7.12: Stage 1 of the receive protocol for determining the ARQN bit.

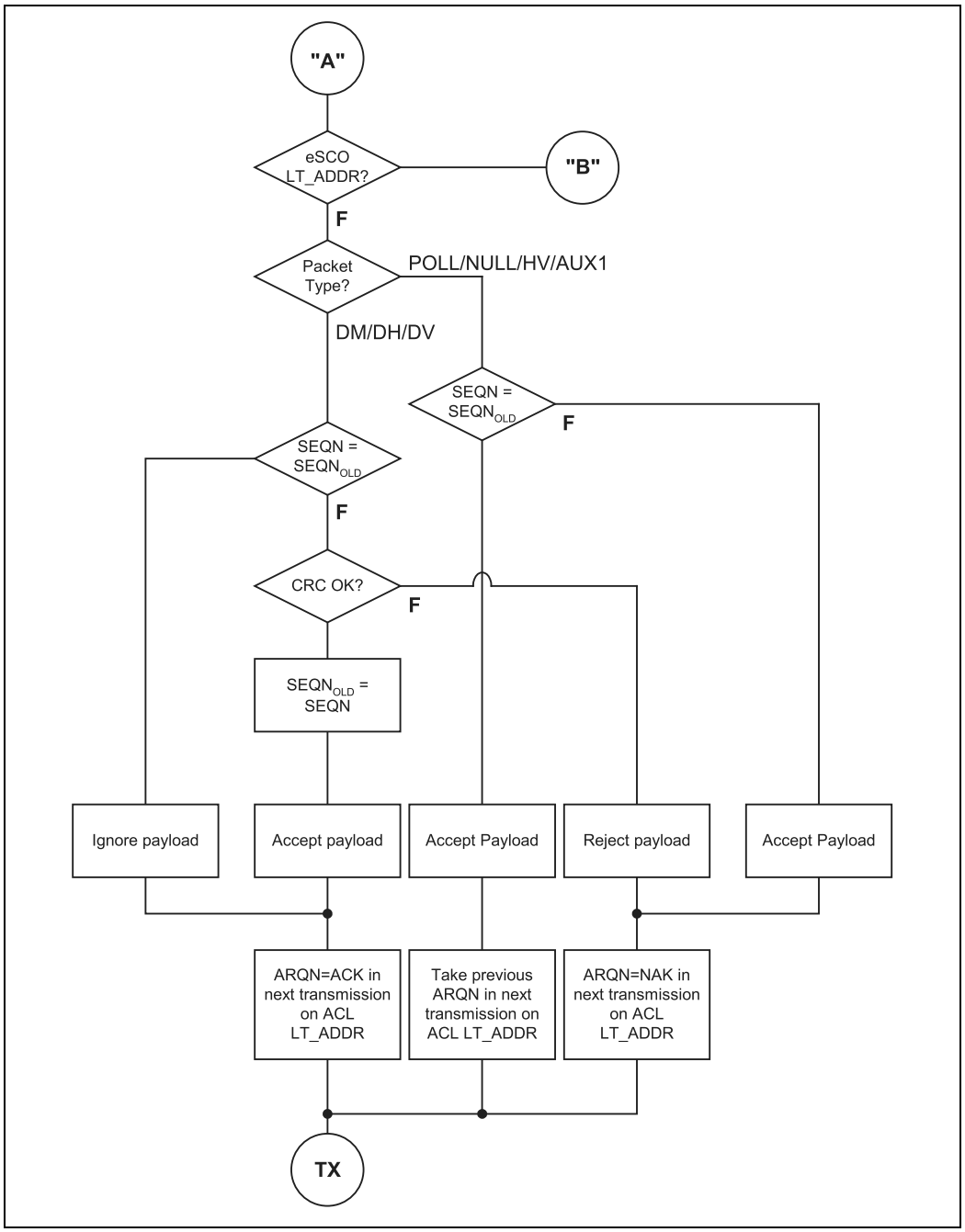


Figure 7.13: Stage 2 (ACL) of the receive protocol for determining the ARQN bit.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 284

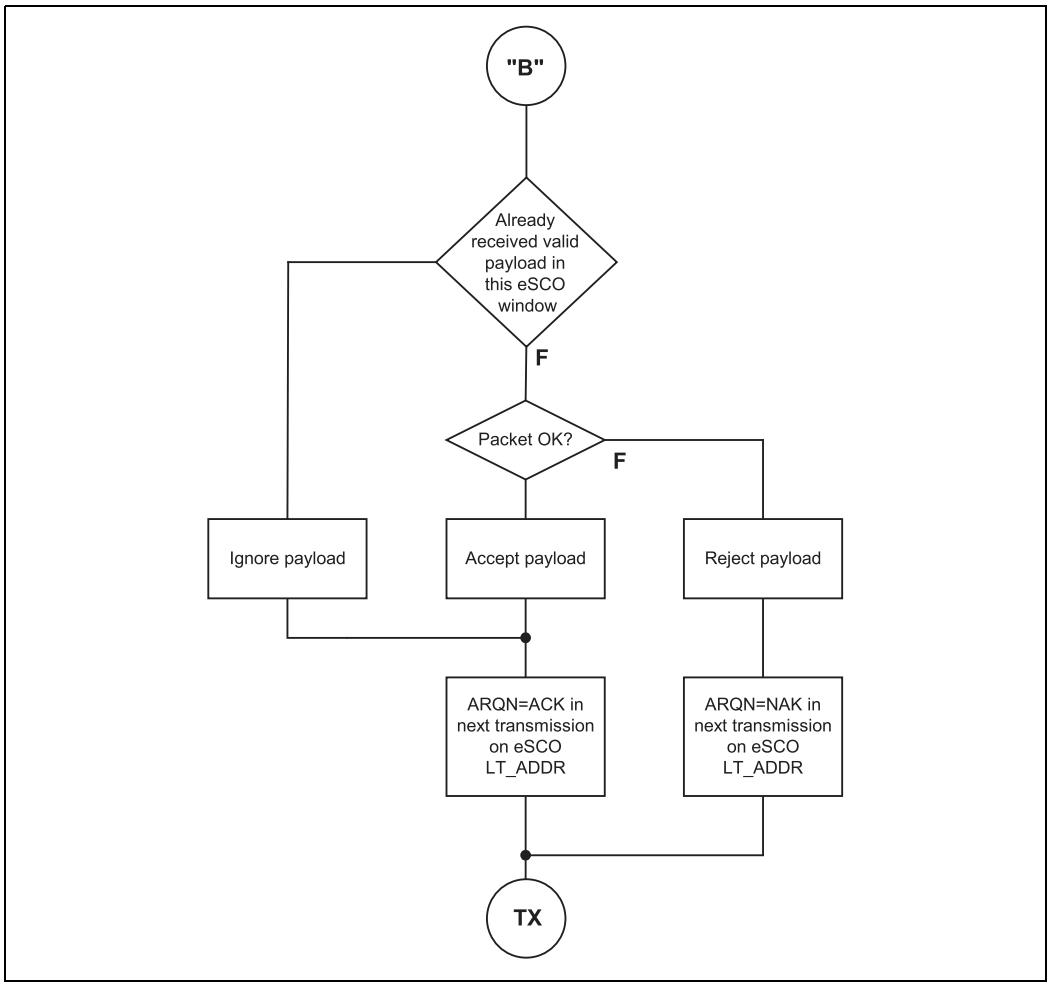


Figure 7.14: Stage 2 (eSCO) of the receive protocol for determining the ARQN bit.

7.6.2 Retransmit filtering

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

7.6.2.1 Initialization of SEQN at start of new connection

The SEQN bit of the first CRC data packet at the start of a connection (as a result of page, page scan, role switch or unpark) on both the master and the slave sides shall be set to 1. The subsequent packets shall use the rules in the following sections.

7.6.2.2 ACL and SCO retransmit filtering

The SEQN bit shall only be affected by the CRC data packets as shown in Figure 7.15. It shall be inverted every time a new CRC data packet is sent. The CRC data packet shall be retransmitted with the same SEQN number until an ACK is received or the packet is flushed. When an ACK is received, a new payload may be sent and on that transmission the SEQN bit shall be inverted. If a device decides to flush (see Section 7.6.3 on page 130), and it has not received an acknowledgement for the current packet, it shall replace the current packet with an ACL-U continuation packet with the same sequence number as the current packet and length zero. If it replaces the current packet in this way it shall not move on to transmit the next packet until it has received an ack.

If the slave receives a packet other than DH, DM, DV or EV with the SEQN bit inverted from that in the last header succesfully received on the same LT_ADDR, it shall set the ARQN bit to NAK until a DH, DM, DV or EV packet is successfully received.

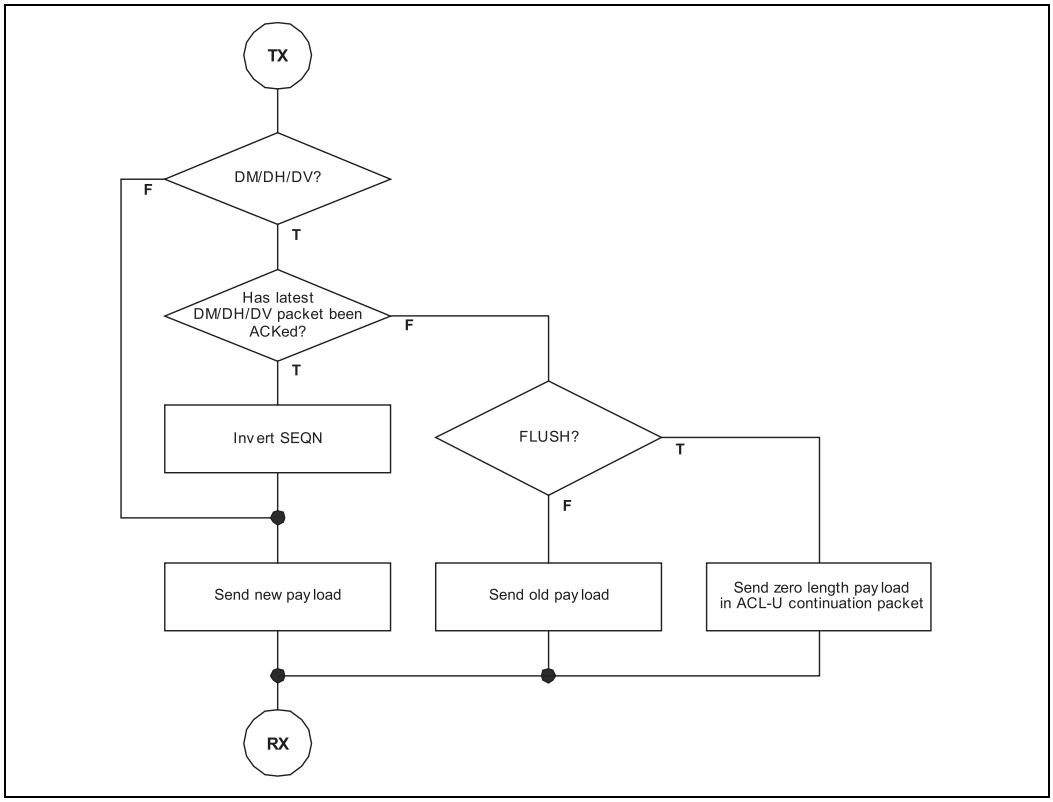


Figure 7.15: Transmit filtering for packets with CRC.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 286

7.6.2.3 eSCO retransmit filtering

In eSCO, the SEQN bit shall be toggled every eSCO window. The value shall be constant for the duration of the eSCO window. The initial value of SEQN shall be zero.

For a given eSCO window the SEQN value shall be constant.

7.6.2.4 FHS retransmit filtering

The SEQN bit in the FHS packet is not meaningful. This bit may be set to any value. Contents of the SEQN bit in the FHS packet shall not be checked.

7.6.2.5 Packets without CRC retransmit filtering

During transmission of packets without a CRC the SEQN bit shall remain the same as it was in the previous packet.

7.6.3 Flushing payloads

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

Flushing results in loss of remaining portions of an L2CAP message. Therefore, the packet following the flush shall have a start packet indication of LLID = 10 in the payload header for the next L2CAP message. This informs the destination of the flush. (see Section 6.6 on page 112). Flushing will not necessarily result in a change in the SEQN bit value, see the previous section.

The Flush Timeout defines a maximum period after which all segments of the ACL-U packet are flushed from the Controller buffer. The Flush Timeout shall start when the First segment of the ACL-U packet is stored in the Controller buffer. After the Flush timeout has expired the Link Controller may continue transmissions according to the procedure described in Section 7.6.2.2 on page 128, however the Baseband Resource Manager shall not continue the transmission of the ACL-U packet to the Link Controller. If the Baseband Resource Manager has further segments of the packet queued for transmission to the Link Controller it shall delete the remaining segments of the ACL-U packet from the queue. In case the complete ACL-U packet was not stored in the Controller buffer yet, any Continuation segments, received for the ACL logical transport, shall be flushed, until a First segment is received. When the complete ACL-U packet has been flushed, the Link Manager shall continue transmission of the next ACL-U packet for the ACL logical transport. The default Flush Timeout shall be infinite, i.e. re-transmissions are carried out until physical link loss occurs. This is also referred to as a 'reliable channel'. All devices shall support the default Flush Timeout.

In eSCO, packets shall be automatically flushed at the end of the eSCO window.

7.6.4 Multi-slave considerations

In a piconet with multiple logical transports, the master shall carry out the ARQ protocol independently on each logical transport.

7.6.5 Broadcast packets

Broadcast packets are packets transmitted by the master to all the slaves simultaneously. If multiple hop sequences are being used each transmission may only be received by some of the slaves. In this case the master shall repeat the transmission on each hop sequence. A broadcast packet shall be

Petitioner Kiosoft Exhibit 1011 Part 1
Page 288

indicated by the all-zero LT_ADDR (note; the FHS packet is the only packet which may have an all-zero LT_ADDR but is not a broadcast packet). Broadcast packets shall not be acknowledged (at least not at the LC level).

Since broadcast messages are not acknowledged, each broadcast packet is transmitted at least a fixed number of times. A broadcast packet should be transmitted N_{BC} times before the next broadcast packet of the same broadcast message is transmitted, see Figure 7.16 on page 131. Optionally, a broadcast packet may be transmitted N_{BC} + 1 times. Note: N_{BC}=1 means that each broadcast packet should be sent only once, but optionally may be sent twice. However, time-critical broadcast information may abort the ongoing broadcast train. For instance, unpark messages sent at beacon instances may do this, see Section 8.9.5 on page 171.

If multiple hop sequences are being used then the master may transmit on the different hop sequences in any order, providing that transmission of a new broadcast packet shall not be started until all transmissions of any previous broadcast packet have completed on all hop sequences. The transmission of a single broadcast packet may be interleaved among the hop sequences to minimize the total time to broadcast a packet. The master has the option of transmitting only N_{BC} times on channels common to all hop sequences.

Broadcast packets with a CRC shall have their own sequence number. The SEQN of the first broadcast packet with a CRC shall be set to SEQN = 1 by the master and shall be inverted for each new broadcast packet with CRC thereafter. Broadcast packets without a CRC have no influence on the sequence number. The slave shall accept the SEQN of the first broadcast packet it receives in a connection and shall check for change in SEQN for subsequent broadcast packets. Since there is no acknowledgement of broadcast messages and there is no end packet indication, it is important to receive the start packets correctly. To ensure this, repetitions of the broadcast packets that are L2CAP start packets and LMP packets shall not be filtered out. These packets shall be indicated by LLID=1X in the payload header as explained in section 6.6 on page 112. Only repetitions of the L2CAP continuation packets shall be filtered out.

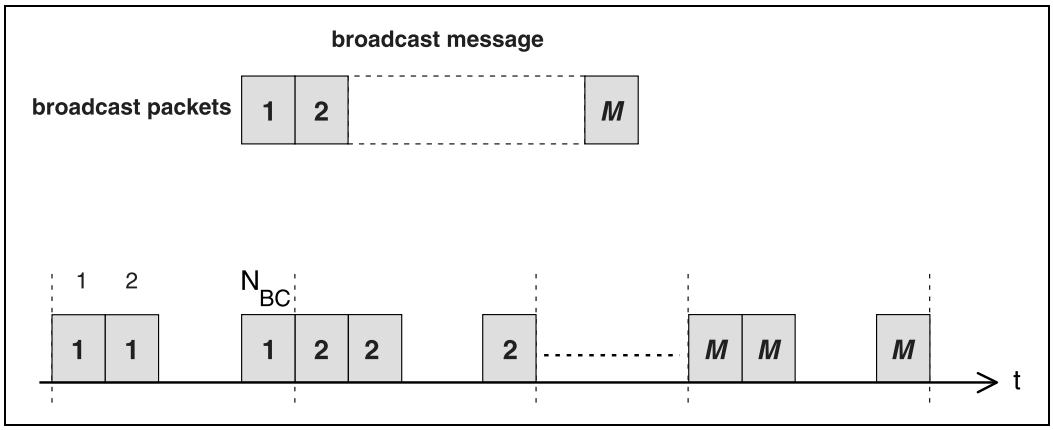


Figure 7.16: Broadcast repetition scheme

Petitioner Kiosoft Exhibit 1011 Part 1
Page 290

8 LINK CONTROLLER OPERATION

This section describes how a piconet is established and how devices can be added to and released from the piconet. Several states of operation of the devices are defined to support these functions. In addition, the operation of several piconets with one or more common members, the scatternet, is discussed.

8.1 OVERVIEW OF STATES

Figure 8.1 on page 133 shows a state diagram illustrating the different states used in the link controller. There are three major states: **STANDBY**, **CONNECTION**, and **PARK**; in addition, there are seven substates, **page**, **page scan**, **inquiry**, **inquiry scan**, **master response**, **slave response**, and **inquiry response**. The substates are interim states that are used to establish connections and enable device discovery. To move from one state or substate to another, either commands from the link manager are used, or internal signals in the link controller are used (such as the trigger signal from the correlator and the timeout signals).

.

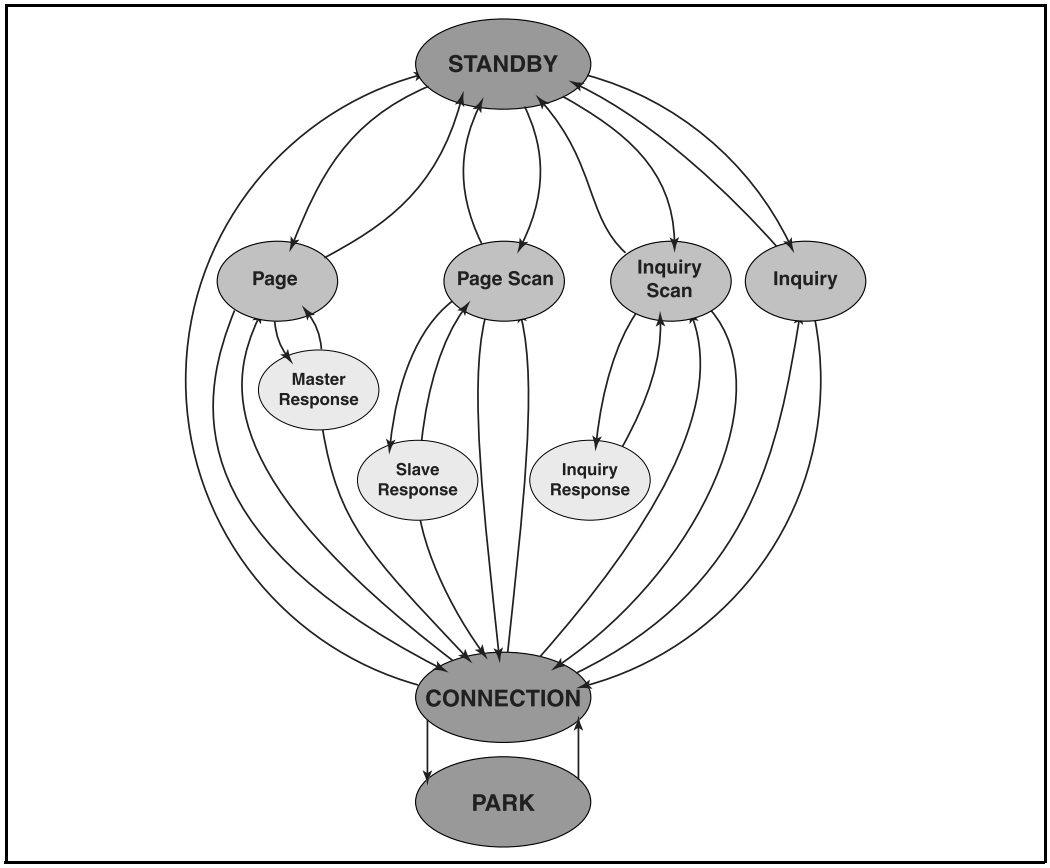


Figure 8.1: State diagram of link controller.

8.2 STANDBY STATE

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

The controller may leave the **STANDBY** state to scan for page or inquiry messages, or to page or inquiry itself.

8.3 CONNECTION ESTABLISHMENT SUBSTATES

In order to establish new connections the paging procedure is used. Only the Bluetooth device address is required to set up a connection. Knowledge about the clock, obtained from the inquiry procedure (see Section 8.4 on page 143) or from a previous connection with this device, and the page scanning mode of the other device will accelerate the setup procedure. A device that establishes a connection carries out a page procedure and will automatically become the master of the connection.

8.3.1 Page scan substate

In the **page scan** substate, a device may be configured to use either the standard or interlaced scanning procedure. During a standard scan, a device listens for the duration of the scan window $T_{w\_page\_scan}$ (11.25ms default, see HCI [Part E] Section 7.3.20 on page 467), while the interlaced scan is performed as two back to back scans of $T_{w\_page\_scan}$. If the scan interval is not at least twice the scan window, then interlaced scan shall not be used. During each scan window, the device shall listen at a single hop frequency, its correlator matched to its device access code (DAC). The scan window shall be long enough to completely scan 16 page frequencies.

When a device enters the **page scan** substate, it shall select the scan frequency according to the page hopping sequence determined by the device's Bluetooth device address, see Section 2.6.4.1 on page 79. The phase in the sequence shall be determined by $CLKN_{16-12}$ of the device's native clock; that is, every 1.28s a different frequency is selected.

In the case of a standard scan, if the correlator exceeds the trigger threshold during the **page scan**, the device shall enter the **slave response** substate described in Section 8.3.3.1 on page 140. The scanning device may also use interlaced scan. In this case, if the correlator does not exceed the trigger threshold during the first scan it shall scan a second time using the phase in the sequence determined by $[CLKN_{16-12} + 16] \bmod 32$. If on this second scan the correlator exceeds the trigger threshold the device shall enter the **slave response** substate using $[CLKN_{16-12} + 16] \bmod 32$ as the frozen CLKN* in the calculation for $Xprs^{(79)}$, see Section 2.6.4.3 on page 80 for details. If the correlator does not exceed the trigger threshold during a scan in normal mode or

during the second scan in interlaced scan mode it shall return to either the **STANDBY** or **CONNECTION** state.

The **page scan** substate can be entered from the **STANDBY** state or the **CONNECTION** state. In the **STANDBY** state, no connection has been established and the device can use all the capacity to carry out the **page scan**. Before entering the **page scan** substate from the **CONNECTION** state, the device should reserve as much capacity as possible for scanning. If desired, the device may place ACL connections in Hold, Park or Sniff, see Section 8.8 on page 165 and Section 8.9 on page 165. Synchronous connections should not be interrupted by the page scan, although eSCO retransmissions should be paused during the scan. The page scan may be interrupted by the reserved synchronous slots which should have higher priority than the **page scan**. SCO packets should be used requiring the least amount of capacity (**HV3** packets). The scan window shall be increased to minimize the setup delay. If one SCO logical transport is present using **HV3** packets and T_{SCO}=6 slots or one eSCO logical transport is present using **EV3** packets and T_{ESCO}=6 slots, a total scan window $T_{w\ page\ scan}$ of at least 36 slots (22.5ms) is recommended; if two SCO links are present using **HV3** packets and T_{SCO}=6 slots or two eSCO links are present using **EV3** packets and T_{ESCO}=6 slots, a total scan window of at least 54 slots (33.75ms) is recommended.

The scan interval $T_{page\ scan}$ is defined as the interval between the beginnings of two consecutive page scans. A distinction is made between the case where the scan interval is equal to the scan window $T_{w\ page\ scan}$ (continuous scan), the scan interval is maximal 1.28s, or the scan interval is maximal 2.56s. These three cases shall determine the behavior of the paging device; that is, whether the paging device shall use R0, R1 or R2, see also Section 8.3.2 on page 136. Table 8.1 illustrates the relationship between $T_{page\ scan}$ and modes R0, R1 and R2. Although scanning in the R0 mode is continuous, the scanning may be interrupted for example by reserved synchronous slots. The scan interval information is included in the SR field in the FHS packet.

SR mode	$T_{page\ scan}$
R0	≤ 1.28s and = $T_{w\ page\ scan}$
R1	≤ 1.28s
R2	≤ 2.56s
Reserved	-

Table 8.1: Relationship between scan interval, and paging modes R0, R1 and R2.

8.3.2 Page substate

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

The page procedure in the master consists of a number of steps. First, the Host communicates the BD_ADDR of the slave to the Controller. This BD_ADDR shall be used by the master to determine the page hopping sequence, see Section 2.6.4.2 on page 80. The slave's BD_ADDR shall be used to determine the page hopping sequence, see Section 2.6.4.2 on page 80. For the phase in the sequence, the master shall use an estimate of the slave's clock. For example, this estimate can be derived from timing information that was exchanged during the last encounter with this particular device (which could have acted as a master at that time), or from an inquiry procedure. With this estimate CLKE of the slave's Bluetooth clock, the master can predict on which hop channel the slave starts page scanning.

The estimate of the Bluetooth clock in the slave can be completely wrong. Although the master and the slave use the same hopping sequence, they use different phases in the sequence and might never select the same frequency. To compensate for the clock drifts, the master shall send its page message during a short time interval on a number of wake-up frequencies. It shall transmit also on hop frequencies just before and after the current, predicted hop frequency. During each TX slot, the master shall sequentially transmit on two different hop frequencies. In the following RX slot, the receiver shall listen sequentially to two corresponding RX hops for ID packet. The RX hops shall be selected according to the page response hopping sequence. The page response hopping sequence is strictly related to the page hopping sequence: for each page hop there is a corresponding page response hop. The RX/TX timing in the **page** substate is described in Section 2.2.5 on page 60, see also Figure 2.7 on page 65. In the next TX slot, it shall transmit on two hop frequencies different from the former ones. Note: The hop rate is increased to 3200 hops/s.

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

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 294

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

When the difference between the Bluetooth clocks of the master and the slave is less than -8x1.28 s or larger than +7x1.28 s, the remaining 16 hops are used to form the new 10 ms train **B.** The second train consists of hops

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

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

The **page** substate may be entered from the **STANDBY** state or the **CONNECTION** state. In the **STANDBY** state, no connection has been established and the device can use all the capacity to carry out the page. Before entering the **page** substate from the **CONNECTION** state, the device should free as much capacity as possible for scanning. To ensure this, it is recommended that the ACL connections are put on hold or park. However, the synchronous connections shall not be disturbed by the page. This means that the page will be interrupted by the reserved SCO and eSCO slots which have higher priority than the page. In order to obtain as much capacity for paging, it is recommended to use the SCO packets which use the least amount of capacity (**HV3** packets). If SCO or eSCO links are present, the repetition number N_{page} of a single train shall be increased, see Table 8.2. Here it has been assumed that the **HV3** packet are used with an interval T_{SCO}=6 slots or **EV3** packets are used with an interval of T_{ESCO}=6 slots, which would correspond to a 64 kb/s synchronous link.

.

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

Table 8.2: Relationship between train repetition, and paging modes R0, R1 and R2 when synchronous links are present.

The construction of the page train shall be independent of the presence of synchronous links; that is, synchronous packets are sent on the reserved slots but shall not affect the hop frequencies used in the unreserved slots, see Figure 8.2 on page 138.

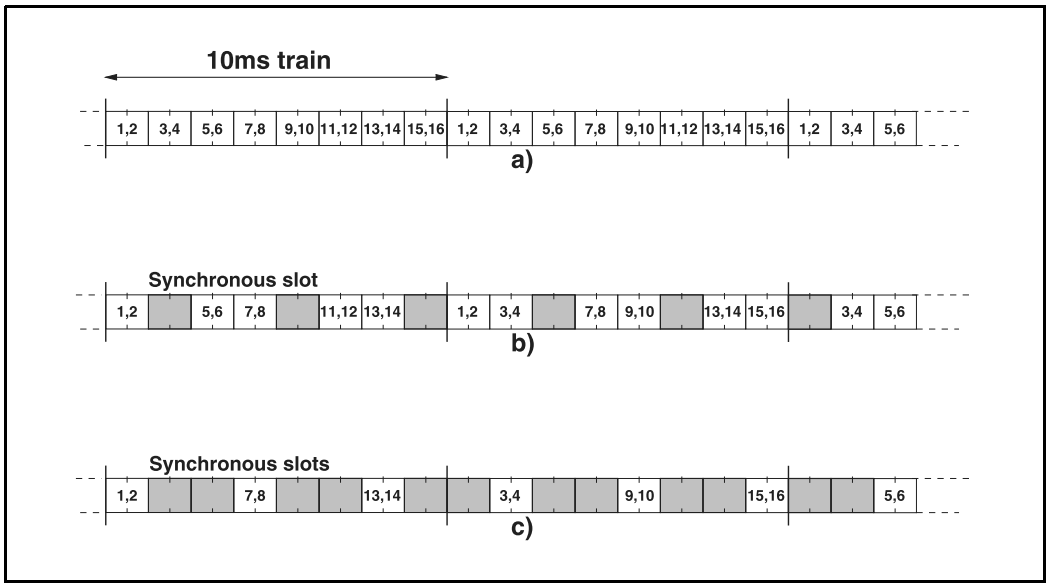


Figure 8.2: Conventional page (a), page while one synchronous link present (b), page while two synchronous links present (c).

8.3.3 Page response substates

When a page message is successfully received by the slave, there is a coarse FH synchronization between the master and the slave. Both the master and the slave enter a response substate to exchange vital information to continue the connection setup. It is important for the piconet connection that both devices shall use the same channel access code, use the same channel hopping sequence, and their clocks are synchronized. These parameters shall be derived from the master device. The device that initializes the connection (starts paging) is defined as the master device (which is thus only valid during the time the piconet exists). The channel access code and channel hopping sequence shall be derived from the Bluetooth device address (BD_ADDR) of the master. The timing shall be determined by the master clock. An offset shall be added to the slave's native clock to temporarily synchronize the slave clock to the master clock. At start-up, the master parameters are transmitted from the master to the slave. The messaging between the master and the slave at start-up is specified in this section.

The initial messaging between master and slave is shown in Table 8.3 on page 139 and in Figure 8.3 on page 140 and Figure 8.4 on page 140. In those two figures frequencies f (k), f(k+1), etc. are the frequencies of the page hopping sequence determined by the slave's BD_ADDR. The frequencies f'(k), f'(k+1), etc. are the corresponding page_response frequencies (slave-to-master). The frequencies g(m) belong to the basic channel hopping sequence.

Step	Message	Packet Type	Direction	Hopping Sequence	Access Code and Clock
1	Page	ID	Master to slave	Page	Slave
2	First slave page response	ID	Slave to master	Page response	Slave
3	Master page response	FHS	Master to slave	Page	Slave
4	Second slave page response	ID	Slave to master	Page response	Slave
5	1st packet master	POLL	Master to slave	Channel	Master
6	1st packet slave	Any type	Slave to master	Channel	Master

Table 8.3: Initial messaging during start-up.

In step 1 (see Table 8.3 on page 139), the master device is in **page** substate and the slave device in the **page scan** substate. Assume in this step that the page message sent by the master reaches the slave. On receiving the page message, the slave enters the **slave response** in step 2. The master waits for a reply from the slave and when this arrives in step 2, it will enter the **master response** in step 3. Note that during the initial message exchange, all parame-

ters are derived from the slave's device address, and that only the page hopping and page response hopping sequences are used (are also derived from the slave's device address). Note that when the master and slave enter the response states, their clock input to the page and page response hop selection is frozen as is described in Section 2.6.4.3 on page 80.

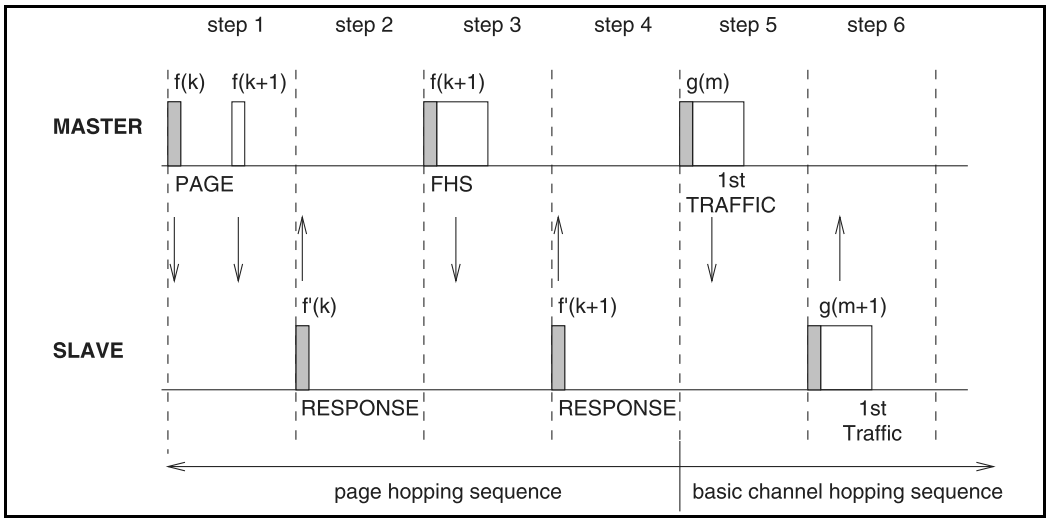


Figure 8.3: Messaging at initial connection when slave responds to first page message.

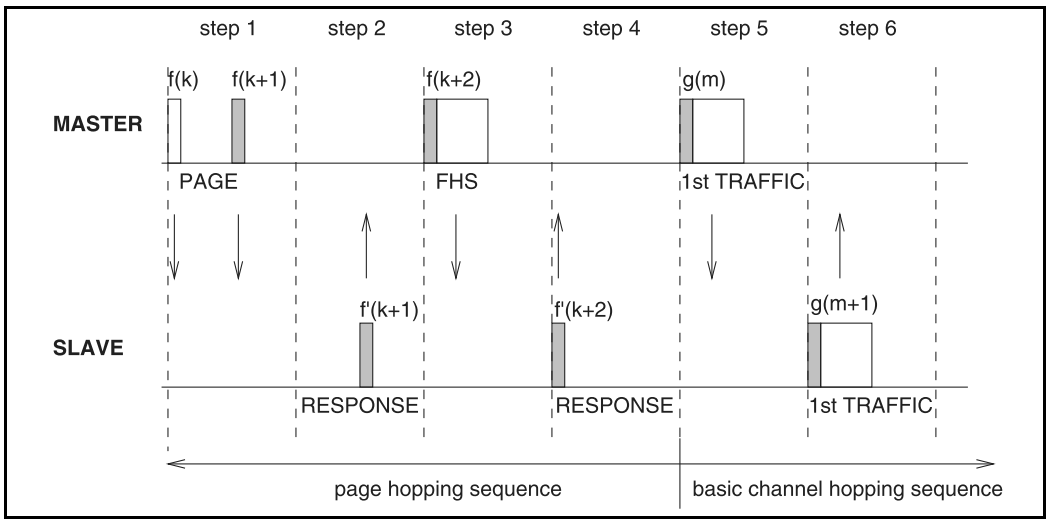


Figure 8.4: Messaging at initial connection when slave responds to second page message.

8.3.3.1 Slave response substate

After having received the page message in step 1, the slave device shall transmit a slave page response message (the slave's device access code) in step 2. This response message shall be the slave's device access code. The slave shall transmit this response 625 µs after the beginning of the received page message and at the response hop frequency that corresponds to the hop fre-

quency in which the page message was received. The slave transmission is therefore time aligned to the master transmission. During initial messaging, the slave shall still use the page response hopping sequence to return information to the master. The clock input $CLKN_{16-12}$ shall be frozen at the value it had at the time the page message was received.

After having sent the response message, the slave's receiver shall be activated 312.5 µs after the start of the response message and shall await the arrival of an **FHS** packet. Note that an **FHS** packet can arrive 312.5 µs after the arrival of the page message as shown in Figure 8.4 on page 140, and not after 625 µs as is usually the case in the piconet physical channel RX/TX timing. More details about the timing can be found in Section 2.4.4 on page 66.

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

If an **FHS** packet is received by the slave in the **slave response** substate, the slave shall return a slave page response message in step 4 to acknowledge reception of the **FHS** packet. This response shall use the page response hopping sequence. The transmission of the slave page response packet is based on the reception of the **FHS** packet. Then the slave shall change to the master's channel access code and clock as received from the **FHS** packet. Only the 26 MSBs of the master clock are transferred: the timing shall be such that CLK_1 and CLK_0 are both zero at the time the **FHS** packet was received as the master transmits in even slots only. The offset between the master's clock and the slave's clock shall be determined from the master's clock in the **FHS** packet and reported to the slave's Baseband Resource Manager.

Finally, the slave enters the **CONNECTION** state in step 5. From then on, the slave shall use the master's clock and the master's BD_ADDR to determine the basic channel hopping sequence and the channel access code. The slave shall use the LT_ADDR in the **FHS** payload as the primary LT_ADDR in the **CONNECTION** state. The connection mode shall start with a POLL packet transmitted by the master. The slave may respond with any type of packet. If the POLL packet is not received by the slave, or the response packet is not received by the master, within *newconnectionTO* number of slots after FHS packet acknowledgement, the master and the slave shall return to **page** and **page scan** substates, respectively. See Section 8.5 on page 148

8.3.3.2 Master response substate

When the master has received a slave page response message in step 2, it shall enter the **master response** routine. It shall freeze the current clock input to the page hop selection scheme. The master shall then transmit an **FHS** packet in step 3 containing the master's real-time Bluetooth clock, the master's BD_ADDR, the BCH parity bits, and the class of device. The **FHS** packet contains all information to construct the channel access code without requiring a mathematical derivation from the master's Bluetooth device address. The LT_ADDR field in the packet header of FHS packets in the master response substate shall be set to all-zeros. The **FHS** packet shall be transmitted at the beginning of the master-to-slave slot following the slot in which the slave responded. The FHS packet shall carry the all-zero LT_ADDR. The TX timing of the **FHS** is not based on the reception of the response packet from the slave. The **FHS** packet may therefore be sent 312.5 µs after the reception of the response packet like shown in Figure 8.4 on page 140 and not 625 µs after the received packet as is usual in the piconet physical channel RX/TX timing, see also Section 2.4.4 on page 66.

After the master has sent its **FHS** packet, it shall wait for a second slave page response message in step 4 acknowledging the reception of the **FHS** packet. This response shall be the slave's device access code. If no response is received, the master shall retransmit the **FHS** packet with an updated clock and still using the slave's parameters. It shall retransmit the FHS packet with the clock updated each time until a second slave page response message is received, or the timeout of *pagerespTO* is exceeded. In the latter case, the master shall return to the **page** substate and send an error message to the Baseband Resource Manager. During the retransmissions of the **FHS** packet, the master shall use the page hopping sequence.

If the slave's response is received, the master shall change to using the master parameters, so it shall use the channel access code and the master clock. The lower clock bits CLK_0 and CLK_1 shall be reset to zero at the start of the **FHS** packet transmission and are not included in the **FHS** packet. Finally, the master enters the **CONNECTION** state in step 5. The master BD_ADDR shall be used to change to a new hopping sequence, the *basic channel hopping sequence*. The basic channel hopping sequence uses all 79 hop channels in a pseudo-random fashion, see also Section 2.6.4.7 on page 82. The master shall now send its first traffic packet in a hop determined with the new (master) parameters. This first packet shall be a POLL packet. See Section 8.5 on page 148. This packet shall be sent within *newconnectionTO* number of slots after reception of the FHS packet acknowledgement. The slave may respond with any type of packet. If the POLL packet is not received by the slave or the POLL packet response is not received by the master within *newconnectionTO* number of slots, the master and the slave shall return to **page** and **page scan** substates, respectively.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 300

8.4 DEVICE DISCOVERY SUBSTATES

In order to discover other devices a device shall enter **inquiry** substate. In this substate, it shall repeatedly transmit the inquiry message (ID packet, see Section 6.5.1.1 on page 106) at different hop frequencies. The **inquiry** hop sequence is derived from the LAP of the GIAC. Thus, even when DIACs are used, the applied hopping sequence is generated from the GIAC LAP. A device that allows itself to be discovered, shall regularly enter the **inquiry scan** substate to respond to inquiry messages. The following sections describe the message exchange and contention resolution during inquiry response. The inquiry response is optional: a device is not forced to respond to an inquiry message.

During the **inquiry** substate, the discovering device collects the Bluetooth device addresses and clocks of all devices that respond to the inquiry message. It can then, if desired, make a connection to any one of them by means of the previously described page procedure.

The inquiry message broadcast by the source does not contain any information about the source. However, it may indicate which class of devices should respond. There is one general inquiry access code (GIAC) to inquire for any device, and a number of dedicated inquiry access codes (DIAC) that only inquire for a certain type of device. The inquiry access codes are derived from reserved Bluetooth device addresses and are further described in Section 6.3.1 on page 98.

8.4.1 Inquiry scan substate

The **inquiry scan** substate is very similar to the **page scan** substate. However, instead of scanning for the device's device access code, the receiver shall scan for the inquiry access code long enough to completely scan for 16 inquiry frequencies. Two types of scans are defined: standard and interlaced. In the case of a standard scan the length of this scan period is denoted $T_{w\_inquiry\_scan}$ (11.25ms default, see HCI [Part E] Section 7.3.22 on page 469). The standard scan is performed at a single hop frequency as defined by Xir_{4-0} (see Section 2.6.4.6 on page 82). The interlaced scan is performed as two back to back scans of $T_{w\_inquiry\_scan}$ where the first scan is on the normal hop frequency and the second scan is defined by [Xir_{4-0} + 16] mod 32. If the scan interval is not at least twice the scan window then interlaced scan shall not be used. The inquiry procedure uses 32 dedicated inquiry hop frequencies according to the inquiry hopping sequence. These frequencies are determined by the general inquiry address. The phase is determined by the native clock of the device carrying out the **inquiry scan**; the phase changes every 1.28s.

Instead of, or in addition to, the general inquiry access code, the device may scan for one or more dedicated inquiry access codes. However, the scanning shall follow the inquiry scan hopping sequence determined by the general inquiry address. If an inquiry message is received during an inquiry wake-up period, the device shall enter the **inquiry response** substate.

The **inquiry scan** substate can be entered from the **STANDBY** state or the **CONNECTION** state. In the **STANDBY** state, no connection has been established and the device can use all the capacity to carry out the **inquiry scan**. Before entering the **inquiry scan** substate from the **CONNECTION** state, the device should reserve as much capacity as possible for scanning. If desired, the device may place ACL logical transports in Sniff, Hold, or Park. Synchronous logical transports are preferably not interrupted by the **inquiry scan**, although eSCO retransmissions should be paused during the scan. In this case, the **inquiry scan** may be interrupted by the reserved synchronous slots. SCO packets should be used requiring the least amount of capacity (**HV3** packets). The scan window, $T_{w\ inquiry\ scan}$, shall be increased to increase the probability to respond to an inquiry message. If one SCO logical transport is present using HV3 packets and T_{SCO}=6 slots or one eSCO logical transport is present using EV3 packets and T_{ESCO}=6 slots, a total scan window of at least 36 slots (22.5ms) is recommended; if two SCO links are present using HV3 packets and T_{SCO}=6 slots or two eSCO links are present using EV3 packets and T_{ESCO}=6 slots, a total scan window of at least 54 slots (33.75ms) is recommended.

The scan interval $T_{inquiry\ scan}$ is defined as the interval between two consecutive inquiry scans. The **inquiry scan** interval shall be less than or equal to 2.56 s.

8.4.2 Inquiry substate

The **inquiry** substate is used to discover new devices. This substate is very similar to the **page** substate; the TX/RX timing shall be the same as in paging, see Section 2.4.4 on page 66 and Figure 2.7 on page 65. The TX and RX frequencies shall follow the inquiry hopping sequence and the inquiry response hopping sequence, and are determined by the general inquiry access code and the native clock of the discovering device. In between inquiry transmissions, the receiver shall scan for inquiry response messages. When a response is received, the entire packet (an **FHS** packet) is read, after which the device shall continue with inquiry transmissions. The device in an **inquiry** substate shall not acknowledge the inquiry response messages. If enabled by the Host (see HCI [Part E] Section 7.3.54 on page 506), the RSSI value of the inquiry response message shall be measured. It shall keep probing at different hop channels and in between listening for response packets. As in the **page** substate, two 10 ms trains **A** and **B** are defined, splitting the 32 frequencies of the inquiry hopping sequence into two 16-hop parts. A single train shall be repeated for at least $N_{inquiry}$=256 times before a new train is used. In order to collect all responses in an error-free environment, at least three train switches must have taken place. As a result, the **inquiry** substate may have to last for 10.24 s unless the inquirer collects enough responses and aborts the **inquiry** substate earlier. If desired, the inquirer may also prolong the **inquiry** substate to increase the probability of receiving all responses in an error-prone environment. If an inquiry procedure is automatically initiated periodically (say a 10 s period every minute), then the interval between two inquiry instances shall be determined randomly. This is done to avoid two devices synchronizing their inquiry procedures.

The **inquiry** substate is continued until stopped by the Baseband Resource Manager (when it decides that it has sufficient number of responses), when a timeout has been reached (*inquiryTO*), or by a command from the host to cancel the inquiry procedure.

The **inquiry** substate can be entered from the **STANDBY** state or the **CONNECTION** state. In the **STANDBY** state, no connection has been established and the device can use all the capacity to carry out the inquiry. Before entering the **inquiry** substate from the **CONNECTION** state, the device should free as much capacity as possible for scanning. To ensure this, it is recommended that the ACL logical transports are placed in Sniff, Hold, or Park. However, the reserved slots of synchronous logical transports shall not be disturbed by the inquiry. This means that the inquiry will be interrupted by the reserved SCO and eSCO slots which have higher priority than the inquiry. In order to obtain as much capacity as possible for inquiry, it is recommended to use the SCO packets which use the least amount of capacity (**HV3** packets). If SCO or eSCO links are present, the repetition number $N_{inquiry}$ shall be increased, see Table 8.4 on page 146.
Here it has been assumed that **HV3** packets are used with an interval T_{SCO}=6

slots or **EV3** packets are used with an interval of T_{ESCO}=6 slots, which would correspond to a 64 kb/s synchronous link.

	No synchronous links	One synchronous link (HV3)	Two synchronous links (HV3)
$N_{inquiry}$	≥ 256	≥ 512	≥ 768

Table 8.4: Increase of train repetition when synchronous links are present.

8.4.3 Inquiry response substate

The slave response substate for inquiries differs completely from the slave response substate applied for pages. When the inquiry message is received in the **inquiry scan** substate, the recipient shall return an inquiry response (FHS) packet containing the recipient's device address (BD_ADDR) and other parameters.

The following protocol in the slave's **inquiry response** shall be used. On the first inquiry message received in this substate the slave shall enter the **inquiry response** substate and shall return an **FHS** response packet to the master 625us after the inquiry message was received. A contention problem may arise when several devices are in close proximity to the inquiring device and all respond to an inquiry message at the same time. However, because every device has a free running clock it is highly unlikely that they all use the same phase of the inquiry hopping sequence. In order to avoid repeated collisions between devices that wake up in the same inquiry hop channel simultaneously, a device shall back-off for a random period of time. Thus, if the device receives an inquiry message and returns an FHS packet, it shall generate a random number, RAND, between 0 and MAX_RAND. For scanning intervals ≥ 1.28s MAX_RAND shall be 1023, however, for scanning intervals < 1.28s MAX_RAND may be as small as 127. A profile that uses a special DIAC may choose to use a smaller MAX_RAND than 1023 even when the scanning interval is ≥ 1.28s. The slave shall return to the **CONNECTION** or **STANDBY** state for the duration of at least RAND time slots. Before returning to the **CONNECTION** and **STANDY** state, the device may go through the **page scan** substate. After at least RAND slots, the device shall add an offset of 1 to the phase in the inquiry hop sequence (the phase has a 1.28 s resolution) and return to the **inquiry scan** substate again. If the slave is triggered again, it shall repeats the procedure using a new RAND. The offset to the clock accumulates each time an **FHS** packet is returned. During a probing window, a slave may respond multiple times, but on different frequencies and at different times. Reserved synchronous slots should have priority over response packets; that is, if a response packet overlaps with a reserved synchronous slot, it shall not be sent but the next inquiry message is awaited.

The messaging during the inquiry routines is summarized in Table 8.5 on page 147. In step 1, the master transmits an inquiry message using the inquiry access code and its own clock. The slave responds with the **FHS** packet containing the slave's Bluetooth device address, native clock and other slave information. This **FHS** packet is returned at times that tend to be random. The **FHS** packet is not acknowledged in the inquiry routine, but it is retransmitted at other times and frequencies as long as the master is probing with inquiry messages.

Step	Message	Packet Type	Direction	Hopping Sequence	Access Code and Clock
1	Inquiry	ID	master to slave	inquiry	inquiry
2	Inquiry response	FHS	slave to master	inquiry response	inquiry

Table 8.5: Messaging during inquiry routines.

8.5 CONNECTION STATE

In the **CONNECTION** state, the connection has been established and packets can be sent back and forth. In both devices, the channel (master) access code, the master's Bluetooth clock, and the AFH_channel_map are used. **CONNECTION** state uses the *basic* or *adapted channel hopping sequence*.

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

The first information packets in the **CONNECTION** state contain control messages that characterize the link and give more details regarding the devices. These messages are exchanged between the link managers of the devices. For example, they may define the SCO logical transport and the sniff parameters. Then the transfer of user information can start by alternately transmitting and receiving packets.

The **CONNECTION** state is left through a **detach** or **reset** command. The **detach** command is used if the link has been disconnected in the normal way; all configuration data in the link controller shall remain valid. The **reset** command is a soft reset of the link controller. The functionality of the soft reset is described in [Part E] Section 7.3.2 on page 442.

In the **CONNECTION** state, if a device is not going to be nominally present on the channel at all times it may describe its unavailability by using sniff mode or hold mode (see Figure 8.5 on page 148).

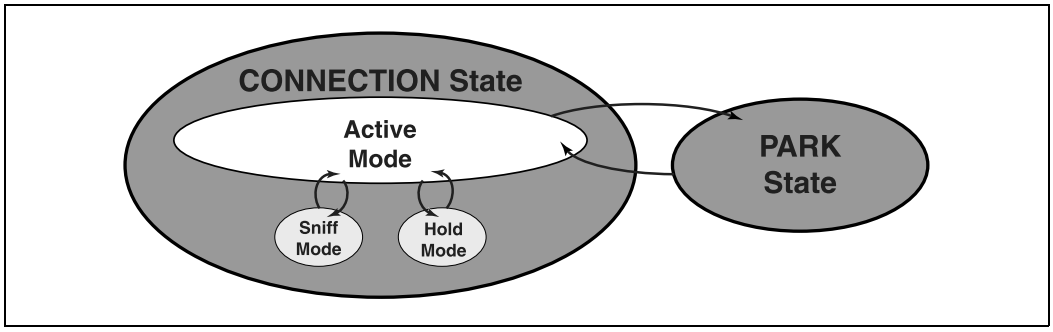


Figure 8.5: Connection state.

8.6 ACTIVE MODE

In the active mode, both master and slave actively participate on the channel. Up to seven slaves may be in the active mode at any given time. The master schedules the transmission based on traffic demands to and from the different slaves. In addition, it supports regular transmissions to keep slaves synchronized to the channel. Slaves in the active mode listen in the master-to-slave slots for packets. These devices are known as *active slaves*. If an active slave is not addressed, it may sleep until the next new master transmission. Slaves can derive the number of slots the master has reserved for transmission from TYPE field in the packet header; during this time, the non-addressed slaves do not have to listen on the master-to-slave slots. When a device is participating in multiple piconets it should listen in the master-to-slave slot for the current piconet. It is recommended that a device not be away from each piconet in which it is participating for more than T_{poll} slots. A periodic master transmission is required to keep the slaves synchronized to the channel. Since the slaves only need the channel access code to synchronize, any packet type can be used for this purpose.

Only the slave that is addressed by one of its LT_ADDRs (primary or secondary) may return a packet in the next slave-to-master slot. If no valid packet header is received, the slave may only respond in its reserved SCO or eSCO slave-to-master slot. In the case of a broadcast message, no slave shall return a packet (an exception is the access window for access requests in the **PARK** state, see Section 8.9 on page 165).

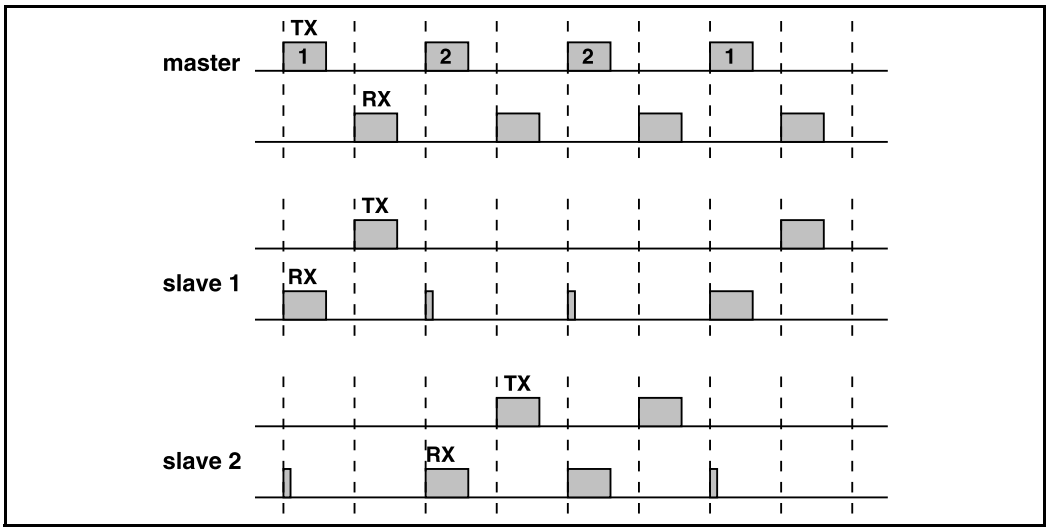


Figure 8.6: RX/TX timing in multi-slave configuration

8.6.1 Polling in the active mode

The master always has full control over the piconet. Due to the TDD scheme, slaves can only communicate with the master and not other slaves. In order to avoid collisions on the ACL logical transport, a slave is only allowed to transmit in the slave-to-master slot when addressed by the LT_ADDR in the packet header in the preceding master-to-slave slot. If the LT_ADDR in the preceding slot does not match, or a valid packet header was not received, the slave shall not transmit.

The master normally attempts to poll a slave's ACL logical transport no less often than once every T_{poll} slots. T_{poll} is set by the Link Manager (see [Part C] Section 4.1.8 on page 220).

The slave's ACL logical transport may be polled with any packet type except for FHS and ID. For example, polling during SCO may use HV packets, since the slave may respond to an HV packet with a DM1 packet (see Section 8.6.2 on page 150).

8.6.2 SCO

The SCO logical transport shall be established by the master sending an SCO setup message via the LM protocol. This message contains timing parameters including the SCO interval T_{SCO} and the offset D_{SCO} to specify the reserved slots.

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

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

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

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

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

The master will send SCO packets to the slave at regular intervals (the SCO interval T_{SCO} counted in slots) in the reserved master-to-slave slots. An HV1

packet can carry 1.25ms of speech at a 64 kb/s rate. An HV1 packet shall therefore be sent every two time slots (T_{SCO}=2). An HV2 packet can carry 2.5ms of speech at a 64 kb/s rate. An HV2 packet shall therefore be sent every four time slots (T_{SCO}=4). An HV3 packet can carries 3.75ms of speech at a 64 kb/s rate. An HV3 packet shall therefore be sent every six time slots (T_{SCO}=6).

The slave is allowed to transmit in the slot reserved for its SCO logical transport unless the (valid) LT_ADDR in the preceding slot indicates a different slave. If no valid LT_ADDR can be derived in the preceding slot, the slave may still transmit in the reserved SCO slot.

Since the DM1 packet is recognized on the SCO logical transport, it may be sent during the SCO reserved slots if a valid packet header with the primary LT_ADDR is received in the preceeding slot. DM1 packets sent during SCO reserved slots shall only be used to send ACL-C data.

The slave shall not transmit anything other than an HV packet in a reserved SCO slot unless it decodes its own slave address in the packet header of the packet in the preceding master-to-slave transmission slot.

8.6.3 eSCO

The eSCO logical transport is established by the master sending an eSCO setup message via the LM protocol. This message contains timing parameters including the eSCO interval T_{ESCO} and the offset D_{ESCO} to specify the reserved slots.

To enter eSCO, the master or slave shall send an eSCO setup command via the LM protocol. This message shall contain the eSCO interval T_{ESCO} and an offset D_{ESCO}. In order to prevent clock wrap-around problems, an initialization flag in the LMP setup message indicates whether initialization procedure 1 or 2 shall be used. The slave shall apply the initialization method as indicated by the initialization flag. The master shall use initialization 1 when the MSB of the current master clock (CLK_{27}) is 0; it shall use initialization 2 when the MSB of the current master clock (CLK_{27}) is 1. The master-to-slave eSCO slots reserved by the master and the slave shall be initialized on the slots for which the clock satisfies the applicable equation:

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

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

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

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

When an eSCO logical transport is established, the master shall assign an additional LT_ADDR to the slave. This provides the eSCO logical transport with an ARQ scheme that is separate from that of the ACL logical transport. All traffic on a particular eSCO logical transport, and only that eSCO traffic, is carried on the eSCO LT_ADDR. The eSCO ARQ scheme uses the ARQN bit in the packet header, and operates similarly to the ARQ scheme on ACL links.

There are two different polling rules in eSCO. In the eSCO reserved slots, the polling rule is the same as to the SCO reserved slots. The master may send a packet in the master slot. The slave may transmit on the eSCO LT_ADDR in the following slot either if it received a packet on the eSCO LT_ADDR in the previous slot, or if it did not receive a valid packet header in the previous slot. When the master-to-slave packet type is a three-slot packet, the slave's transmit slot is the fourth slot of the eSCO reserved slots. A master shall transmit in an eSCO retransmission window on a given eSCO LT_ADDR only if it addressed that eSCO LT_ADDR in the immediately preceeding eSCO reserved slots. A slave may transmit on an eSCO LT_ADDR in the eSCO reserved slots only if the slave did not received a valid packet header with a different LT_ADDR in the eSCO reserved slots. Inside retransmission windows, the same polling rule as for ACL traffic is used: the slave shall transmit on the eSCO channel only if it received a valid packet header and correct LT_ADDR on the eSCO channel in the previous master-to-slave transmission slot. The master may transmit on any non-eSCO LT_ADDR in any master-to-slave transmission slot inside the eSCO retransmission window. The master shall only transmit on an eSCO LT_ADDR in the retransmission window if there are enough slots left in the for both the master and slave packets to complete in the retransmission window. The master may refrain from transmitting in any slot during the eSCO retransmission window. When there is no data to transmit from master to slave, either due to the traffic being unidirectional or due to the master-to-slave packet having been ACK'ed, the master shall use the POLL packet. When the master-to-slave packet has been ACK'ed, and the slave-to-master packet has been correctly received, the master shall not address the slave on the eSCO LT_ADDR until the next eSCO reserved slot, with the exception that the master may transmit a NULL packet with ARQN=ACK on the eSCO LT_ADDR. When there is no data to transmit from slave to master, either due to the traffic being unidirectional or due to the slave-to-master packet having been ACK'ed, the slave shall use NULL packets. eSCO traffic should be given priority over ACL traffic in the retransmission window.

Figure 8.7 on page 153 shows the eSCO window when single slot packets are used.

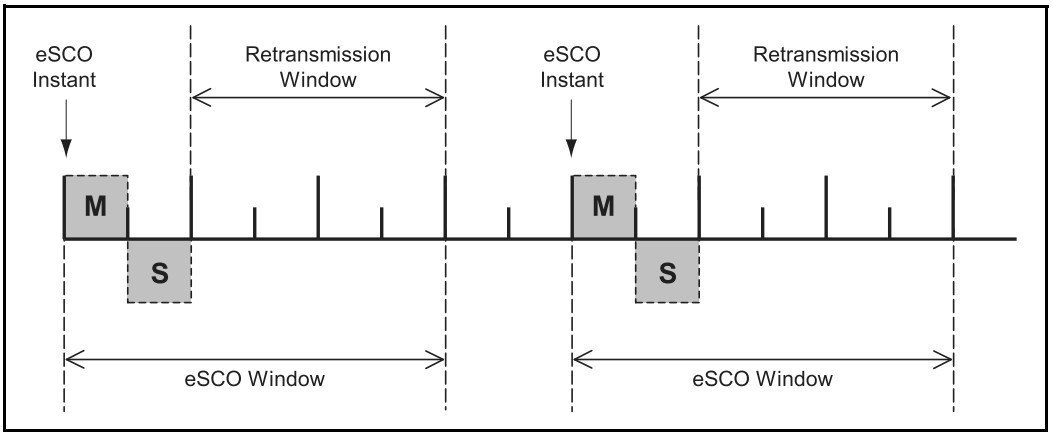


Figure 8.7: eSCO Window Details for Single-Slot Packets

When multi-slot packets are used in either direction of the eSCO logical transport, the first transmission continues into the following slots. The retransmission window in this case starts the slot after the end of the slave-to-master packet, i.e. two, four or six slots immediately following the eSCO instant are reserved and should not be used for other traffic. Figure 8.8 on page 153 shows the eSCO window when multi-slot packets are used in one direction and single-slot packets are used in the other direction.

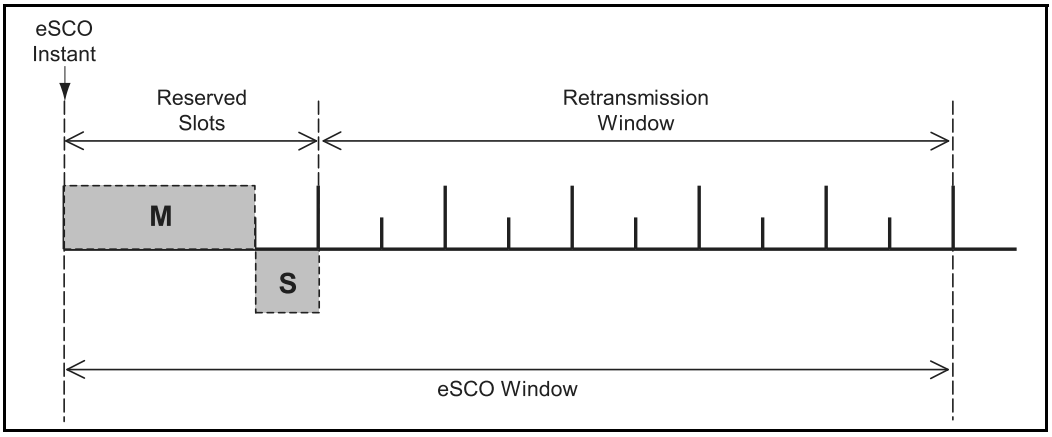


Figure 8.8: eSCO Window Details for Asymmetric Traffic

eSCO windows may overlap on the master, but shall not overlap on an individual slave.

In the reserved slots both master and slave shall only listen and transmit at their allocated slots at the first transmission time of each eSCO window. Intermittent lapses due to, for instance, time-critical signaling during connection establishment are allowed. Both master and slave may refrain from sending data and may use instead POLL and NULL packets respectively. When the master transmits a POLL packet instead of an EV4 or EV5 packet the slave shall transmit starting in the same slot as if the master transmitted an EV4 or

EV5 packet. If the slave does not receive anything in the reserved master-to-slave transmission slot it shall transmit in the same slot as if the master had transmitted the negotiated packet type. For example, if the master had negotiated an EV5 packet the slave would transmit three slots later. [If the master does not receive a slave transmission in response to an eSCO packet it causes an implicit NAK of the packet in question. The listening requirements for the slave during the retransmission window are the same as for an active ACL logical transport.

8.6.4 Broadcast scheme

The master of the piconet can broadcast messages to all slaves. A broadcast packet shall have an LT_ADDR set to all zero. Each new broadcast message (which may be carried by a number of packets) shall start with the start of L2CAP message indication (LLID=10).

A broadcast packet shall never be acknowledged. In an error-prone environment, the master may carry out a number of retransmissions to increase the probability for error-free delivery, see also Section 7.6.5 on page 130.

In order to support the **PARK** state (as described in Section 8.9 on page 165), a master transmission shall take place at fixed intervals. This master transmission will act as a beacon to which slaves can synchronize. If no traffic takes place at the beacon event, broadcast packets shall be sent. More information is given in Section 8.9 on page 165.

8.6.5 Role switch

There are several occasions when a role switch is used.

- a role switch is necessary when joining an existing piconet by paging, since by definition, the paging device is initially master of a "small" piconet only involving the pager (master) and the paged (slave) device.
- a role switch is necessary in order for a slave in an existing piconet to set up a new piconet with itself as master and the original piconet master as slave. If the original piconet had more than one slave, then this implies a double role for the original piconet master; it becomes a slave in the new piconet while still maintaining the original piconet as master.

Prior to the role switch, encryption if present, shall be stopped in the old piconet. A role switch shall not be performed if the physical link is in Sniff or Hold mode, in the **PARK** state, or has any synchronous logical transports.

For the master and slave involved in the role switch, the switch results in a reversal of their TX and RX timing: a TDD switch. Additionally, since the piconet parameters are derived from the Bluetooth device address and clock of the master, a role switch inherently involves a redefinition of the piconet as well: a piconet switch. The new piconet's parameters shall be derived from the former slave's device address and clock.

Assume device A is to become master; device B was the former master. Then there are two alternatives: either the slave initiates the role switch or the master initiates the role switch. These alternatives are described in Link Manager Protocol, [Part C] Section 4.4.2 on page 243. The baseband procedure is the same regardless of which alternative is used.

In step 1, the slave A and master B shall perform a TDD switch using the former hopping scheme (still using the Bluetooth device address and clock of device B), so there is no piconet switch yet. The slot offset information sent by slave A shall not be used yet but shall be used in step 3. Device A now becomes the master, device B the slave. The LT_ADDR formerly used by device A in its slave role, shall now be used by slave B.

At the moment of the TDD switch, both devices A and B shall start a timer with a time out of *newconnectionTO*. The timer shall be stopped in slave B as soon as it receives an FHS packet from master A on the TDD-switched channel. The timer shall be stopped in master A as soon as it receives an ID packet from slave B. If the *newconnectionTO* expires, the master and slave shall return to the old piconet timing and AFH state, taking their old roles of master and slave. The FHS packet shall be sent by master A using the "old" piconet parameters. The LT_ADDR in the FHS packet header shall be the former LT_ADDR used by device A. The LT_ADDR carried in the FHS payload shall be the new LT_ADDR intended for device B when operating on the new piconet. After the FHS acknowledgment, which is the ID packet and shall be sent by the slave on the old hopping sequence, both master A and slave B shall use the new channel parameters of the new piconet as indicated by the FHS with the sequence selection set to *basic channel hopping sequence*. If the new master has physi-

cal links that are *AFH enabled*, following the piconet switch the new master is responsible for controlling the AFH operational mode of its new slave.

Since the old and new masters' clocks are synchronized, the clock information sent in the FHS payload shall indicate the new master's clock at the begining of the FHS packet transmission. Furthermore, the 1.25 ms resolution of the clock information given in the FHS packet is not sufficient for aligning the slot boundaries of the two piconets. The slot-offset information in the LMP message previously sent by device A shall be used to provide more accurate timing information. The slot offset indicates the delay between the start of the master-to-slave slots of the old and new piconet channels. This timing information ranges from 0 to 1249 μs with a resolution of 1μs. It shall be used together with the clock information in the FHS packet to accurately position the correlation window when switching to the new master's timing after acknowledgment of the FHS packet.

After reception of the FHS packet acknowledgment, the new master A shall switch to its own timing with the sequence selection set to the *basic channel hopping sequence* and shall send a POLL packet to verify the switch. Both the master and the slave shall start a new timer with a time out of *newconnectionTO* on FHS packet acknowledgment. The start of this timer shall be aligned with the beginning of the first master TX slot boundary of the new piconet, following the FHS packet acknowledgment. The slave shall stop the timer when the POLL packet is received; the master shall stop the timer when the POLL packet is acknowledged. The slave shall respond with any type of packet to acknowledge the POLL. Any pending AFH_Instant shall be cancelled once the POLL packet has been received by the slave. If no response is received, the master shall re-send the POLL packet until *newconnectionTO* is reached. If this timer expires, both the slave and the master shall return to the old piconet timing with the old master and slave roles. Expiry of the timer shall also restore the state associated with AFH (including any pending AFH_Instant), Channel Quality Driven Data Rate (CQDDR, Link Manager Protocol [Part C] Section 4.1.7 on page 219) and power control (Link Manager Protocol [Part C] Section 4.1.3 on page 211). The procedure may then start again beginning at step 1. Aligning the timer with TX boundaries of the new piconet ensures that no device returns to the old piconet timing in the middle of a master RX slot.

After the role switch the ACL logical transport is reinitialized as if it were a new connection. For example, the SEQN of the first data packet containing a CRC on the new piconet channel shall be set according to the rules in section 7.6.2 on page 127.

A parked slave must be unparked before it can participate in a role switch.

8.6.6 Scatternet

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

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

A master or slave can become a slave in another piconet by being paged by the master of this other piconet. On the other hand, a device participating in one piconet can page the master or slave of another piconet. Since the paging device always starts out as master, a master-slave role switch is required if a slave role is desired. This is described in the Section 8.6.5 on page 155.

8.6.6.1 Inter-piconet communications

Time multiplexing must be used to switch between piconets. Devices may achieve the time multiplexing necessary to implement scatternet by using sniff mode or by remaining in an active ACL connection. For an ACL connection in piconets where the device is a slave in the **CONNECTION** state, it may choose not to listen in every master slot. In this case it should be recognized that the quality of service on this link may degrade abruptly if the slave is not present enough to match up with the master polling that slave. Similarly, in piconets where the device is master it may choose not to transmit in every master slot. In this case it is important to honor T_{poll} as much as possible. Devices in sniff mode may have sufficient time to visit another piconet in between sniff slots. When the device is a slave using sniff mode and there are not sufficient idle slots, the device may choose to not listen to all master transmission slots in the sniff_attempts period or during the subsequent sniff_timeout period. A master is not required to transmit during sniff slots and therefore has flexibility for scatternet. If SCO or eSCO links are established, other piconets shall only be visited in the non-reserved slots in between reserved slots. This is only possible if there is a single SCO logical transport using HV3 packets or eSCO links where at least four slots remain in between the reserved slots. Since the multiple piconets are not synchronized, guard time must be left to account for misalignment. This means that only 2 slots can effectively be used to visit another piconet in between the HV3 packets.

Since the clocks of two masters of different piconets are not synchronized, a slave device participating in two piconets shall maintain two offsets that, added to its own native clock, create the two master clocks. Since the two master clocks drift independently, the slave must regularly update the offsets in order to keep synchronization to both masters.

8.6.7 Hop sequence switching

Hop sequence adaptation is controlled by the master and can be set to either *enabled* or *disabled*. Once enabled, hop sequence adaptation shall apply to all logical transports on a physical link. Once enabled, the master may periodically update the set of *used* and *unused* channels as well as disable hop sequence adaptation on a physical link. When a master has multiple physical links the state of each link is independent of all other physical links.

When hop sequence adaptation is enabled, the *sequence selection* hop selection kernel input is set to *adapted channel hopping sequence* and the *AFH_channel_map* input is set to the appropriate set of *used* and *unused* channels. Additionally, the *same channel* mechanism shall be used. When hop sequence adaptation is enabled with all channels *used* this is known as AHS(79).

When hop sequence adaptation is disabled, the *sequence selection* input of the hop selection kernel is set to *basic channel hopping sequence* (the *AFH_channel_map* input is unused in this case) and the *same channel* mechanism shall not be used.

The hop sequence adaptation state shall be changed when the master sends the LMP_set_AFH PDU and a baseband acknowledgement is received. When the baseband acknowledgement is received prior to the hop sequence switch instant, *AFH_Instant*, (See Link Manager Protocol [Part C] Section 4.1.4 on page 213) the hop sequence proceeds as shown in Figure 8.9 on page 158.

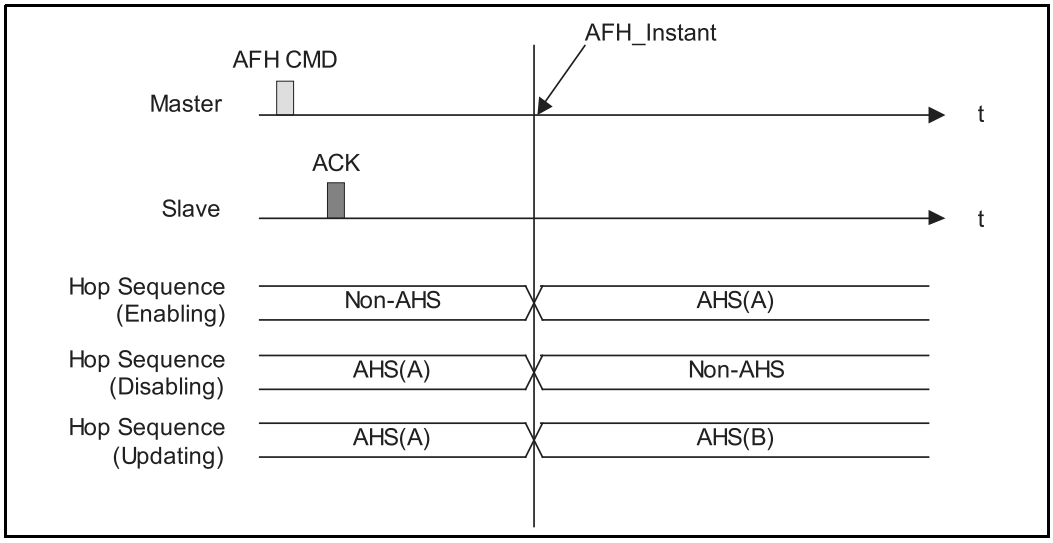


Figure 8.9: Successful hop sequence switching

When the baseband acknowledgement is not received prior to the *AFH_Instant* the master shall use a recovery hop sequence for the slave(s) that did not respond with an acknowledgement (this may be because the slave did not hear the master's transmission or the master did not hear the slave's transmission). When hop sequence adaptation is being enabled or disabled the recovery sequence shall be the *AFH_channel_map* specified in the LMP_set_AFH PDU. When the *AFH_channel_map* is being updated the master shall choose a recovery sequence that includes all of the RF channels marked as *used* in either the old or new *AFH_channel_map,* e.g. AHS(79). Once the baseband acknowledgement is received the master shall use the *AFH_channel_map* in the LMP_set_AFH PDU starting with the next transmission to the slave. See Figure 8.10 on page 159.

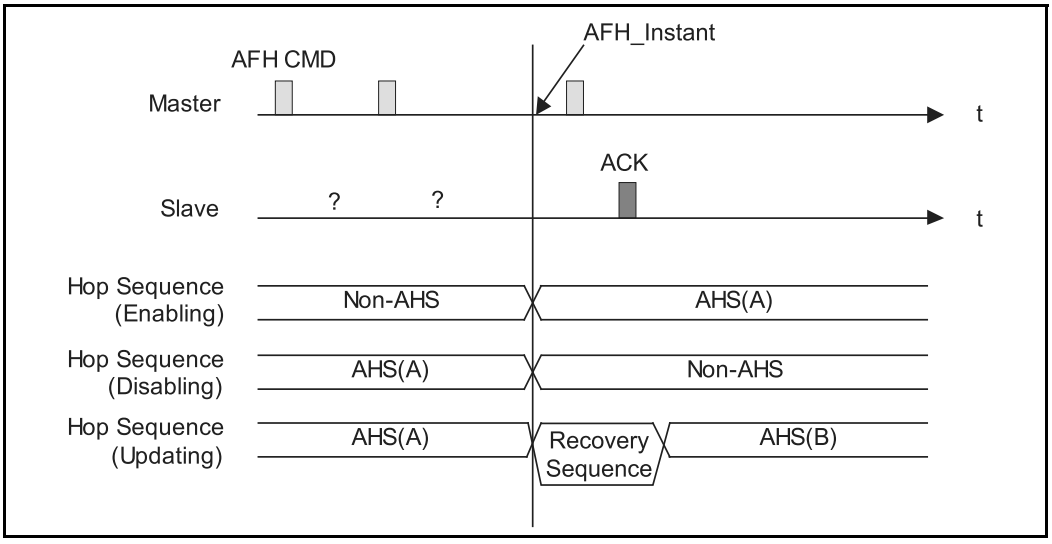


Figure 8.10: Recovery hop sequences

When the *AFH_Instant* occurs during a multi-slot packet transmitted by the master, the slave shall use the same hopping sequence parameters as the master used at the start of the multi-slot packet. An example of this is shown in Figure 8.11 on page 160. In this figure the *basic channel hopping sequence* is designated f. The first *adapted channel hopping sequence* is designated with f', and the second *adapted channel hopping sequence* is designated f''.

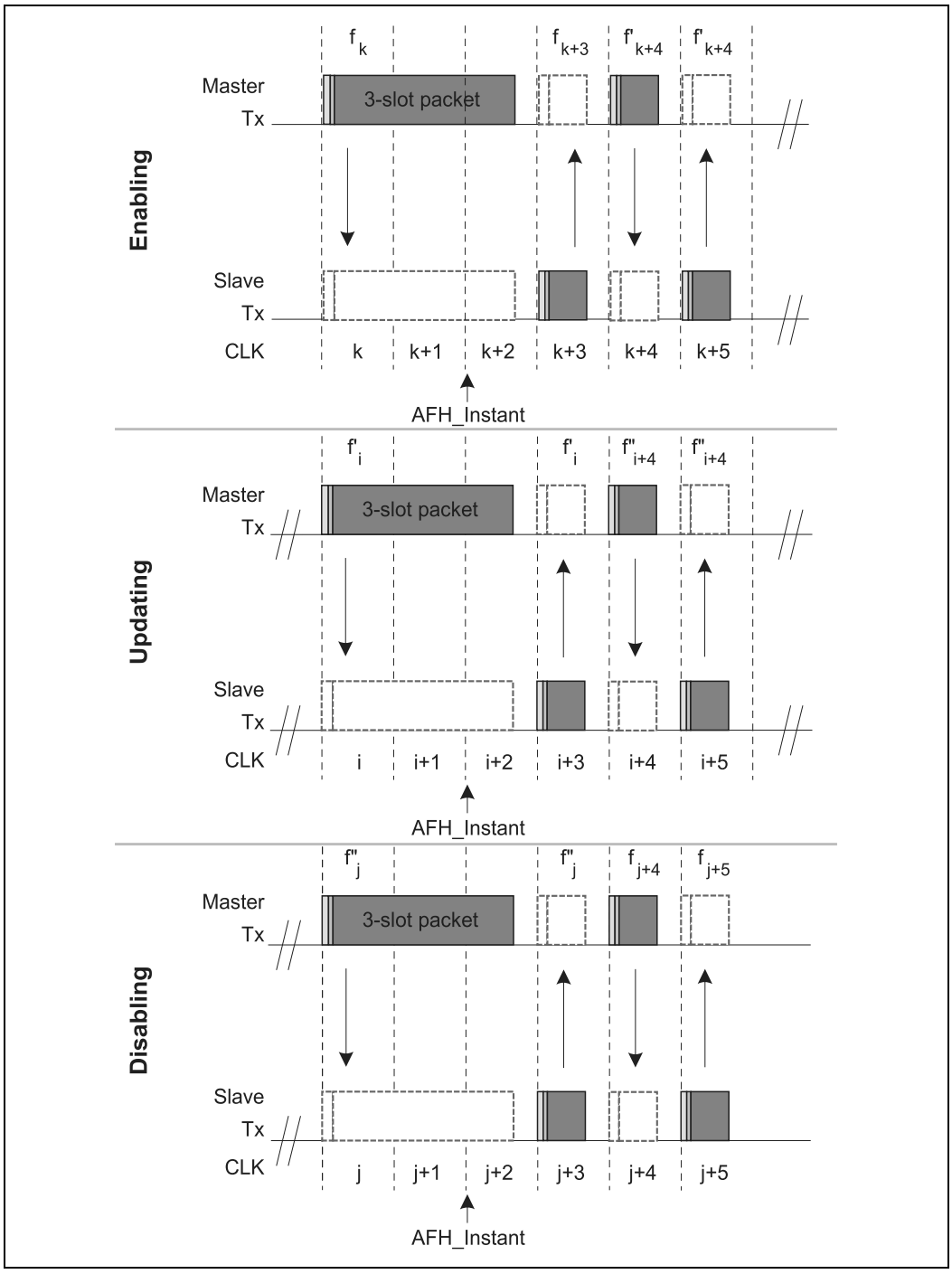


Figure 8.11: AFH_Instant changes during multi-slot packets transmitted by the master.

8.6.8 Channel classification and channel map selection

RF channels are classified as being *unknown, bad or good*. These classifications are determined individually by the master and slaves based on local information (e.g. active or passive channel assessment methods or from the Host via HCI). Information received from other devices via LMP (e.g. an *AFH_channel_map* from a master or a channel classification report from a slave) shall not be included in a device's channel classification.

The three possible channel classifications shall be as defined in Table 8.6 on page 161.

Classification	Definition
unknown	A channel shall be classified as *unknown* if the channel assessment measurements are insufficient to reliably classify the channel, and the channel is not marked as *bad* in the most recent HCI *Set_AFH_Channel_Classification*.
bad	A channel may be classified as *bad* if an ACL or synchronous throughput failure measure associated with it has exceeded a threshold (defined by the particular channel assessment algorithm employed). A channel may also be classified as *bad* if an interference-level measure associated with it, determining the interference level that the link poses upon other systems in the vicinity, has exceeded a threshold (defined by the particular channel assessment algorithm employed). A channel shall be classified as *bad* when it is marked as *bad* in the most recent HCI *Set_AFH_Channel_Classification* command.
good	A channel shall be classified as *good* if it is not either *unknown* or *bad*.

Table 8.6: Channel classification descriptions

A master with AFH enabled physical links shall determine an *AFH_channel_map* based on any combination of the following information:

- Channel classification from local measurements (e.g. active or passive channel assessment in the Controller), if supported and enabled. The Host may enable or disable local measurements using the HCI *Write_AFH_Channel_Classification_Mode* command, defined in the HCI Functional Specification [Part E] Section 7.3.58 on page 510 if HCI is present.
- Channel classification information from the Host using the HCI *Set_AFH_channel_classification* command, defined in the HCI Functional Specification [Part E] Section 7.3.58 on page 510 if HCI is present. Channels classified as *bad* in the most recent *AFH_Host_Channel_Classification* shall be marked as *unused* in the *AFH_channel_map*.
- Channel classification reports received from slaves in *LMP_channel_classification* PDUs, defined in the LMP Specification [Part C] Section 4.1.5 on page 216.

The algorithm used by the master to combine these information sources and generate the *AFH_channel_map* is not defined in the specification and will be implementation specific. At no time shall the number of channels used be less than N_{min}, defined in Section 2.3.1 on page 63.

If a master that determines that all channels should be *used* it may keep AFH operation enabled using an *AFH_channel_map* of 79 *used* channels, i.e. AHS(79).

8.6.9 Power Management

Features are provided to allow low-power operation. These features are both at the microscopic level when handling the packets, and at the macroscopic level when using certain operation modes.

8.6.9.1 Packet handling

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

8.6.9.2 Slot occupancy

As was described in Section 6.5 on page 105, the packet type indicates how many slots a packet may occupy. A slave not addressed in the packet header may go to sleep for the remaining slots the packet occupies. This can be read from the TYPE code.

8.6.9.3 Recommendations for low-power operation

The most common and flexible methods for reducing power consumption are the use of sniff and park. Hold can also be used by repeated negotiation of hold periods.

8.7 SNIFF MODE

In sniff mode, the duty cycle of the slave's activity in the piconet may be reduced. If a slave is in active mode on an ACL logical transport, it shall listen in every ACL slot to the master traffic, unless that link is being treated as a scatternet link or is absent due to hold mode. With sniff mode, the time slots when a slave is listening are reduced, so the master shall only transmit to a slave in specified time slots. The sniff anchor points are spaced regularly with an interval of T_{sniff}.

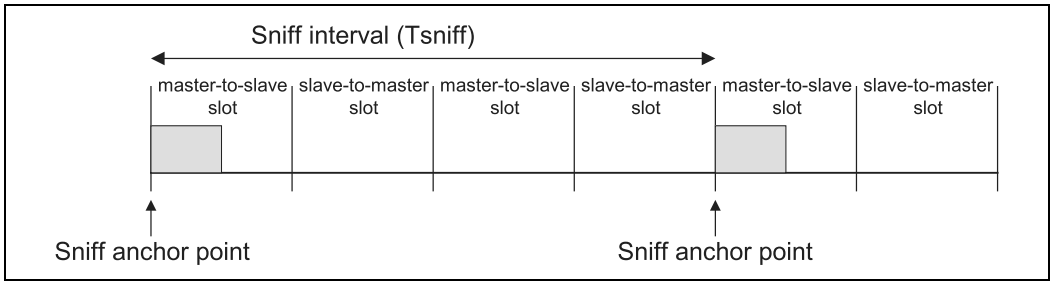


Figure 8.12: Sniff anchor points

The slave listens in master-to-slave transmission slots starting at the sniff anchor point. It shall use the following rules to determine whether to continue listening:

- If fewer than $N_{sniff\ attempt}$ master-to-slave transmission slots have elapsed since the sniff anchor point then the slave shall continue listening.
- If the slave has received a packet with a matching LT_ADDR that contains ACL data (DM, DH, DV, or AUX1 packets) in the preceding $N_{sniff\ timeout}$ master-to-slave transmission slots then it shall continue listening.
- If the slave has transmitted a packet containing ACL data (DM, DH, DV, or AUX1 packets) in the preceding $N_{sniff\ timeout}$ slave-to-master transmission slots then it shall continue listening.
- If the slave has received any packet with a matching LT_ADDR in the preceding $N_{sniff\ timeout}$ master-to-slave transmission slots then it may continue listening.
- A device may override the rules above and stop listening prior to $N_{sniff\ timeout}$ or the remaining $N_{sniff\ attempt}$ slots if it has activity in another piconet.

It is possible that activity from one sniff timeout may extend to the next sniff anchor point. Any activity from a previous sniff timeout shall not affect activity after the next sniff anchor point. So in the above rules, only the slots since the last sniff anchor point are considered.

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

$N_{sniff\ attempt}$ =0 shall not be used.

Sniff mode only applies to asynchronous logical transports and their associated LT_ADDR. Sniff mode shall not apply to synchronous logical transports, therefore, both masters and slaves shall still respect the reserved slots and retransmission windows of synchronous links.

To enter sniff mode, the master or slave shall issue a sniff command via the LM protocol. This message includes the sniff interval T_{sniff} and an offset $D_{sniff.}$ In addition, an initialization flag indicates whether initialization procedure 1 or 2 shall be used. The device shall use initialization 1 when the MSB of the current master clock (CLK_{27}) is 0; it shall use initialization 2 when the MSB of the current master clock (CLK_{27}) is 1. The slave shall apply the initialization method as indicated by the initialization flag irrespective of its clock bit value CLK_{27}. The sniff anchor point determined by the master and the slave shall be initialized on the slots for which the clock satisfies the applicable equation:

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

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

this implies that D_{sniff} must be even

After initialization, the clock value CLK(k+1) for the next sniff anchor point shall be derived by adding the fixed interval T_{sniff} to the clock value of the current sniff anchor point:

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

8.7.1 Sniff Transition Mode

Sniff transition mode is a special mode which is used during the transition between Sniff and active mode. It is required because during this transition it is unclear which mode (Sniff or Active) the slave is in and it is necessary to ensure that the slave is polled correctly regardless of which mode it is in.

In sniff transition mode the master shall maintain the active mode poll interval in case the slave is in active mode. In addition the master shall poll the slave at least once in the sniff attempt transmit slots starting at each sniff instant: note that this transmission counts for the active mode polling as well. The master must use its high power accurate clock when in Sniff Transition Mode.

The precise circumstances under which the master enters Sniff Transition Mode are defined in [Part C] Section 4.5.3.1 on page 254.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 322

8.8 HOLD MODE

During the **CONNECTION** state, the ACL logical transport to a slave can be put in a **hold** mode. In **hold** mode the slave temporarily shall not support ACL packets on the channel. Any synchronous packet during reserved synchonous slots (from SCO and eSCO links) shall be supported. With the **hold** mode, capacity can be made free to do other things like scanning, paging, inquiring, or attending another piconet. The device in **hold** mode can also enter a low-power sleep mode. During **hold** mode, the slave device keeps its logical transport address(es) (LT_ADDR).

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

8.9 PARK STATE

When a slave does not need to participate on the piconet channel, but still needs to remain synchronized to the channel, it can enter **PARK** state. **PARK** state is a state with very little activity in the slave. In the **PARK** state, the slave shall give up its logical transport address LT_ADDR. Instead, it shall receive two new addresses to be used in the **PARK** state

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

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

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

The master shall maintain separate non-overlapping park beacon structures for each hop sequence. The beacon structures shall not overlap either their beacon slots or access windows.

In addition for using it for low power consumption, park is used to connect more than seven slaves to a single master. At any one time, only seven slaves can be in the **CONNECTION** state. However, by swapping active and parked slaves out respectively in the piconet, the number of slaves can be much larger (255 if the PM_ADDR is used, and an arbitrarily large number if the BD_ADDR is used).

8.9.1 Beacon train

To support parked slaves, the master establishes a beacon train when one or more slaves are parked. The beacon train consists of one beacon slot or a train of equidistant beacon slots which is transmitted periodically with a constant time interval. The beacon train is illustrated in Figure 8.13 on page 167. A train of N_B ($N_B \geq 1$) beacon slots is defined with an interval of T_B slots. The beacon slots in the train are separated by Δ_B . The start of the first beacon slot is referred to as the **beacon instant** and serves as the beacon timing reference. The beacon parameters N_B and T_B are chosen such that there are sufficient beacon slots for a parked slave to synchronize to during a certain time window in an error-prone environment.

When parked, the slave shall receive the beacon parameters through an LMP command. In addition, the timing of the beacon instant is indicated through the offset D_B. As with the SCO logical transport (see Section 8.6.2 on page 150), two initialization procedures 1 or 2 are used. The master shall use initialization 1 when the MSB of the current master clock (CLK_{27}) is 0; it shall use initialization 2 when the MSB of the current master clock (CLK_{27}) is 1. The chosen initialization procedure shall also be carried by an initialization flag in the LMP command. The slave shall apply the initialization method as indicated by the initialization flag irrespective of its clock bit CLK_{27}. The master-to-slave slot positioned at the beacon instant shall be initialized on the slots for which the clock satisfies the applicable equation:

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

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

this implies that DB will be even

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

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

The beacon train serves four purposes:

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

Since a slave can synchronize to any packet which is preceded by the proper channel access code, the packets carried on the beacon slots do not have to contain specific broadcast packets for parked slaves to be able to synchronize; any packet may be used. The only requirement placed on the beacon slots is that there is a master-to-slave transmission present on the hopping sequence associated with the park structure. If there is no information to be sent, **NULL** packets may be transmitted by the master. If there is indeed broadcast information to be sent to the parked slaves, the first packet of the broadcast message shall be repeated in every beacon slot of the beacon train. However, synchronous traffic in the synchronous reserved slots may interrupt the beacon transmission if it is on the same hopping sequence as the parked slaves. The master shall configure its park beacon structure so that reserved slots of synchronous logical transports do not cause slaves to miss synchronisation on a beacon slot. For example, a master that has active slaves using AHS, and parked slaves using Non-AHS shall ensure that the Park beacons cannot be interrupted by AHS synchronous reserved slots.

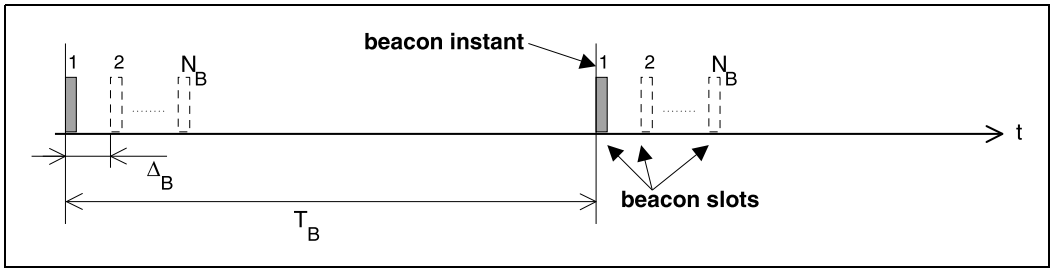


Figure 8.13: General beacon train format

The master can place parked slaves in any of the AFH operating modes, but shall ensure that all parked slaves use the same hop sequence. Masters should use AHS(79) or AHS when all the slaves in the Piconet are AFH capable.

A master that switches a slave between AFH enabled, AFH disabled or AHS(79) operation shall ensure that the AFH_Instant occurs before transmission of the beacon train using this hop sequence.

The master communicates with parked slaves using broadcast messages. Since these messages can be time - critical, an ongoing repetition train of broadcast message may be prematurely aborted by broadcast information destined to parked slaves in beacon slots and in access windows (see Section 8.9.2 on page 168).

8.9.2 Beacon access window

In addition to the beacon slots, an access window is defined where the parked slaves can send requests to be unparked. To increase reliability, the access window may be repeated M_{access} times ($M_{access} \geq 1$), see Figure 8.14 on page 168. The access window starts a fixed delay D_{access} after the beacon instant. The width of the access window is T_{access}.

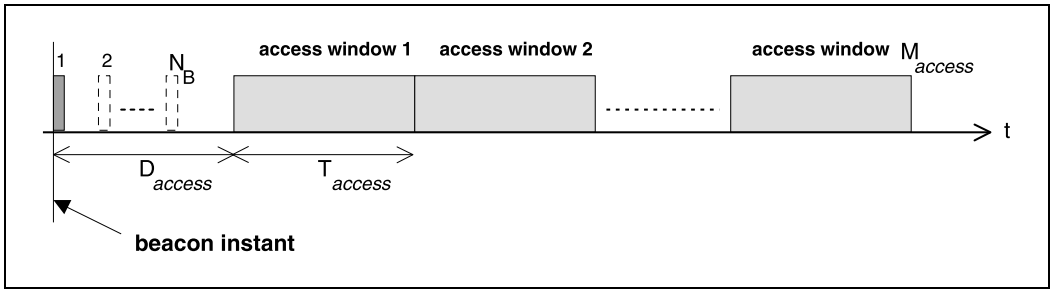


Figure 8.14: Definition of access window

The access window supports a polling slave access technique. The format of the polling technique is shown in Figure 8.15 on page 168. The same TDD structure is used as on the piconet channel, i.e. master-to-slave transmission is alternated with slave-to-master transmission. The slave-to-master slot is divided into two half slots of 312.5 µs each. The half slot a parked slave is allowed to respond in corresponds to its access request address (AR_ADDR), see also section 8.9.6 on page 171. For counting the half slots to determine the access request slot, the start of the access window is used, see Figure 8.15 on page 168. The slave shall only send an access request in the proper slave-to-master half slot if a broadcast packet has been received in the preceding master-to-slave slot. In this way, the master polls the parked slaves.

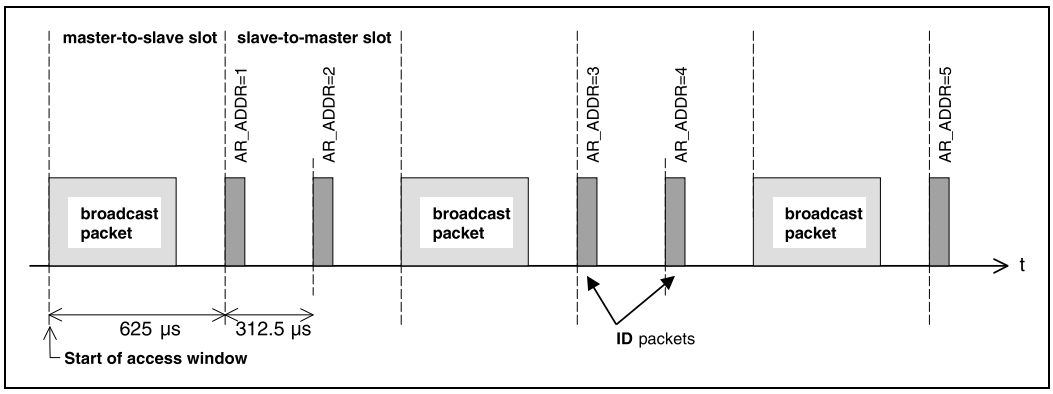


Figure 8.15: Access procedure applying the polling technique.

The slots of the access window may also be used for traffic on the piconet if required. For example, if a synchronous connection has to be supported, the slots reserved for the synchronous link may carry synchronous information

instead of being used for access requests, i.e. if the master-to-slave slot in the access window contains a packet different from a broadcast packet, the following slave-to-master slot shall not be used for slave access requests. If the master transmits a broadcast packet in the access window then it shall use the hop sequence associated with the park structure. Slots in the access window not affected by piconet traffic may still be used according to the defined access structure, (an example is shown in Figure 8.16 on page 169) the access procedure shall be continued as if no interruption had taken place.

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

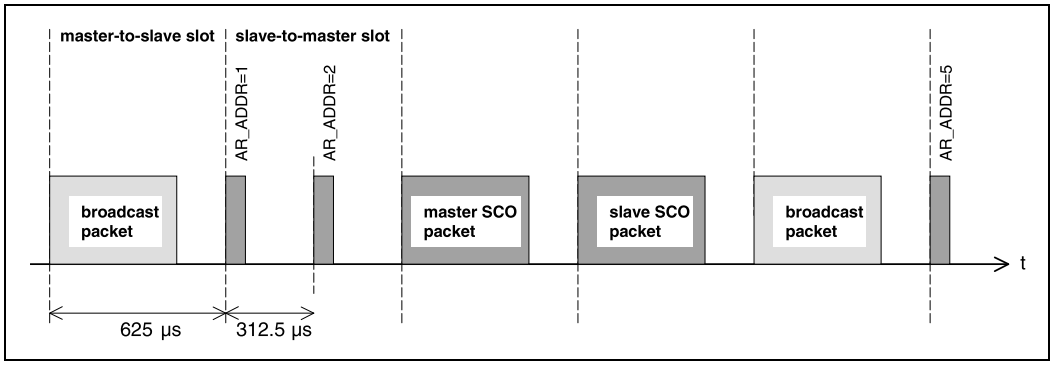


Figure 8.16: Disturbance of access window by SCO traffic

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

8.9.3 Parked slave synchronization

Parked slaves wake up periodically to re-synchronize to the channel. Any packet exchanged on the channel can be used for synchronization. Since master transmission is mandatory on the beacon slots, parked slaves will use the beacon train to re-synchronize. A parked slave may wake-up at the beacon instant to read the packet sent on the first beacon slot. If this fails, it may retry on the next beacon slot in the beacon train; in total, there are N_B opportunities per beacon instant to re-synchronize. During the search, the slave may increase its search window, see also Section 2.2.5.2 on page 62. The separation between the beacon slots in the beacon train Δ_B shall be chosen such that consecutive search windows will not overlap.

The parked slave may not wake up at every beacon instant. Instead, a sleep interval may be applied which is longer than the beacon interval T_B, see Figure 8.17 on page 170. The slave sleep window shall be a multiple N_{B_sleep} of T_B.

The precise beacon instant the slave may wake up on shall be indicated by the master with D_{B_sleep} which indicates the offset (in multiples of T_B) with respect to the beacon instant ($0 < D_{B\_sleep} < N_{B\_sleep}-1$). To initialize the wake-up period, the applicable equation shall be used:

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

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

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

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

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

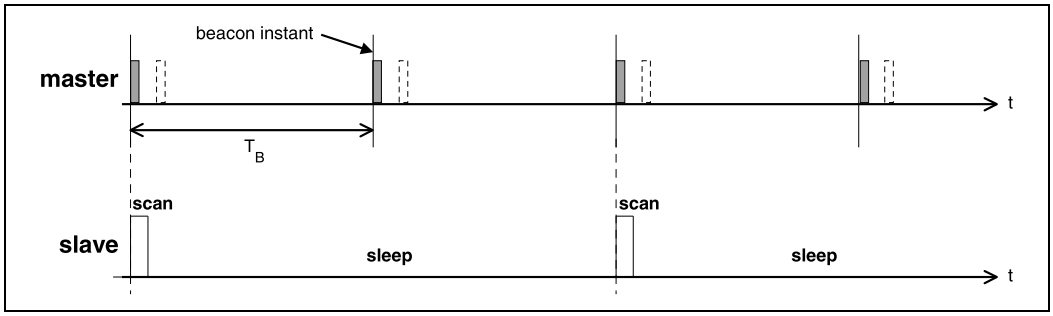


Figure 8.17: Extended sleep interval of parked slaves.

8.9.4 Parking

A master can park an active slave through the exchange of LMP commands. Before being put into park, the slave shall be assigned a PM_ADDR and an AR_ADDR. Every parked slave shall have a unique PM_ADDR or a PM_ADDR of 0. The AR_ADDR is not necessarily unique. The beacon parameters shall be given by the master when the slave is parked. The slave shall then give up its LT_ADDR and shall enter **PARK** state. A master can park only a single slave at a time. The park message is carried with a normal data packet and addresses the slave through its LT_ADDR.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 328

8.9.5 Master-initiated unparking

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

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

When a device is unparked, the SEQN bit for the link shall be reset to 1 on both the master and the slave (see Section 7.6.2.1 on page 128).

8.9.6 Slave-initiated unparking

A slave can request access to the channel through the access window defined in section 8.9.2 on page 168. As shown in Figure 8.15 on page 168, the access window includes several slave-to-master half slots where the slave may send an access request message. The specific half slot the slave is allowed to respond in, corresponds to its access request address (AR_ADDR) which it received when it was parked. The order of the half slots (in Figure 8.15 the AR_ADDR numbers linearly increase from 1 to 5) is not fixed: an LMP command sent in the beacon slots may reconfigure the access window. When a slave desires access to the channel, it shall send an access request message in the proper slave-to-master half slot. The access request message of the slave is the **ID** packet containing the device access code (DAC) of the master (which is the channel access code without the trailer). The parked slave shall only transmit an access request message in the half slot, when in the preceding master-to-slave slot a broadcast packet has been received. This broadcast message may contain any kind of broadcast information not necessarily related to the parked slave(s). If no broadcast information is available, a broadcast **NULL** or broadcast **POLL** packet shall be sent.

After having sent an access request, the parked slave shall listen for an unpark message from the master. As long as no unpark message is received, the slave shall repeat the access requests in the subsequent access windows. After the last access window (there are M_{access} windows in total, see Section 8.9.2 on page 168), the parked slave shall listen for an additional N_{poll} time slots for an unpark message. If no unpark message is received within N_{poll} slots after the end of the last access window, the slave may return to sleep and retry an access attempt after the next beacon instant.

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

When a device is unparked, the SEQN bit for the link shall be reset to 1 on both the master and the slave (see Section 7.6.2.1 on page 128).

8.9.7 Broadcast scan window

In the beacon train, the master can support broadcast messages to the parked slaves. However, it may extend its broadcast capacity by indicating to the parked slaves that more broadcast information is following after the beacon train. This is achieved by an LMP command ordering the parked slaves (as well as the active slaves) to listen to the channel for broadcast messages during a limited time window. This time window starts at the beacon instant and continues for the period indicated in the LMP command sent in the beacon train.

8.9.8 Polling in the park state

In the **PARK** state, parked slaves may send access requests in the access window provided a broadcast packet is received in the preceding master-to-slave slot. Slaves in the **CONNECTION** state shall not send in the slave-to-master slots following the broadcast packet, since they are only allowed to send if addressed specifically.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 330

9 AUDIO

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

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

Table 9.1 on page 173 summarizes the voice coding schemes supported on the air interface. The appropriate voice coding scheme is selected after negotiations between the Link Managers.

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

Table 9.1: Voice coding schemes supported on the air interface.

9.1 LOG PCM CODEC

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

9.2 CVSD CODEC

A more robust format for voice over the air interface is delta modulation. This modulation scheme follows the waveform where the output bits indicate whether the prediction value is smaller or larger then the input waveform. To reduce slope overload effects, syllabic companding is applied: the step size is adapted according to the average signal slope. The input to the CVSD encoder shall be 64 ksamples/s linear PCM (typically 16 bits, but actual value is implementation specific). Block diagrams of the CVSD encoder and CVSD decoder are shown in Figure 9.1 on page 174, Figure 9.2 on page 174 and Figure 9.3 on page 174. The system shall be clocked at 64 kHz.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 331

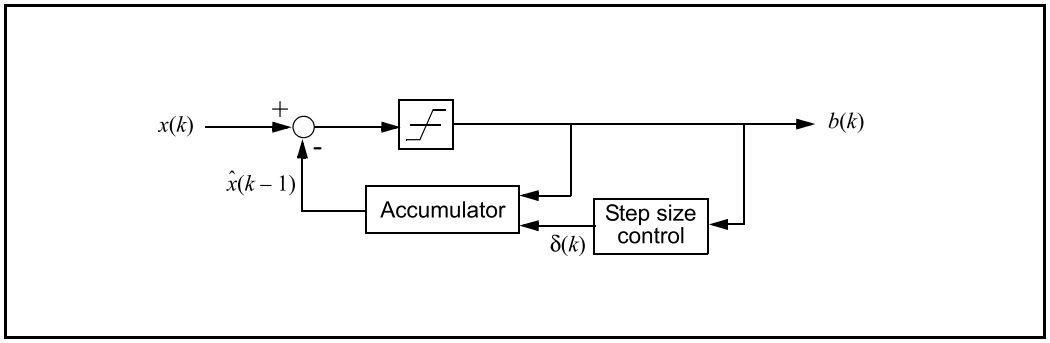


Figure 9.1: Block diagram of CVSD encoder with syllabic companding.

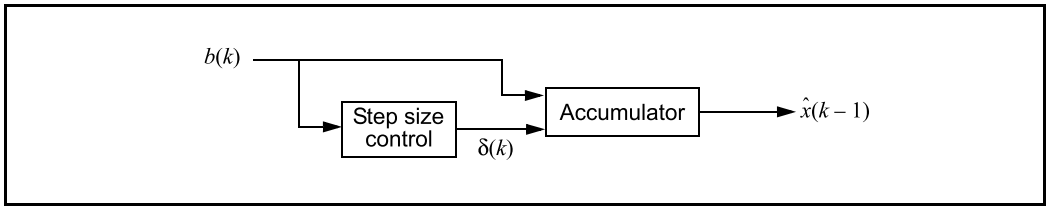


Figure 9.2: Block diagram of CVSD decoder with syllabic companding.

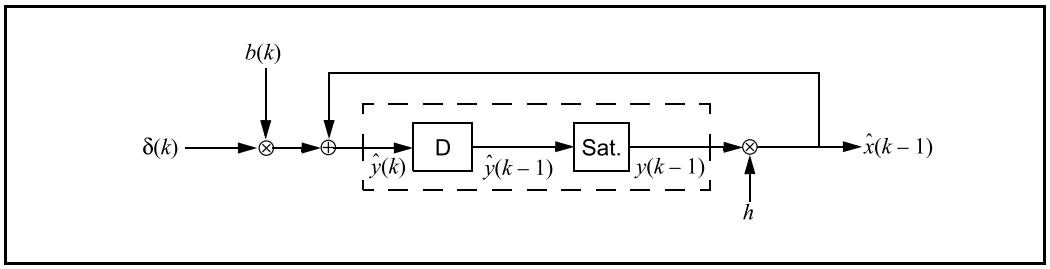


Figure 9.3: Accumulator procedure.

Let $\mathrm{sgn}(x) = 1$ for $x \geq 0$, otherwise $\mathrm{sgn}(x) = -1$. On air these numbers shall be represented by the sign bit; i.e. negative numbers are mapped on “1” and positive numbers are mapped on “0”.

Denote the CVSD encoder output bit $b(k)$, the encoder input $x(k)$, the accumulator contents $y(k)$, and the step size $\delta(k)$. Furthermore, let *h* be the decay factor for the accumulator, let β denote the decay factor for the step size, and, let α be the syllabic companding parameter. The latter parameter monitors the slope by considering the *K* most recent output bits

Let

$$\hat{x}(k) = hy(k). \qquad \text{(EQ 12)}$$

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

Petitioner Kiosoft Exhibit 1011 Part 1
Page 332

$$b(k) = \mathrm{sgn}\{x(k) - \hat{x}(k-1)\}, \quad \text{(EQ 13)}$$

$$\alpha = \begin{cases} 1, & \text{if } J \text{ bits in the last } K \text{ output bits are equal,} \\ 0, & \text{otherwise,} \end{cases} \quad \text{(EQ 14)}$$

$$\delta(k) = \begin{cases} \min\{\delta(k-1) + \delta_{min}, \delta_{max}\}, & \alpha = 1, \\ \max\{\beta\delta(k-1), \delta_{min}\}, & \alpha = 0, \end{cases} \quad \text{(EQ 15)}$$

$$y(k) = \begin{cases} \min\{\hat{y}(k), y_{max}\}, & \hat{y}(k) \geq 0. \\ \max\{\hat{y}(k), y_{min}\}, & \hat{y}(k) < 0. \end{cases} \quad \text{(EQ 16)}$$

where

$$\hat{y}(k) = \hat{x}(k-1) + b(k)\delta(k). \quad \text{(EQ 17)}$$

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

For a 64 kb/s CVSD, the parameters as shown in Table 9.2 shall be used. The numbers are based on a 16 bit signed number output from the accumulator. These values result in a time constant of 0.5 ms for the accumulator decay, and a time constant of 16 ms for the step size decay

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

Table 9.2: CVSD parameter values. The values are based on a 16 bit signed number output from the accumulator.

9.3 ERROR HANDLING

In the **DV**, **HV3**,**EV3** and **EV5** packets, the voice is not protected by FEC. The quality of the voice in an error-prone environment then depends on the robustness of the voice coding scheme and, in the case of eSCO, the retransmission scheme. CVSD, in particular, is rather insensitive to random bit errors, which are experienced as white background noise. However, when a packet is rejected because either the channel access code, the HEC test was unsuccessful, or the CRC has failed, measures have to be taken to “fill” in the lost speech segment.

The voice payload in the **HV2** and **EV4** packets are protected by a 2/3 rate FEC. For errors that are detected but cannot be corrected, the receiver should try to minimize the audible effects. For instance, from the 15-bit FEC segment with uncorrected errors, the 10-bit information part as found before the FEC decoder should be used. The **HV1** packet is protected by a 3 bit repetition FEC. For this code, the decoding scheme will always assume zero or one-bit errors. Thus, there exist no detectable but uncorrectable error events for **HV1** packets.

9.4 GENERAL AUDIO REQUIREMENTS

9.4.1 Signal levels

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

Full swing at the 16 bit linear PCM interface to the CVSD encoder is defined to be 3 dBm0.

9.4.2 CVSD audio quality

For Bluetooth audio quality the requirements are put on the transmitter side. The 64 ksamples/s linear PCM input signal should have negligible spectral power density above 4 kHz. The power spectral density in the 4-32 kHz band of the decoded signal at the 64 ksample/s linear PCM output, should be more than 20 dB below the maximum in the 0-4 kHz range.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 334

10 LIST OF FIGURES

Figure 1.1: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c). 53
Figure 1.2: Standard packet format. 53
Figure 1.3: Bluetooth clock. 54
Figure 1.4: Format of BD_ADDR. 55

Figure 2.1: Multi-slot packets 59
Figure 2.2: Derivation of CLK in master (a) and in slave (b). 60
Figure 2.3: RX/TX cycle of master transceiver in normal mode for single-slot packets. 61
Figure 2.4: RX/TX cycle of slave transceiver in normal mode for single-slot packets. 62
Figure 2.5: RX timing of slave returning from hold mode. 63
Figure 2.6: Derivation of CLKE. 64
Figure 2.7: RX/TX cycle of transceiver in PAGE mode. 65
Figure 2.8: Timing of page response packets on successful page in first half slot 66
Figure 2.9: Timing of page response packets on successful page in second half slot 67
Figure 2.10: Timing of inquiry response packet on successful inquiry in first half slot 69
Figure 2.11: Timing of inquiry response packet on successful inquiry in second half slot 69
Figure 2.12: General block diagram of hop selection scheme. 71
Figure 2.13: Hop selection scheme in CONNECTION state. 72
Figure 2.14: Single- and multi-slot packets. 73
Figure 2.15: Example of the same channel mechanism. 73
Figure 2.16: Block diagram of the basic hop selection kernel for the hop system. 74
Figure 2.17: XOR operation for the hop system. 75
Figure 2.18: Permutation operation for the hop system. 76
Figure 2.19: Butterfly implementation. 76
Figure 2.20: Block diagram of adaptive hop selection mechanism 78

Figure 4.1: Functional diagram of TX buffering. 87
Figure 4.2: Functional diagram of RX buffering 91

Figure 6.1: General packet format. 97
Figure 6.2: Access code format 98
Figure 6.3: Preamble 99
Figure 6.4: Construction of the sync word. 100

Figure 6.5: LFSR and the starting state to generate 102
Figure 6.6: Trailer in CAC when MSB of sync word is 0 (a), and when MSB of sync word is 1 (b). 102
Figure 6.7: Header format. 103
Figure 6.8: Format of the FHS payload. 107
Figure 6.9: DV packet format 109
Figure 6.10: Payload header format for single-slot ACL packets. 113
Figure 6.11: Payload header format for multi-slot ACL packets. 113

Figure 7.1: Header bit processes. 117
Figure 7.2: Payload bit processes. 117
Figure 7.3: The LFSR circuit generating the HEC. 118
Figure 7.4: Initial state of the HEC generating circuit. 119
Figure 7.5: HEC generation and checking. 119
Figure 7.6: The LFSR circuit generating the CRC. 120
Figure 7.7: Initial state of the CRC generating circuit. 120
Figure 7.8: CRC generation and checking. 120
Figure 7.9: Data whitening LFSR. 121
Figure 7.10: Bit-repetition encoding scheme. 122
Figure 7.11: LFSR generating the (15,10) shortened Hamming code. 123
Figure 7.12: Stage 1 of the receive protocol for determining the ARQN bit. 125
Figure 7.13: Stage 2 (ACL) of the receive protocol for determining the ARQN bit. 126
Figure 7.14: Stage 2 (eSCO) of the receive protocol for determining the ARQN bit. 127
Figure 7.15: Transmit filtering for packets with CRC. 128
Figure 7.16: Broadcast repetition scheme 131

Figure 8.1: State diagram of link controller. 133
Figure 8.2: Conventional page (a), page while one synchronous link present (b), page while two synchronous links present (c). 138
Figure 8.3: Messaging at initial connection when slave responds to first page message. 140
Figure 8.4: Messaging at initial connection when slave responds to second page message. 140
Figure 8.5: Connection state. 148
Figure 8.6: RX/TX timing in multi-slave configuration 149
Figure 8.7: eSCO Window Details for Single-Slot Packets 153
Figure 8.8: eSCO Window Details for Asymmetric Traffic 153
Figure 8.9: Successful hop sequence switching 158
Figure 8.10: Recovery hop sequences 159
Figure 8.11: AFH_Instant changes during multi-slot packets transmitted by the master. 160

Petitioner Kiosoft Exhibit 1011 Part 1
Page 336

Figure 8.12: Sniff anchor points 163
Figure 8.13: General beacon train format 167
Figure 8.14: Definition of access window 168
Figure 8.15: Access procedure applying the polling technique. 168
Figure 8.16: Disturbance of access window by SCO traffic 169
Figure 8.17: Extended sleep interval of parked slaves. 170

Figure 9.1: Block diagram of CVSD encoder with syllabic companding. ...174
Figure 9.2: Block diagram of CVSD decoder with syllabic companding. ...174
Figure 9.3: Accumulator procedure. 174

Appendix A: General Audio Requirements

Figure SLR measurement set-up. 183
Figure RLR measurement set-up. 183
Figure Plot of recommended frequency mask for Bluetooth. The GSM send frequency mask is given for comparison (dotted line) 184

Appendix C: Recommendations for AFH Operation in Park, Hold and Sniff

Figure Timing constraint on AFH_Instant with slaves in park, hold and sniff 188

Petitioner Kiosoft Exhibit 1011 Part 1
Page 338

11 LIST OF TABLES

Table 2.1: Control of the butterflies for the hop system 75
Table 2.2: Control for hop system. 79

Table 6.1: Summary of access code types. 98
Table 6.2: Packets defined for synchronous and asynchronous logical transport types. 105
Table 6.3: Description of the FHS payload 107
Table 6.4: Contents of SR field 108
Table 6.5: Contents of page scan mode field.....108
Table 6.6: Logical link LLID field contents.....113
Table 6.7: Use of payload header flow bit on the logical links. 115
Table 6.8: Link control packets 116
Table 6.9: ACL packets.....116
Table 6.10: Synchronous packets.....116

Table 8.1: Relationship between scan interval, and paging modes R0, R1 and R2. 135
Table 8.2: Relationship between train repetition, and paging modes R0, R1 and R2 when synchronous links are present.....138
Table 8.3: Initial messaging during start-up. 139
Table 8.4: Increase of train repetition when synchronous links are present. 146
Table 8.5: Messaging during inquiry routines. 147
Table 8.6: Channel classification descriptions 161

Table 9.1: Voice coding schemes supported on the air interface.....173
Table 9.2: CVSD parameter values. The values are based on a 16 bit signed number output from the accumulator.....175

Appendix A: General Audio Recommendations
Table Recommended Frequency Mask for Bluetooth.....184

APPENDIX A: GENERAL AUDIO RECOMMENDATIONS

MAXIMUM SOUND PRESSURE

It is the sole responsibility of each manufacturer to design their audio products in a safe way with regards to injury to the human ear. The Bluetooth Specification doesn't specify maximum sound pressure from an audio device.

OTHER TELEPHONY NETWORK REQUIREMENTS

It is the sole responsibility of each manufacturer to design the Bluetooth audio product so that it meets the regulatory requirements of all telephony networks that it may be connected to.

AUDIO LEVELS FOR BLUETOOTH

Audio levels shall be calculated as Send Loudness Rating, SLR, and Receive Loudness Rating, RLR. The calculation methods are specified in ITU-T Recommendation P.79.

The physical test set-up for Handsets and Headsets is described in ITU-T Recommendation P.51 and P.57

The physical test set-up for speakerphones and "Vehicle handsfree systems" is specified in ITU-T Recommendation P.34.

A general equation for computation of loudness rating (LR) for telephone sets is given by ITU-T recommendations P.79 and is given by

$$LR = -\frac{10}{m}\log_{10}\left(\sum_{i=N_1}^{N_2} 10^{m(s_i - w_i)/10}\right), \qquad \text{(EQ 18)}$$

where

m is a constant (~ 0.2).

w_i = weighting coefficient (different for the various LRs).

S_i = the sensitivity at frequency F_i, of the electro-acoustic path

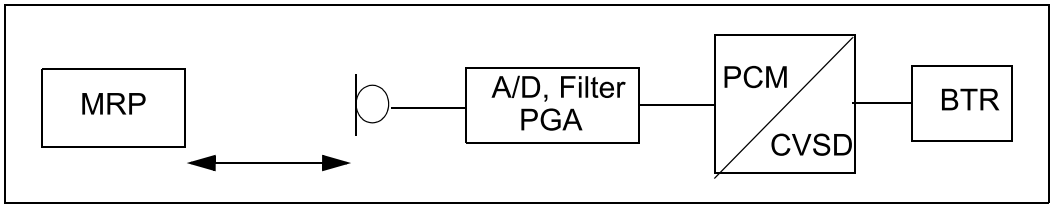
N_1,N_2, consecutive filter bank numbers (Art. Index: 200-4000 Hz)

(EQ 18) on page 182 is used for calculating the (SLR) according to Figure on page 183, and (RLR) according to Figure on page 183. When calculating LRs one must only include those parts of the frequency band where an actual signal transmission can occur in order to ensure that the additive property of LRs is retained. Therefore ITU-T P.79 uses only the frequency band 200-4000 Hz in LR computations.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 340

MICROPHONE PATH

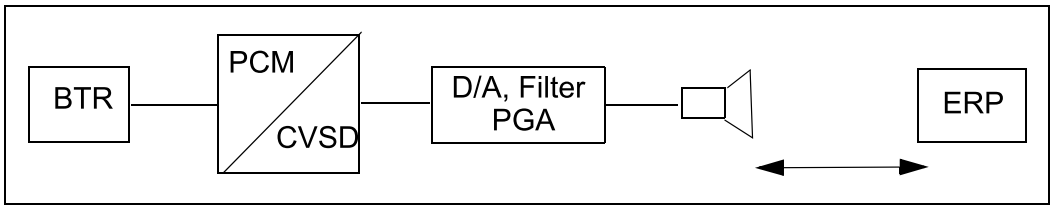
SLR measurement model



SLR measurement set-up.

LOUDSPEAKER PATH

RLR measurement model



RLR measurement set-up.

BLUETOOTH VOICE INTERFACE

The specification for the Bluetooth voice interface should follow in the first place the *ITU-T Recommendations P.79*, which specifies the loudness ratings for telephone sets. These recommendations give general guidelines and specific algorithms used for calculating the loudness ratings of the audio signal with respect to Ear Reference Point (ERP).

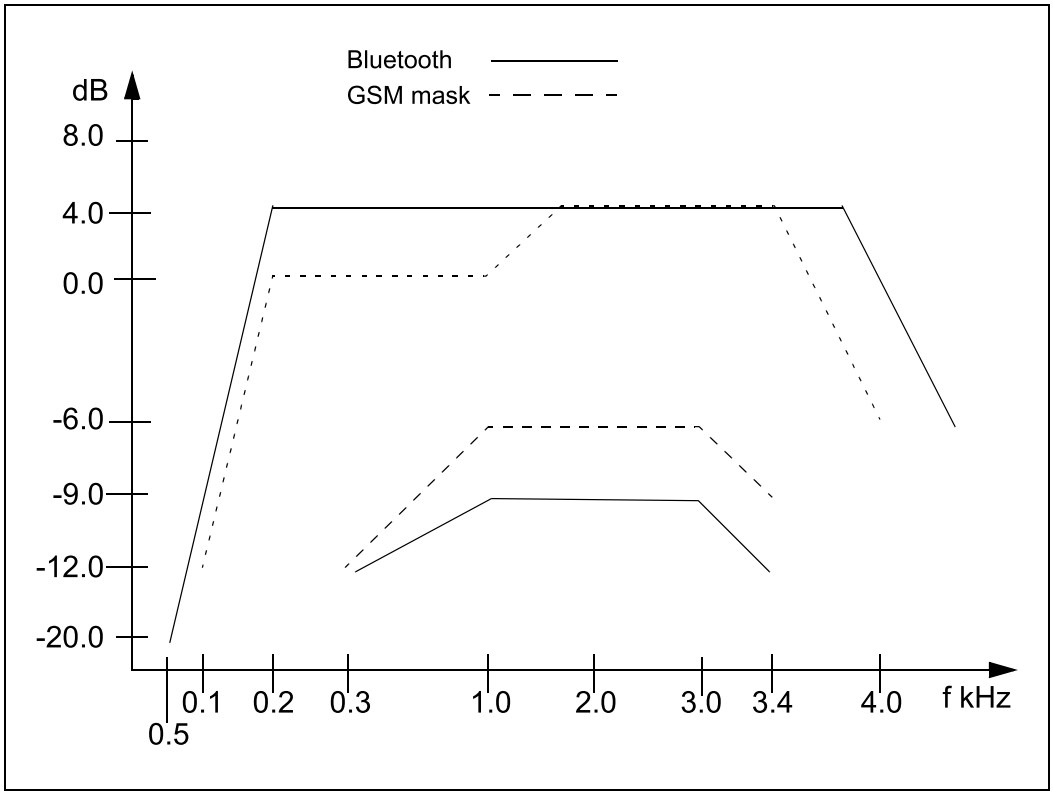
For Bluetooth voice interface to the different cellular system terminals, loudness and frequency recommendations based on the cellular standards should be used. For example, GSM 03.50 gives recommendation for both the loudness ratings and frequency mask for a GSM terminal interconnection with Bluetooth.

1- The output of the CVSD decoder are 16-bit linear PCM digital samples, at a sampling frequency of 8 ksample/second. Bluetooth also supports 8-bit log PCM samples of A-law and μ-law type. The sound pressure at the ear reference point for a given 16-bit CVSD sample, should follow the sound pressure level given in the cellular standard specification.

2- A maximum sound pressure which can be represented by a 16-bit linear PCM sample at the output of the CVSD decoder should be specified according to the loudness rating, in ITU P.79 and at PGA value of 0 dB. Programmable Gain Amplifiers (PGAs) are used to control the audio level at the terminals by the user. For conversion between various PCM representations: A-law, μ-law and linear PCM, ITU-T G.711, G.712, G.714 give guidelines and PCM value relationships. Zero-code suppression based on ITU-T G.711 is also recommended to avoid network mismatches.

FREQUENCY MASK

For interfacing a Bluetooth terminal to a digital cellular mobile terminal, a compliance of the CVSD decoder signal to the frequency mask given in the cellular standard, is recommended to guarantee correct function of the speech coders. A recommendation for a frequency mask is given in the Table below. The Figure below shows a plot of the frequency mask for Bluetooth (solid line). The GSM frequency mask (dotted line) is shown in Figure for comparison.



Plot of recommended frequency mask for Bluetooth. The GSM send frequency mask is given for comparison (dotted line)

Frequency (Hz)	Upper Limit (dB)	Lower Limit (dB)
50	-20	-
300	4	-12
1000	4	-9
2000	4	-9
3000	4	--9
3400	4	-12
4000	0	-

Recommended Frequency Mask for Bluetooth

Petitioner Kiosoft Exhibit 1011 Part 1
Page 342

APPENDIX B: TIMERS

This appendix contains a list of Baseband timers related to inactivity timeout defined in this specification. Definitions and default values of the timers are listed below

All timer values are given in slots.

LIST OF TIMERS

inquiryTO

The *inquiryTO* defines the number of slots the **inquiry** substate will last. The timer value may be changed by the host. HCI provides a a command to change the timer value.

pageTO

The *pageTO* defines the number of slots the **page** substate can last before a response is received. The timer value may be changed by the host. HCI provides a a command to change the timer value.

pagerespTO

In the slave, it defines the number of slots the slave awaits the master's response, FHS packet, after sending the page acknowledgment ID packet. In the master, *pagerespTO* defines the number of slots the master should wait for the FHS packet acknowledgment before returning to **page** substate. Both master and slave units should use the same value for this timeout, to ensure common page/scan intervals after reaching *pagerespTO*.

The *pagerespTO*value is 8 slots.

newconnectionTO

Every time a new connection is started through paging, scanning, role switch or unparking, the master sends a POLL packet as the first packet in the new connection. Transmission and acknowledgment of this POLL packet is used to confirm the new connection. If the POLL packet is not received by the slave or the response packet is not received by the master for *newconnectionTO* number of slots, both the master and the slave will return to the previous substate.

newconnectionTO value is 32 slots.

supervisionTO

The *supervisionTO* is used by both the master and slave to monitor link loss. Ifa device does not receive any packets that pass the HEC check and have theproper LT_ADDR for a period of *supervisionTO*, it will reset the link. The supervision timer keeps running in hold mode, sniff mode and park state.

The *supervisionTO* value may be changed by the host. HCI provides a a command to change the timer value. At the baseband level a default value that is equivalent to 20 seconds will be used.

APPENDIX C: RECOMMENDATIONS FOR AFH OPERATION IN PARK, HOLD AND SNIFF

The three possible AFH operation modes for an AFH capable slave in park, hold and sniff are the same three AFH operation modes used during **CONNECTION** state:

- *Enabled* (using the same AHS as slaves in the **CONNECTION** state)
- *Enabled* (using AHS(79))
- *Disabled*

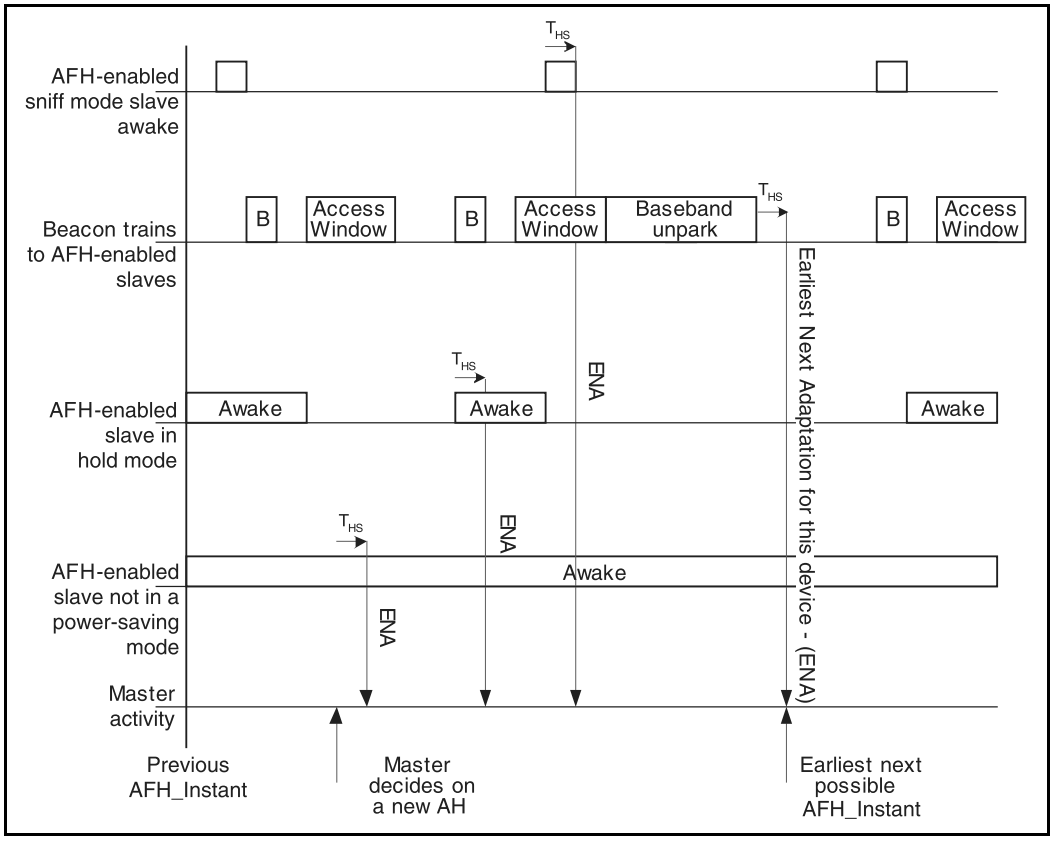
The master may place an AFH capable slave in any of the three AFH operating modes.

Operation at the Master

A master that has one or more slaves in park, hold or sniff and decides to update them simultaneously shall schedule an *AFH_Instant* for a time that allows it to update all these slaves (as well as its active slaves) with the new adaptation.

A master that has multiple slaves with non-overlapping “wake” times (e.g. slaves in sniff mode with different timing parameters) may keep them *enabled* on the same adaptation provided that its scheduling of the *AFH_Instant* allows enough time to update them all.

This timing is summarized in the figure below. In this example the master decides that a hop sequence adaptation is required. However it cannot schedule an *AFH_Instant* until it has informed its active slave, its slave in hold, its slave in sniff, and had time to un-park its parked slaves.



Timing constraint on AFH_Instant with slaves in park, hold and sniff

Operation in park

A slave that is in the Park state cannot send or receive any AFH LMP messages. Once the slave has left the Park state the master may subsequently update the slave's adaptation.

AFH Operation in Sniff

Once a slave has been placed in sniff mode, the master may periodically change its AHS without taking the slave out of sniff mode.

AFH Operation in Hold

A slave that is in hold mode cannot send or receive any LMP messages. Once the slave has left hold mode the master may subsequently update the slave's adaptation.

Core System Package [Controller volume]
Part C



LINK MANAGER PROTOCOL

This specification describes the Link Manager Protocol (LMP) which is used for link set-up and control. The signals are interpreted and filtered out by the Link Manager on the receiving side and are not propagated to higher layers.

CONTENTS

1 Introduction 195

2 General Rules 197

2.1 Message Transport 197
2.2 Synchronization 197
2.3 Packet Format 198
2.4 Transactions 199
2.4.1 LMP Response Timeout 200
2.5 Error Handling 200
2.5.1 Transaction collision resolution 201
2.6 Procedure Rules 201
2.7 General Response Messages 202
2.8 Specification Writing Policies 202

3 Device Features 203

3.1 General Description 203
3.2 Feature Definitions 203
3.3 Feature Mask Definition 207
3.4 Link Manager Interoperability policy 208

4 Procedure Rules 209

4.1 Connection Control 209
4.1.1 Connection establishment 209
4.1.2 Detach 210
4.1.3 Power control 211
4.1.4 Adaptive frequency hopping 213
4.1.4.1 Master enables AFH 214
4.1.4.2 Master disables AFH 214
4.1.4.3 Master updates AFH 215
4.1.4.4 AFH operation in park, hold and sniff modes 215
4.1.5 Channel classification 216
4.1.5.1 Channel classification reporting enabling and disabling 217
4.1.6 Link supervision 218
4.1.7 Channel quality driven data rate change (CQDDR) 219
4.1.8 Quality of service (QoS) 220
4.1.8.1 Master notifies slave of the quality of service 220
4.1.8.2 Device requests new quality of service 221

4.1.9 Paging scheme parameters 222
4.1.9.1 Page mode 222
4.1.9.2 Page scan mode 222
4.1.10 Control of multi-slot packets 223
4.2 Security 224
4.2.1 Authentication 224
4.2.1.1 Claimant has link key 224
4.2.1.2 Claimant has no link key 225
4.2.1.3 Repeated attempts 225
4.2.2 Pairing 226
4.2.2.1 Responder accepts pairing 226
4.2.2.2 Responder has a fixed PIN 227
4.2.2.3 Responder rejects pairing 227
4.2.2.4 Creation of the link key 228
4.2.2.5 Repeated attempts 228
4.2.3 Change link key 229
4.2.4 Change current link key type 230
4.2.4.1 Change to a temporary link key 230
4.2.4.2 Make the semi-permanent link key the current link key 231
4.2.5 Encryption 232
4.2.5.1 Encryption mode 232
4.2.5.2 Encryption key size 233
4.2.5.3 Start encryption 234
4.2.5.4 Stop encryption 235
4.2.5.5 Change encryption mode, key or random number 235
4.2.6 Request supported encryption key size 236
4.3 Informational Requests 237
4.3.1 Timing accuracy 237
4.3.2 Clock offset 238
4.3.3 LMP version 238
4.3.4 Supported features 239
4.3.5 Name request 241
4.4 Role Switch 242
4.4.1 Slot offset 242
4.4.2 Role switch 243
4.5 Modes of Operation 245
4.5.1 Hold mode 245
4.5.1.1 Master forces hold mode 245
4.5.1.2 Slave forces hold mode 246
4.5.1.3 Master or slave requests hold mode 246

4.5.2 Park state 247
4.5.2.1 Master requests slave to enter park state 249
4.5.2.2 Slave requests to enter park state 250
4.5.2.3 Master sets up broadcast scan window 251
4.5.2.4 Master modifies beacon parameters....252
4.5.2.5 Unparking slaves 252
4.5.3 Sniff mode 253
4.5.3.1 Master or slave requests sniff mode 254
4.5.3.2 Moving a slave from sniff mode to active mode .. 255
4.6 Logical Transports....256
4.6.1 SCO logical transport 256
4.6.1.1 Master initiates an SCO link....256
4.6.1.2 Slave initiates an SCO link....257
4.6.1.3 Master requests change of SCO parameters258
4.6.1.4 Slave requests change of SCO parameters .258
4.6.1.5 Remove an SCO link 258
4.6.2 eSCO logical transport 259
4.6.2.1 Master initiates an eSCO link....259
4.6.2.2 Slave initiates an eSCO link....260
4.6.2.3 Master or slave requests change of eSCO parameters261
4.6.2.4 Remove an eSCO link 261
4.6.2.5 Rules for the LMP negotiation and renegotiation 262
4.6.2.6 Negotiation state definitions....263
4.7 Test Mode 264
4.7.1 Activation and deactivation of test mode....264
4.7.2 Control of test mode 265
4.7.3 Summary of test mode PDUs....266

5 Summary 269
5.1 PDU Summary 269
5.2 Parameter Definitions 277
5.3 Default Values 284

6 List of Figures....285

7 List of Tables 289

1 INTRODUCTION

The Link Manager Protocol (LMP) is used to control and negotiate all aspects of the operation of the Bluetooth connection between two devices. This includes the set-up and control of logical transports and logical links, and for control of physical links.

The Link Manager Protocol is used to communicate between the Link Managers (LM) on the two devices which are connected by the ACL logical transport. All LMP messages shall apply solely to the physical link and associated logical links and logical transports between the sending and receiving devices.

The protocol is made up of a series of messages which shall be transferred over the ACL-C logical link on the default ACL logical transport between two devices. LMP messages shall be interpreted and acted-upon by the LM and shall not be directly propagated to higher protocol layers.

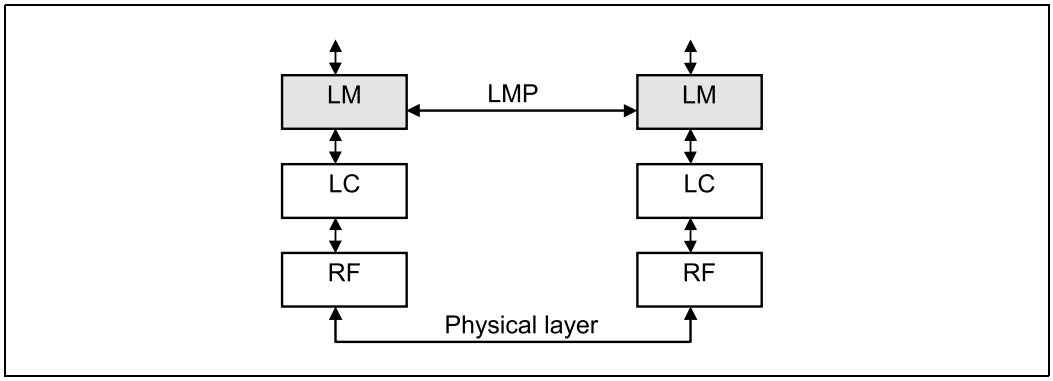


Figure 1.1: Link Manager Protocol signalling layer.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 354

2 GENERAL RULES

2.1 MESSAGE TRANSPORT

LMP messages shall be exchanged over the ACL-C logical link that is carried on the default ACL logical transport (Baseband Specification, Section 4.4, on page 86). The ACL-C logical link is distinguished from the ACL-U (which carries L2CAP and user data) by the Logical Link Identifier (LLID) field carried in the payload header of variable-length packets (Baseband Specification, Section 6.6.2, on page 113).

The ACL-C has a higher priority than other traffic - see Baseband Specification, Section 5.6, on page 96.

The error detection and correction capabilities of the baseband ACL logical transport are generally sufficient for the requirements of the LMP. Therefore LMP messages do not contain any additional error detection information beyond what can be realized by means of sanity checks performed on the contents of LMP messages. Any such checks and protections to overcome undetected errors in LMP messages is an implementation matter.

2.2 SYNCHRONIZATION

This section is informative and explains why many of the LMP message sequences are defined as they are.

LMP messages are carried on the ACL-C logical link, which does not guarantee a time to deliver or acknowledge packets. LMP procedures take account of this when synchronizing state changes in the two devices. For example, criteria are defined that specify when a logical transport address (LT_ADDR) may be re-used after it becomes available due to a device leaving the piconet or entering the park state. Other LMP procedures, such as hold or role switch include the Bluetooth clock as a parameter in order to define a fixed synchronization point. The transitions into and out of sniff mode are protected with a transition mode.

The LC normally attempts to communicate with each slave no less often than every T_{poll} slots (see Section 4.1.8 on page 220) based on the T_{poll} for that slave.

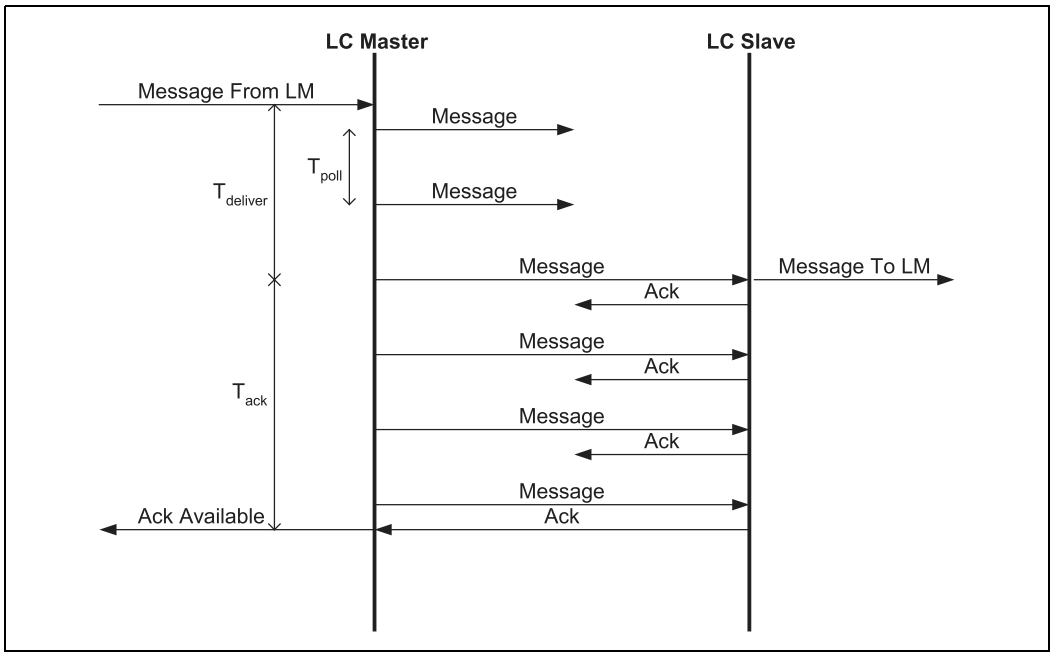


Figure 2.1: Transmission of a message from master to slave[1].

Figure 2.1 on page 198 illustrates the fundamental problem. It shows the transmission of a packet from the master to the slave in conditions of heavy interference for illustrative purposes. It is obvious that neither side can determine the value of either $T_{deliver}$ or T_{ack}. It is therefore not possible to use simple messages to identify uniquely the instant at which a state change occurs in the other device.

2.3 PACKET FORMAT

Each PDU is assigned either a 7 or a 15 bit opcode used to uniquely identify different types of PDUs, see Table 5.1 on page 269. The first 7 bits of the opcode and a transaction ID are located in the first byte of the payload body. If the initial 7 bits of the opcode have one of the special escape values 124-127 then an additional byte of opcode is located in the second byte of the payload and the combination uniquely identifies the PDU.

The FLOW bit in the packet header is always 1 and is ignored on reception.

If the PDU contains one or more parameters these are placed in the payload starting immediately after the opcode, i.e. at byte 2 if the PDU has a 7 bit opcode or byte 3 if the PDU has a 15 bit opcode. The number of bytes used

1. Note the diagram shows the limiting case where the master transmits the message at intervals of T_{poll}. In the case of heavy interference improved performance is gained by transmitting more often.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 356

depends on the length of the parameters. All parameters have a little-endian format, i.e. the least significant byte is transferred first.

LMP messages shall be transmitted using DM1 packets, however if an HV1 SCO link is in use and the length of the payload is less than 9 bytes then DV packets may be used.

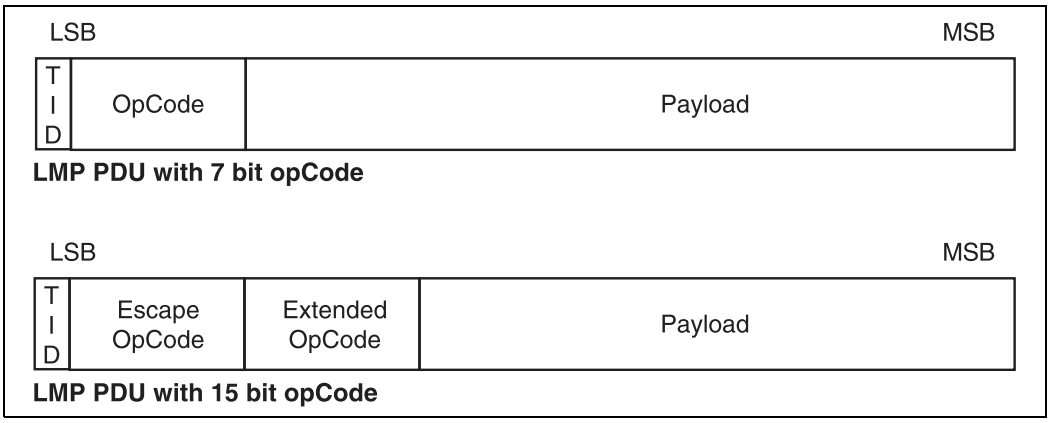


Figure 2.2: Payload body when LMP PDUs are sent.

2.4 TRANSACTIONS

The LMP operates in terms of transactions. A transaction is a connected set of message exchanges which achieve a particular purpose. All PDUs which form part of the same transaction shall have the same value for the transaction ID which is stored as part of the first byte of the opCode (see Section 2.3 on page 198).

The transaction ID is in the least significant bit. It shall be 0 if the PDU forms part of a transaction that was initiated by the master and 1 if the transaction was initiated by the slave.

Each sequence described in Section 4 on page 209 shall be defined as a transaction. For pairing, see Section 4.2.2 on page 226, and encryption, see Section 4.2.5 on page 232, all sequences belonging to each section shall be counted as one transaction and shall use the same transaction ID. For connection establishment, see Section 4.1.1 on page 209, LMP_host_connection_req and the response with LMP_accepted or LMP_not_accepted shall form one transaction and have the transaction ID of 0. LMP_setup_complete is a stand-alone PDU, which forms a transaction by itself.

For error handling, see Section 2.5 on page 200, the PDU to be rejected and LMP_not_accepted or LMP_not_accepted_ext shall form a single transaction.

2.4.1 LMP Response Timeout

The time between receiving a baseband packet carrying an LMP PDU and sending a baseband packet carrying a valid response PDU, according to the procedure rules in Section 4 on page 209, shall be less than the LMP Response Timeout. The value of this timeout is 30 seconds. Note that the LMP Response Timeout is applied not only to sequences described in Section 4 on page 209, but also to the series of the sequences defined as the transactions in Section 4 on page 209. It shall also be applied to the series of LMP transactions that take place during a period when traffic on the ACL-U logical link is disabled for the duration of the series of LMP transactions, for example during the enabling of encryption.

The LMP Response Timeout shall restart each time an LMP PDU which requires a reply is queued for transmission by the baseband.

2.5 ERROR HANDLING

If the LM receives a PDU with unrecognized opcode, it shall respond with LMP_not_accepted or LMP_not_accepted_ext with the error code *unknown LMP PDU.* The opcode parameter that is echoed back is the unrecognized opcode.

If the LM receives a PDU with invalid parameters, it shall respond with LMP_not_accepted or LMP_not_accepted_ext with the error code *invalid LMP parameters*.

If the maximum response time, see Section 2.4 on page 199, is exceeded or if a link loss is detected (see Baseband Specification, Section 3.1, on page 83), the party that waits for the response shall conclude that the procedure has terminated unsuccessfully.

Erroneous LMP messages can be caused by errors on the channel or systematic errors at the transmit side. To detect the latter case, the LM should monitor the number of erroneous messages and disconnect if it exceeds a threshold, which is implementation-dependent.

When the LM receives a PDU that is not allowed, and the PDU normally expects a PDU reply, for example LMP_host_connection_req or LMP_unit_key, the LM shall return PDU LMP_not_accepted or LMP_not_accepted_ext with the error code *PDU not allowed*. If the PDU normally doesn't expect a reply, for example LMP_sres or LMP_temp_key, the PDU will be ignored.

The reception of an optional PDU which is not supported shall be handled in one of two ways: if the LM simply does not know the opcode (e.g. it was added at a later version of the specification) it shall respond with LMP_not_accepted or LMP_not_accepted_ext with the error code *unknown LMP PDU*. If the LM recognises the PDU as optional but unsupported then it shall reply with LMP_not_accepted or LMP_not_accepted_ext with the error code *unsupported LMP feature* if the PDU would normally generate a reply otherwise no reply is generated.

2.5.1 Transaction collision resolution

Since LMP PDUs are not interpreted in real time, collision situations can occur where both LMs initiate the same procedure and both cannot be completed. In this situation, the master shall reject the slave-initiated procedure by sending LMP_not_accepted or LMP_not_accepted_ext with the error code *LMP error transaction collision*. The master-initiated procedure shall then be completed.

Collision situations can also occur where both LMs initiate different procedures and both cannot be completed. In this situation, the master shall reject the slave-initiated procedure by sending LMP_not_accepted or LMP_not_accepted_ext with the error code *LMP error different transaction collision*. The master- initiated procedure shall then be completed.

2.6 PROCEDURE RULES

Each procedure is described and depicted with a sequence diagram. The following symbols are used in the sequence diagrams:

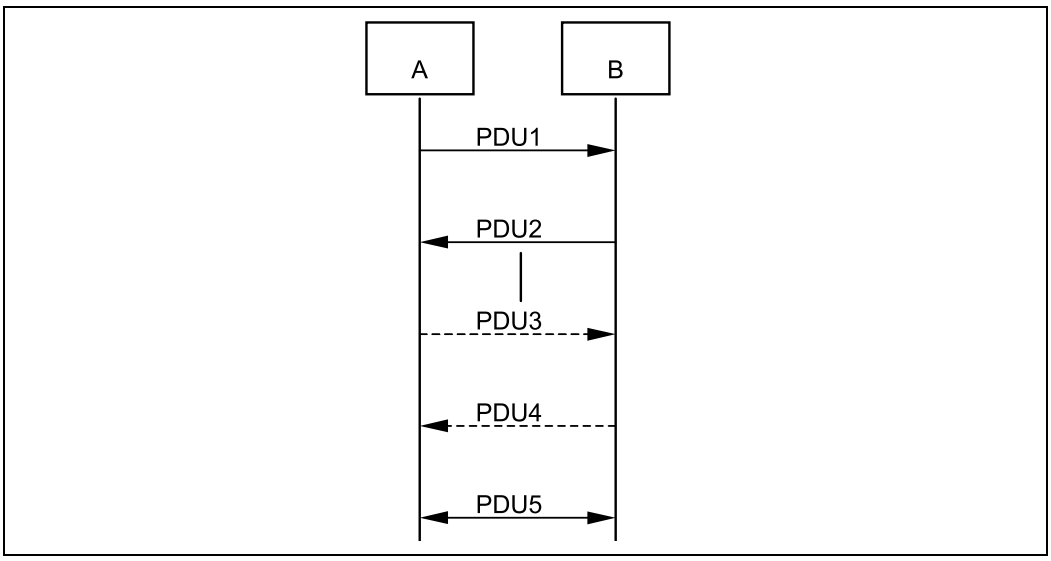


Figure 2.3: Symbols used in sequence diagrams.

PDU1 is a PDU sent from A to B. PDU2 is a PDU sent from B to A. PDU3 is a PDU that is optionally sent from A to B. PDU4 is a PDU that is optionally sent from B to A. PDU5 is a PDU sent from either A or B. A vertical line indicates that more PDUs can optionally be sent.

2.7 GENERAL RESPONSE MESSAGES

The PDUs LMP_accepted, LMP_accepted_ext, LMP_not_accepted and LMP_not_accepted_ext are used as response messages to other PDUs in a number of different procedures. LMP_accepted or LMP_accepted_ext includes the opcode of the message which is accepted. LMP_not_accepted or LMP_not_accepted_ext includes the opcode of the message which is not accepted and the error code why it is not accepted.

LMP_accepted_ext and LMP_not_accepted_ext shall be used when the opcode is 15 bits in length. LMP_accepted and LMP_not_accepted shall be used when the opcode is 7 bits in length.

M/O	PDU	Contents
M	LMP_accepted	op code
M	LMP_not_accepted	op code error code
O	LMP_accepted_ext	escape op code extended op code
O	LMP_not_accepted_ext	escape op code extended op code error code

Table 2.1: General response messages.

2.8 LMP MESSAGE CONSTRAINTS

This section is informative.

- No LMP message shall exceed the maximum payload length of a single DM1 packet i.e. 17 bytes in length (Baseband Specification, Section 6.5.4.1, on page 111).
- All LM messages are of fixed length apart from those sent using broadcast in park state.
- The LMP version number shall not be used to indicate the presence or absence of functionality.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 360

3 DEVICE FEATURES

3.1 GENERAL DESCRIPTION

Each PDU is either Mandatory or Optional as defined by the M/O field in the tables of Section 4 on page 209. An M in this field shall indicate that support for the feature is mandatory. An O in this field shall indicate that support for the PDU is optional and it shall be followed by the number(s) of the feature(s) involved in brackets.

All features added after the 1.1 specification have associated LMP feature bits. Support of these features may be made “mandatory” by the qualification process but the LM still considers them to be optional since it must interoperate with older devices which do not support them.

The LM does not need to be able to transmit a PDU which is optional. Support of optional PDUs is indicated by a device's features mask. The features mask can be read (see Section 4.3.4 on page 239). An LM shall not send or be sent any PDU which is incompatible with the features signalled in its or its peer's features mask, as detailed in Section 3.2 .

3.2 FEATURE DEFINITIONS

Feature	Definition
Extended features	This feature indicates whether the device is able to support the extended features mask using the LMP sequences defined in Section 4.3.4 on page 239.
Timing accuracy	This feature indicates whether the LM supports requests for timing accuracy using the sequence defined in Section 4.3.1 on page 237.
Enhanced inquiry scan	This feature indicates whether the device is capable of supporting the enhanced inquiry scan mechanism as defined in Baseband Specification, Section 8.4.1, on page 144. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Interlaced inquiry scan	This feature indicates whether the device is capable of supporting the interlaced inquiry scan mechanism as defined in Baseband Specification, Section 8.4.1, on page 144. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Interlaced page scan	This feature indicates whether the device is capable of supporting the interlaced page scan mechanism as defined in Baseband Specification, Section 8.3.1, on page 134. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.

Table 3.1: Feature definitions.

Feature	Definition
RSSI with inquiry results	This feature indicates whether the device is capable of reporting the RSSI with inquiry results as defined in Baseband Specification, Section 8.4.2, on page 145. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Paging parameter negotiation	This feature indicates whether the LM is capable of supporting the signaling of changes in the paging scheme as defined in Section 4.1.9 on page 222.
3 slot packets	This feature indicates whether the device supports the transmission and reception of both DM3 and DH3 packets for the transport of traffic on the ACL-U logical link.
5 slot packets	This feature indicates whether the device supports the transmission and reception of both DM5 and DH5 packets for the transport of traffic on the ACL-U logical link.
Flow control lag	This is defined as the total amount of ACL-U data that can be sent following the receipt of a valid payload header with the payload header flow bit set to 0 and is in units of 256 bytes. See further in Baseband Specification, Section 6.6.2, on page 113.
AFH capable slave	This feature indicates whether the device is able to support the Adaptive Frequency Hopping mechanism as a slave as defined in Baseband Specification, Section 2.3, on page 63 using the LMP sequences defined in Section 4.1.4 on page 213.
AFH classification slave	This feature indicates whether the device is able to support the AFH classification mechanism as a slave as defined in Baseband Specification, Section 8.6.8, on page 161 using the LMP sequences defined Section 4.1.5 on page 216.
AFH capable master	This indicates whether the device is able to support the Adaptive Frequency Hopping mechanism as a master as defined in Baseband Specification, Section 2.3, on page 63 using the LMP sequences defined in Section 4.1.4 on page 213.
AFH classification master	This feature indicate whether the device is able to support the AFH classification mechanism as a master as defined in Baseband Specification, Section 8.6.8, on page 161 using the LMP sequences defined Section 4.1.5 on page 216.
Power control	This feature indicates whether the device is capable of adjusting its transmission power. This feature indicates the ability to receive the LMP_incr_power and LMP_decr_power PDUs and transmit the LMP_max_power and LMP_min_power PDUs, using the sequences defined in Section 4.1.3 on page 211. These sequences may be used even if the remote device does not support the power control feature, as long as it supports the Power control requests feature.

Table 3.1: Feature definitions.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 362

Feature	Definition
Power control requests	This feature indicates whether the device is capable of determining if the transmit power level of the other device should be adjusted and will send the LMP_incr_power and LMP_decr_power PDUs to request the adjustments. It indicates the ability to receive the LMP_max_power and LMP_min_power PDUs, using the sequences in Section 4.1.3 on page 211. These sequences may be used even if the remote device does not support the RSSI feature, as long as it supports the power control feature.
Channel Quality Driven Data Rate	This feature indicates whether the LM is capable of recommending a packet type (or types) depending on the channel quality using the LMP sequences defined in section Section 4.1.7 on page 219.
Broadcast encryption	This feature indicates whether the device is capable of supporting broadcast encryption as defined in [Part H] Section 4.2 on page 764 and also the sequences defined in Section 4.2.6 on page 236 and Section 4.2.4 on page 230. Note: Devices compliant to versions of this specification 1.1 and earlier may support broadcast encryption even though this feature bit is not set.
Encryption	This feature indicates whether the device supports the encryption of packet contents using the sequence defined in Section 4.2.5 on page 232.
Slot offset	This feature indicates whether the LM supports the transfer of the slot offset using the sequence defined in Section 4.4.1 on page 242.
Role switch	This feature indicates whether the device supports the change of master and slave roles as defined by baseband Section 8.6.5 on page 155 using the sequence defined in Section 4.4.2 on page 243.
Hold mode	This feature indicates whether the device is able to support Hold mode as defined in baseband Section 8.8 on page 165 using the LMP sequences defined in Section 4.5.1 on page 245.
Sniff mode	This feature indicates whether the device is able to support Sniff mode as defined in baseband Section 8.7 on page 163 using the LMP sequences defined in Section 4.5.3 on page 253.
Park state	This feature indicates whether the device is able to support Park state as defined in baseband Section 8.9 on page 165 using the LMP sequences defined in Section 4.5.2 on page 247.
SCO link	This feature shall indicate whether the device is able to support the SCO logical transport as defined in Baseband Specification, Section 4.3, on page 86, the HV1 packet defined in Baseband Specification, Section 6.5.2.1, on page 109 and receiving the DV packet defined in Baseband Specification, Section 6.5.2.4, on page 109 using the LMP sequence in Section 4.6.1 on page 256.
HV2 packets	This feature indicates whether the device is capable of supporting the HV2 packet type as defined in baseband Section 6.5.2.2 on page 109 on the SCO logical transport.

Table 3.1: Feature definitions.

Feature	Definition
HV3 packets	This feature indicates whether the device is capable of supporting the HV3 packet type as defined in baseband Section 6.5.2.3 on page 109 on the SCO logical transport.
µ-law log synchronous data	This feature indicates whether the device is capable of supporting µ-Law Log format data as defined in Baseband Specification, Section 9.1, on page 173 on the SCO and eSCO logical transports.
A-law log synchronous data	This feature indicates whether the device is capable of supporting A-Law Log format data as defined in Baseband Specification, Section 9.1, on page 173 on the SCO and eSCO logical transports.
CVSD synchronous data	This feature indicates whether the device is capable of supporting CVSD format data as defined in Baseband Specification, Section 9.2, on page 173 on the SCO and eSCO logical transports.
Transparent synchronous data	This feature indicates whether the device is capable of supporting transparent synchronous data as defined in Baseband Specification, Section 6.4.3, on page 104 on the SCO and eSCO logical transports.
Extended SCO link	This feature indicates whether the device is able to support the eSCO logical transport as defined Baseband Specification, Section 5.5, on page 96 and the EV3 packet defined in Baseband Specification, Section 6.5.3.1, on page 110 using the LMP sequences defined in Section 4.6.2 on page 259.
EV4 packets	This feature indicates whether the device is capable of supporting the EV4 packet type defined in Baseband Specification, Section 6.5.3.2, on page 110 on the eSCO logical transport.
EV5 packets	This feature indicates whether the device is capable of supporting the EV5 packet type defined in Baseband Specification, Section 6.5.3.3, on page 110 on the eSCO logical transport.

Table 3.1: Feature definitions.

3.3 FEATURE MASK DEFINITION

The features are represented as a bit mask when they are transferred in LMP messages. For each feature a single bit is specified which shall be set to 1 if the feature is supported and set to 0 otherwise. The single exception is the flow control lag which is coded as a 3 bit field with the least significant bit in byte 2 bit 4 and the most significant bit in byte 2 bit 6. All unknown or unassigned feature bits shall be set to 0.

No.	Supported feature	Byte	Bit
0	3 slot packets	0	0
1	5 slot packets	0	1
2	Encryption	0	2
3	Slot offset	0	3
4	Timing accuracy	0	4
5	Role switch	0	5
6	Hold mode	0	6
7	Sniff mode	0	7
8	Park state	1	0
9	Power control requests	1	1
10	Channel quality driven data rate (CQDDR)	1	2
11	SCO link	1	3
12	HV2 packets	1	4
13	HV3 packets	1	5
14	μ-law log synchronous data	1	6
15	A-law log synchronous data	1	7
16	CVSD synchronous data	2	0
17	Paging parameter negotiation	2	1
18	Power control	2	2
19	Transparent synchronous data	2	3
20	Flow control lag(least significant bit)	2	4
21	Flow control lag(middle bit)	2	5
22	Flow control lag(most significant bit)	2	6
23	Broadcast encryption	2	7
24	Reserved	3	0

Table 3.2: Feature mask definition

No.	Supported feature	Byte	Bit
25	Reserved	3	1
26	Reserved	3	2
27	Enhanced inquiry scan	3	3
28	Interlaced inquiry scan	3	4
29	Interlaced page scan	3	5
30	RSSI with inquiry results	3	6
31	Extended SCO link (EV3 packets)	3	7
32	EV4 packets	4	0
33	EV5 packets	4	1
34	Reserved	4	2
35	AFH capable slave	4	3
36	AFH classification slave	4	4
37	Reserved	4	5
38	Reserved	4	6
43	AFH capable master	5	3
44	AFH classification master	5	4
63	Extended features	7	7

Table 3.2: Feature mask definition

3.4 LINK MANAGER INTEROPERABILITY POLICY

Link managers of any version will interoperate using the lowest common subset of functionality by reading the LMP features mask (defined in Table 3.2 on page 207).

An optional LMP PDU shall only be sent to a device if the corresponding feature bit is set in its feature mask. The exception to this are certain PDUs (see Section 4.1.1 on page 209) which can be sent before the features mask is requested. Note: a later version device with a restricted feature set is indistinguishable from an earlier version device with the same features.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 366

4 PROCEDURE RULES

4.1 CONNECTION CONTROL

4.1.1 Connection establishment

After the paging procedure, LMP procedures with for clock offset request, LMP version, supported features, name request and detach may then be initiated.

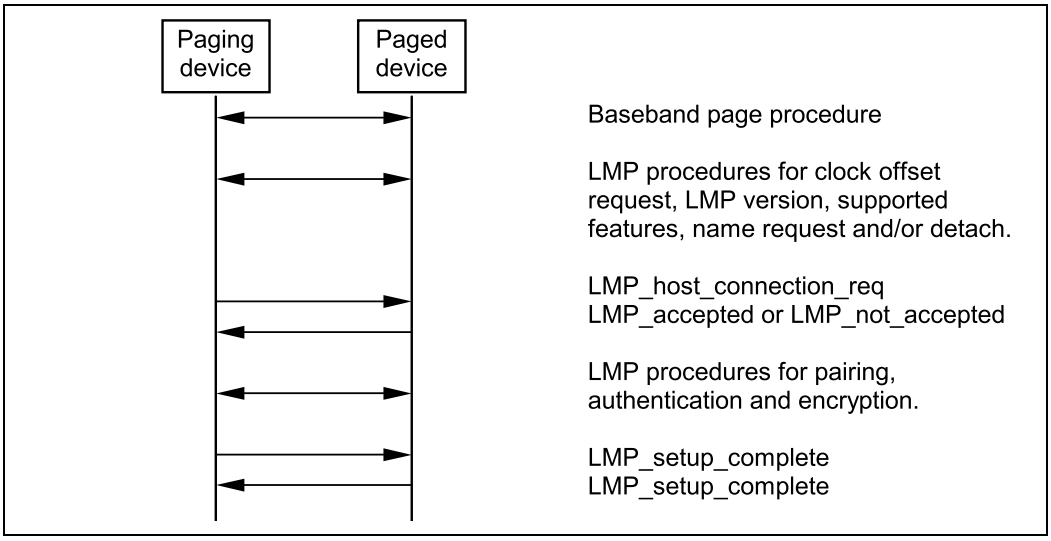


Figure 4.1: Connection establishment.

When the paging device wishes to create a connection involving layers above LM, it sends an LMP_host_connection_req PDU. When the other side receives this message, the host is informed about the incoming connection. The remote device can accept or reject the connection request by sending an LMP_accepted PDU or an LMP_not_accepted PDU. Alternatively, if the slave needs a role switch, see Section 4.4.2 on page 243, it sends an LMP_slot_offset PDU and LMP_switch_req PDU after it has received an LMP_host_connection_req PDU. If the role switch fails the LM shall continue with the creation of the connection unless this cannot be supported due to limited resources in which case the connection shall be terminated with an LMP_detach PDU with error code *other end terminated connection: low resources*. When the role switch has been successfully completed, the old slave will reply with an LMP_accepted PDU or an LMP_not_accepted PDU to the LMP_host_connection_req PDU (with the transaction ID set to 0).

If the paging device receives an LMP_not_accepted PDU in response to LMP_host_connection_req it shall immediately disconnect the link using the mechanism described in Section 4.1.2 on page 210.

If the LMP_host_connection_req PDU is accepted, LMP security procedures (pairing, authentication and encryption) may be invoked. When a device is not going to initiate any more security procedures during connection establishment it sends an LMP_setup_complete PDU. When both devices have sent LMP_setup_complete PDUs the traffic can be transferred on the ACL-U logical transport.

M/O	PDU	Contents
M	LMP_host_connection_req	-
M	LMP_setup_complete	-

Table 4.1: PDUs used for connection establishment.

4.1.2 Detach

The connection between two Bluetooth devices may be detached anytime by the master or the slave. An error code parameter is included in the message to inform the other party of why the connection is detached.

M/O	PDU	Contents
M	LMP_detach	error code

Table 4.2: PDU used for detach.

The initiating LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 96). The initiating LM then queues the LMP_detach for transmission and it shall start a timer for $6*T_{poll}$ slots where T_{poll} is the poll interval for the connection. If the initiating LM receives the baseband acknowledgement before the timer expires it starts a timer for $3*T_{poll}$ slots. When this timer expires, and if the initiating LM is the master, the LT_ADDR(s) may be re-used immediately. If the initial timer expires then the initiating LM drops the link and starts a timer for $T_{linksupervisiontimeout}$ slots after which the LT_ADDR(s) may be re-used if the initiating LM is the master.

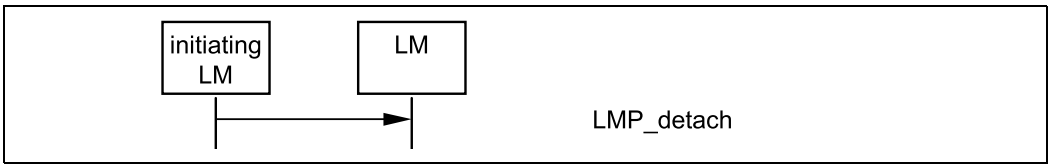
When the receiving LM receives the LMP_detach, it shall start a timer for $6*T_{poll}$ slots if it is the master and $3*T_{poll}$ if it is the slave. On timer expiration, the link shall be detached and, if the receiving LM is the master, the LT_ADDR(s) may be re-used immediately. If the receiver never receives the LMP_detach then a link supervision timeout will occur, the link will be detached, and the LT_ADDR may be re-used immediately.

If at any time during this or any other LMP sequence the Link supervision timeout expires then the link shall be terminated immediately and the LT_ADDR(S) may be re-used immediately.

If the connection is in hold mode, the initiating LM shall wait for hold mode to end before initiating the procedure defined above. If the connection is in sniff mode or park state, the initiating LM shall perform the procedure to exit sniff

Petitioner Kiosoft Exhibit 1011 Part 1
Page 368

mode or park state before initiating the procedure defined above. If the procedure to exit sniff mode or park state does not complete within the LMP response timeout (30 seconds) the procedure defined above shall be initiated anyway.



Sequence 1: Connection closed by sending LMP_detach.

4.1.3 Power control

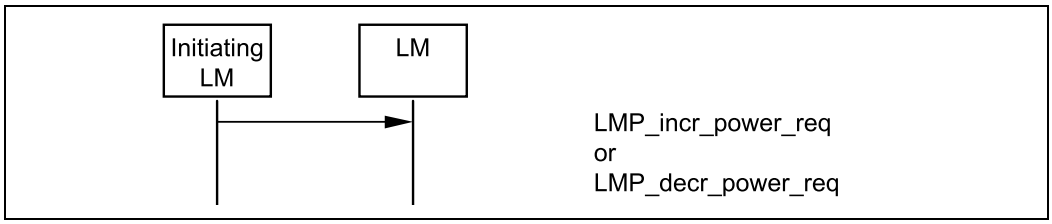
If the received signal characteristics differs too much from the preferred value of a Bluetooth device, it may request an increase or a decrease of the other device's TX power. The power adjustment requests may be made at anytime following a successful baseband paging procedure.

If a device does not support power control requests this is indicated in the supported features list and thus no power control requests shall be sent after the supported features response has been processed. Prior to this time, a power control adjustment might be sent and if the recipient does not support power control it is allowed to send LMP_max_power in response to LMP_incr_power_req and LMP_min_power in response to LMP_decr_power_req. Another possibility is to send LMP_not_accepted with the error code *unsupported LMP feature*.

Upon receipt of an LMP_incr_power_req PDU or LMP_decr_power_req PDU the output power shall be increased or decreased one step. See Radio Specification Section 3, on page 33 for the definition of the step size. The TX power is a property of the physical link, and affects all logical transports carried over the physical link. Power control requests carried over the default ACL-C logical link shall only affect the physical link associated with the default ACL-C logical link: they shall not affect the power level used on the physical links to other slaves.

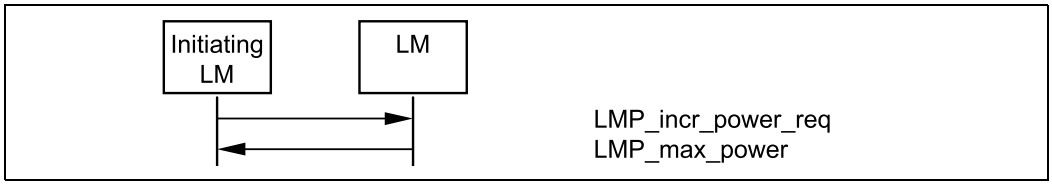
M/O	PDU	Contents
O(9)	LMP_incr_power_req	for future use (1 Byte)
O(9)	LMP_decr_power_req	for future use (1 Byte)
O(18)	LMP_max_power	-
O(18)	LMP_min_power	-

Table 4.3: PDUs used for power control.

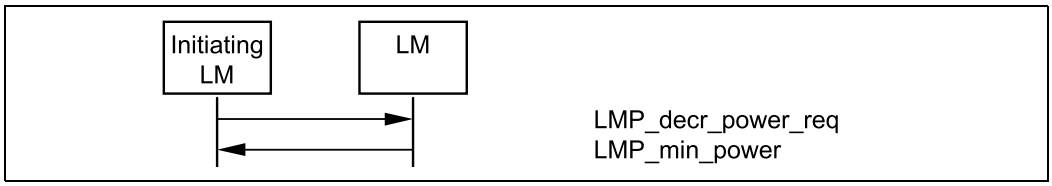


Sequence 2: A device requests a change of the other device's TX power.

If the receiver of LMP_incr_power_req is at maximum power LMP_max_power shall be returned. The device shall only request an increase again after having requested a decrease at least once. If the receiver of LMP_decr_power_req is at minimum power then LMP_min_power shall be returned and the device shall only request a decrease after having requested an increase at least once.



Sequence 3: The TX power cannot be increased.



Sequence 4: The TX power cannot be decreased.

One byte is reserved in LMP_incr/decr_power_req for future use. The parameter value shall be 0x00 and ignored upon receipt.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 370

4.1.4 Adaptive frequency hopping

AFH is used to improve the performance of physical links in the presence of interference as well as reducing the interference caused by physical links on other devices in the ISM band. AFH shall only be used during the connection state.

M/O	PDU	Contents
O(35) Rx O(43) Tx	LMP_set_AFH	AFH_Instant, AFH_Mode, AFH_Channel_Map

Table 4.4: PDUs used for AFH

The LMP_set_AFH PDU contains three parameters: AFH_Instant, AFH_Mode, and AFH_Channel_Map. The parameter, AFH_Instant, specifies the instant at which the hopset switch will become effective. This is specified as a Bluetooth Clock value of the master's clock, that is available to both devices. The AFH instant is chosen by the master and shall be an even value at least $6*T_{poll}$ or 96 slots (whichever is greater) in the future, where T_{poll} is at least the longest poll interval for all AFH enabled physical links. The AFH_instant shall be within 12 hours of the current clock value. The parameter AFH_Mode, specifies whether AFH shall be enabled or disabled. The parameter AFH_Channel_Map, specifies the set of channels that shall be used if AFH is enabled.

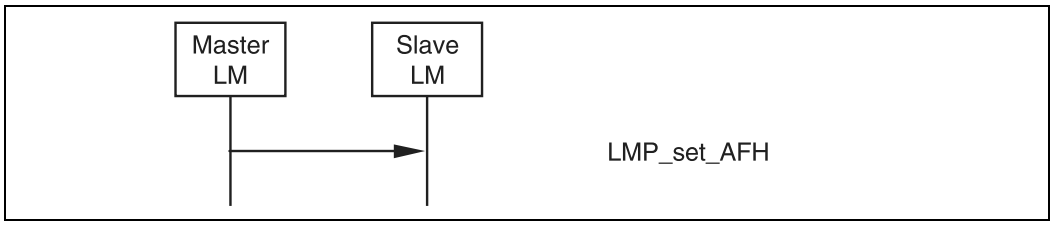
When the LMP_set_AFH PDU is received the AFH instant shall be compared with the current Bluetooth clock value. If it is in the past then the AFH_instant has passed and the slave shall immediately configure the hop selection kernel (see Baseband Specification, Section 2.6.3, on page 77) with the new AFH_mode and AFH_channel_map specified in the LMP_set_AFH PDU. If it is in the future then a timer shall be started to expire at the AFH instant. When this timer expires it shall configure the hop selection kernel with the new AFH_mode and AFH_channel_map.

The master shall not send a new LMP_set_AFH PDU to a slave until it has received the baseband acknowledgement for any previous LMP_set_AFH addressed to that slave and the instant has passed.

Role switch while AFH is enabled shall follow the procedures define by Baseband Specification, Section 8.6.5, on page 155.

4.1.4.1 Master enables AFH

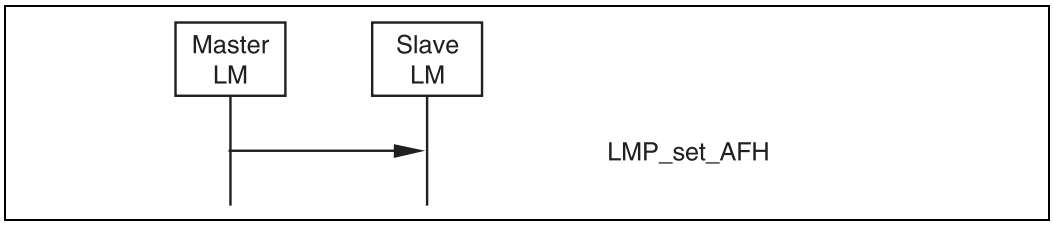
Prior to enabling AFH the master LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 96). The master shall then enable AFH on a physical link by sending the LMP_set_AFH PDU with AFH_mode set to AFH_enabled, the AFH_channel_map parameter containing the set of used and unused channels, and an AFH_instant. The LM shall not calculate the AFH instant until after traffic on the ACL-U logical link has been stopped. The master considers the physical link to be AFH_enabled once the baseband acknowledgement has been received and the AFH_instant has passed. Once the baseband acknowledgement has been received the master shall restart transmission on the ACL-U logical link.



Sequence 5: Master Enables AFH.

4.1.4.2 Master disables AFH

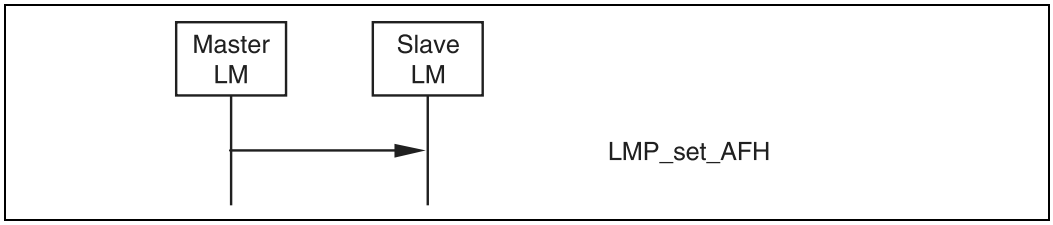
Prior to disabling AFH the master LM shall pause traffic on the ACL-U logical link (Baseband Specification, Section 5.3.1, on page 96). The master shall then disable AFH operation on a physical link by sending the LMP_set_AFH PDU with AFH_mode set to AFH_disabled and an AFH_instant. The AFH_channel_map parameter is not valid when AFH_mode is AFH_disabled. The LM shall not calculate the AFH instant until after traffic on the ACL-U logical link has been stopped. The master considers the physical link to have entered AFH_disabled operation once the baseband acknowledgement has been received and the AFH_instant has passed. Once the baseband acknowledgement has been received the master shall restart transmission on the ACL-U logical link.



Sequence 6: Master disables AFH.

4.1.4.3 Master updates AFH

A master shall update the AFH parameters on a physical link by sending the LMP_set_AFH PDU with AFH_mode set to AFH_enabled, an AFH_instant and a new AFH_channel_map. The master shall consider the slave to have the updated AFH parameters once the baseband acknowledgement has been received and the AFH_instant has passed.



Sequence 7: Master Updates AFH.

4.1.4.4 AFH operation in park, hold and sniff modes

A slave in Park, Hold or Sniff shall retain the AFH_mode and AFH_channel_map prior to entering those modes. A master may change the AFH_mode while a slave is in Sniff.

A master that receives a request from an AFH_enabled slave to enter Park, Hold or Sniff and decides to operate the slave using a different hop sequence shall respond with an LMP_set_AFH PDU specifying the new hop sequence.

The master continues with the LMP signalling, for Park, Hold or Sniff initiation, once the baseband acknowledgement for the LMP_set_AFH PDU has been received. Optionally, the master may delay the continuation of this LMP signalling until after the instant. An AFH_capable_slave device shall support both of these cases.

A master that receives a request from an AFH_enabled slave to enter Park, Hold or Sniff and decides not to change the slave's hop sequence shall respond exactly as it would do without AFH. In this case, AFH operation has no effect on the LMP signalling.

4.1.5 Channel classification

A master may request channel classification information from a slave that is AFH_enabled.

A slave that supports the AFH_classification_slave feature shall perform channel classification and reporting according to its AFH_reporting_mode. The master shall control the AFH_reporting_mode using the LMP_channel_classification_req PDU. The slave shall report its channel classification using the LMP_channel_classification PDU.

The slave shall report pairs of channels as *good*, *bad* or *unknown*. See Table 5.2 on page 277 for the detailed format of the AFH_Channel_Classification parameter. When one channel in the n^{th} channel pair is good and the other channel is unknown the n^{th} channel pair shall be reported as good. When one channel in the n^{th} channel pair is bad and the other is unknown the n^{th} channel pair shall be reported as bad. It is implementation dependent what to report when one channel in a channel pair is good and the other is bad.

M/O	PDU	Contents
O(36) Rx O(44) Tx	LMP_channel_classification_req	AFH_Reporting_Mode, AFH_Min_Interval, AFH_Max_Interval
O(36) Tx O(44) Rx	LMP_channel_classification	AFH_Channel_Classification

Table 4.5: PDUs used for Channel Classification Reporting.

The LMP_channel_classification_req PDU contains three parameters: AFH_Reporting_Mode, AFH_Min_Interval, and AFH_Max_Interval. In the AFH_reporting_disabled state, the slave shall not generate any channel classification reports. The parameter AFH_min_interval, defines the minimum amount of time from the last LMP_channel_classification command that was sent before the next LMP_channel_classification PDU may be sent. The parameter AFH_max_interval, defines the maximum amount of time between the change in the radio environment being detected by a slave and its generation of an LMP_channel_classification PDU. The AFH_max_interval shall be equal to or larger than AFH_min_interval.

The AFH_reporting_mode parameter shall determine if the slave is in the AFH_reporting_enabled or AFH_reporting_disabled state. The default state, prior to receipt of any LMP_channel_classification_req PDUs, shall be AFH_reporting_disabled.

AFH_reporting_mode is implicitly set to the AFH_reporting_disabled state when any of the following occur:

- Establishment of a connection at the baseband level
- Master-slave role switch

Petitioner Kiosoft Exhibit 1011 Part 1
Page 374

- Entry to park state operation
- Entry to hold mode operation

AFH_reporting_mode is implicitly restored to its former value when any of the following occur:

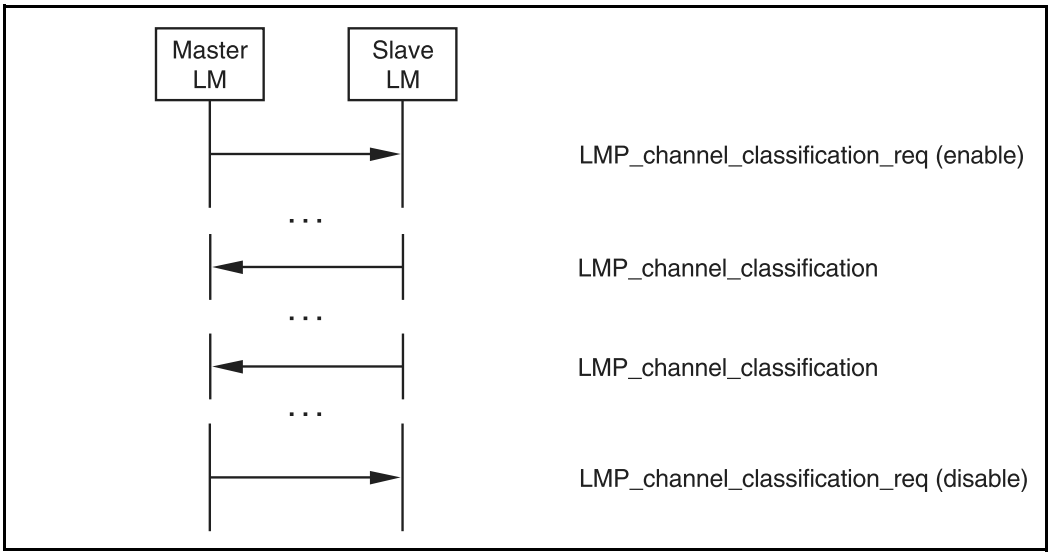
- Exit from park state operation
- Exit from hold mode
- Failure of Master-slave role switch

4.1.5.1 Channel classification reporting enabling and disabling

A master enables slave channel classification reporting by sending the LMP_channel_classification_req PDU with the AFH_reporting_mode parameter set to AFH_reporting_enabled.

When a slave has had classification reporting enabled by the master it shall send the LMP_channel_classification PDU according to the information in the latest LMP_channel_classification_req PDU. The LMP_channel_classification PDU shall not be sent if there has been no change in the slave's channel classification.

A master disables slave channel classification reporting by sending the LMP_channel_classification_req PDU with the AFH_reporting_mode parameter set to AFH_reporting_disabled.



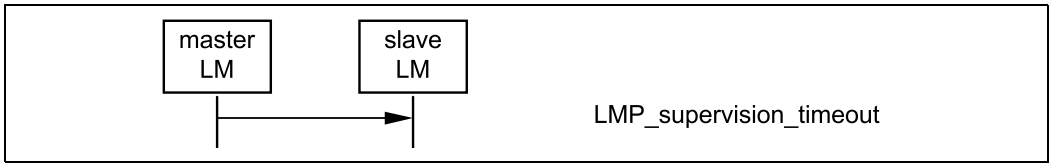
Sequence 8: Channel classification reporting.

4.1.6 Link supervision

Each physical link has a timer that is used for link supervision. This timer is used to detect physical link loss caused by devices moving out of range, or being blocked by interference, a device's power-down, or other similar failure cases. Link supervision is specified in Baseband Specification, Section 3.1, on page 83.

M/O	PDU	Contents
M	LMP_supervision_timeout	supervision timeout

Table 4.6: PDU used to set the supervision timeout.



Sequence 9: Setting the link supervision timeout.

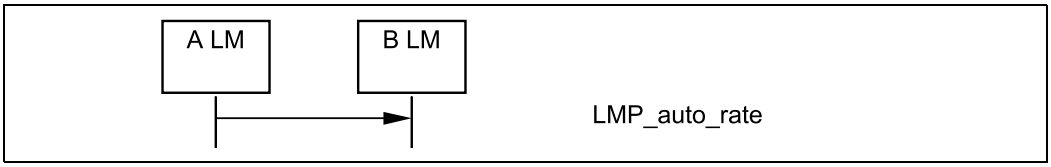
4.1.7 Channel quality driven data rate change (CQDDR)

The data throughput for a given packet type depends on the quality of the RF channel. Quality measurements in the receiver of one device can be used to dynamically control the packet type transmitted from the remote device for optimization of the data throughput. Device A sends the LMP_auto_rate PDU once to notify device B to enable this feature. Once enabled, device B may change the packet type(s) that A transmits by sending the LMP_preferred_rate PDU. This PDU has a parameter which determines the preferred coding (with or without 2/3FEC) and optionally the preferred size in slots of the packets. Device A is not required to change to the packet type specified by this parameter. Device A shall not send a packet that is larger than max slots (see Section 4.1.10 on page 223) even if the preferred size is greater than this value.

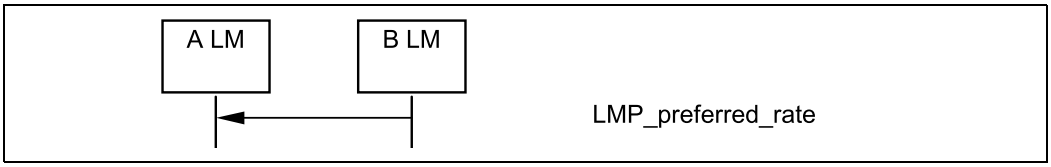
These PDUs may be sent at any time after connection setup is completed.

M/O	PDU	Contents
O(10)	LMP_auto_rate	-
O(10)	LMP_preferred_rate	data rate

Table 4.7: PDUs used for quality driven change of the data rate.



Sequence 10: A notifies B to enable CQDDR



Sequence 11: B sends A a preferred packet type

4.1.8 Quality of service (QoS)

The LM provides QoS capabilities. A poll interval, T_{poll}, that is defined as the maximum time between transmissions from the master to a particular slave on the ACL logical transport, is used to support bandwidth allocation and latency control - see Baseband Specification, Section 8.6.1, on page 150 for details. The poll interval is guaranteed in the active and sniff modes except when there are collisions with page, page scan, inquiry and inquiry scan, during time critical LMP sequences in the current piconet and any other piconets in which the Bluetooth device is a member, and during critical baseband sequences (such as the page response, initial connection state until the first POLL, and master slave switch). These PDUs maybe sent at anytime after connection setup is completed.

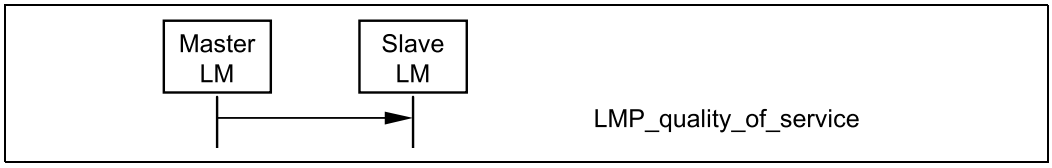
Master and slave negotiate the number of repetitions for broadcast packets (N_{BC}), see Baseband Specification, Section 7.6.5, on page 130.

M/O	PDU	Contents
M	LMP_quality_of_service	poll interval N_{BC}
M	LMP_quality_of_service_req	poll interval N_{BC}

Table 4.8: PDUs used for quality of service.

4.1.8.1 Master notifies slave of the quality of service

The master notifies the slave of the new poll interval and N_{BC} by sending the LMP_quality_of_service PDU. The slave cannot reject the notification.

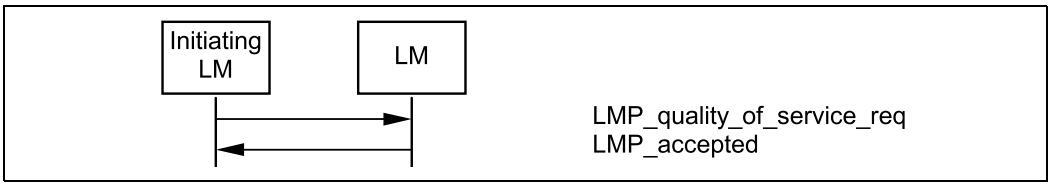


Sequence 12: Master notifies slave of quality of service.

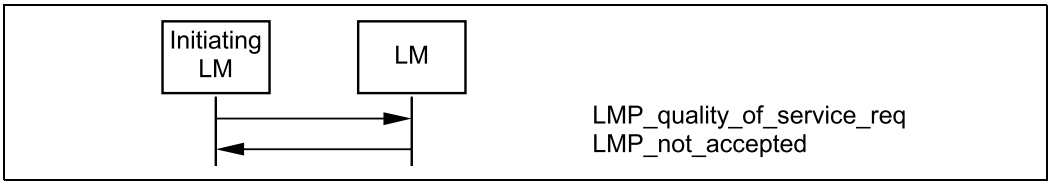
4.1.8.2 Device requests new quality of service

Either the master or the slave may request a new poll interval and N_{BC} by sending an LMP_quality_of_service_req PDU to the slave. The parameter N_{BC} is meaningful only when it is sent by a master to a slave. For transmission of LMP_quality_of_service_req PDUs from a slave, this parameter shall be ignored by the master. The request can be accepted or rejected. This allows the master and slave to dynamically negotiate the quality of service as needed.

The selected poll interval by the slave shall be less than or equal to the specified Access Latency for the outgoing traffic of the ACL link (see L2CAP “Quality of Service (QoS) Option” on page 60[vol. 3]).



Sequence 13: Device accepts new quality of service



Sequence 14: Device rejects new quality of service.

4.1.9 Paging scheme parameters

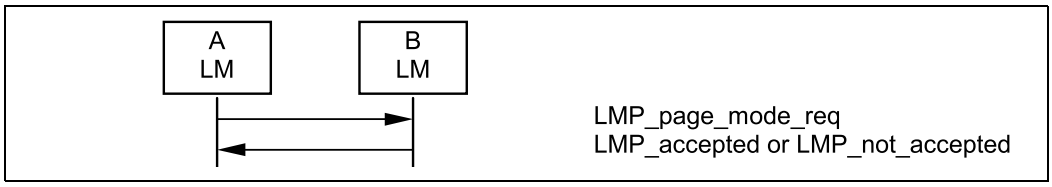
LMP provides a means to negotiate the paging scheme parameters that are used the next time a device is paged.

M/O	PDU	Contents
O(17)	LMP_page_mode_req	paging scheme paging scheme settings
O(17)	LMP_page_scan_mode_req	paging scheme paging scheme settings

Table 4.9: PDUs used to request paging scheme.

4.1.9.1 Page mode

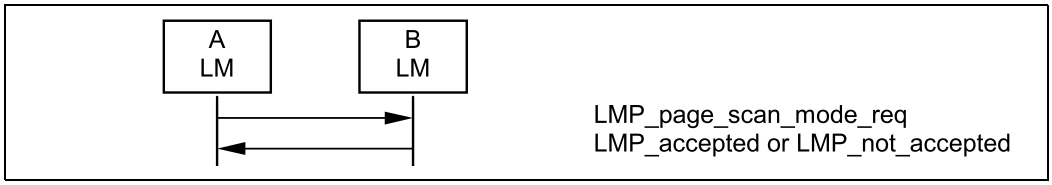
This procedure is initiated from device A and negotiates the paging scheme used when device A pages device B. Device A proposes a paging scheme including the parameters for this scheme and device B can accept or reject. On rejection the old setting will not be changed. A request to switch to a reserved paging scheme shall be rejected.



Sequence 15: Negotiation for page mode.

4.1.9.2 Page scan mode

This procedure is initiated from device A and negotiates the paging scheme and paging scheme settings used when device B pages device A. Device A proposes a paging scheme and paging scheme settings and device B may accept or reject. On reject the old setting is not changed. A request specifying the mandatory scheme shall be accepted. A request specifying a non-mandatory scheme shall be rejected. This procedure should be used when device A changes its paging scheme settings. A slave should also send this message to the master after connection establishment, to inform the master of the slave's current paging scheme and paging scheme settings.



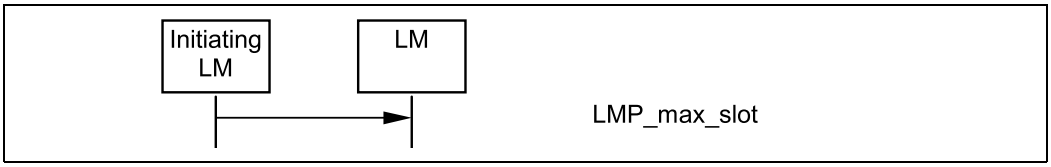
Sequence 16: Negotiation for page scan mode

4.1.10 Control of multi-slot packets

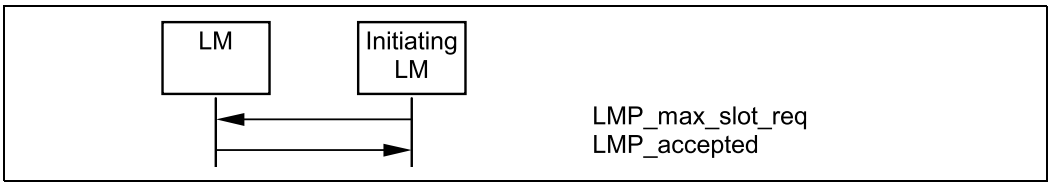
The number of consecutive slots used by a device on an ACL-U logical link can be limited. It does not affect traffic on the eSCO links where the packet sizes are defined as part of link setup. A device allows the remote device to use a maximum number of slots by sending the PDU LMP_max_slot providing max slots as parameter. Each device can request to use a maximal number of slots by sending the PDU LMP_max_slot_req providing max slots as parameter. After a new connection, as a result of page, page scan, role switch or unpark, the default value is 1 slot. These PDUs can be sent at anytime after connection setup is completed.

M/O	PDU	Contents
M	LMP_max_slot	max slots
M	LMP_max_slot_req	max slots

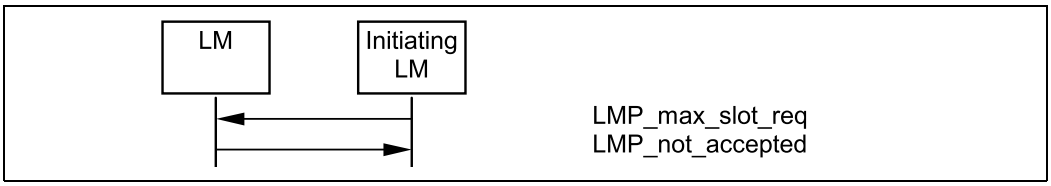
Table 4.10: PDUs used to control the use of multi-slot packets.



Sequence 17: Device allows Remote Device to use a maximum number of slots.



Sequence 18: Device requests a maximum number of slots. Remote Device accepts.



Sequence 19: Device requests a maximum number of slots. Remote Device rejects.

4.2 SECURITY

4.2.1 Authentication

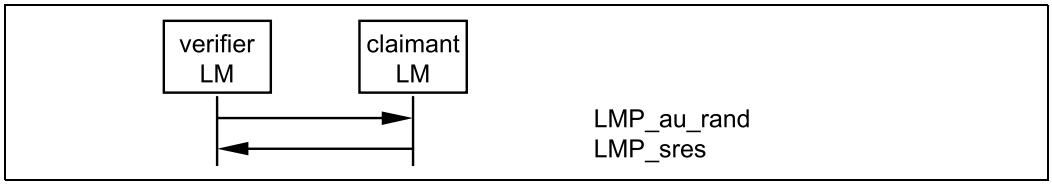
The authentication procedure is based on a challenge-response scheme as described in [Part H] Section 5 on page 773. The verifier sends an LMP_au_rand PDU that contains a random number (the challenge) to the claimant. The claimant calculates a response, that is a function of this challenge, the claimant's BD_ADDR and a secret key. The response is sent back to the verifier, that checks if the response was correct or not. The response shall be calculated as described in [Part H] Section 6.1 on page 777. A successful calculation of the authentication response requires that two devices share a secret key. This key is created as described in Section 4.2.2 on page 226. Both the master and the slave can be verifiers.

M/O	PDU	Contents
M	LMP_au_rand	random number
M	LMP_sres	authentication response

Table 4.11: PDUs used for authentication.

4.2.1.1 Claimant has link key

If the claimant has a link key associated with the verifier, it shall calculate the response and sends it to the verifier with LMP_sres. The verifier checks the response. If the response is not correct, the verifier can end the connection by sending an LMP_detach PDU with the error code *authentication failure,* see Section 4.1.2 on page 210.



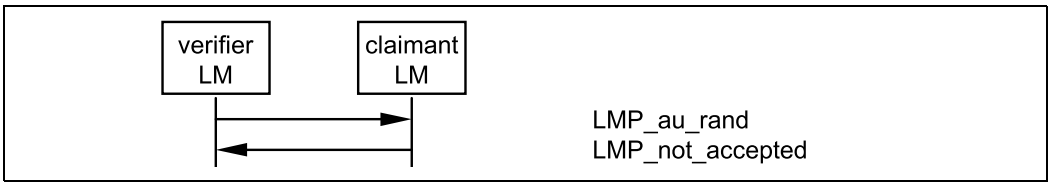
Sequence 20: Authentication. Claimant has link key.

Upon reception of an LMP_au_rand, an LM shall reply with LMP_sres before initiating its own authentication.
Note: there can be concurrent requests caused by the master and slave simultaneously initiating an authentication. The procedures in Section 2.5.1 on page 201 assures that devices will not have different Authenticated Cyphering Offset (ACO, see [Part H] Section 6.1 on page 777) when they calculate the encryption key.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 382

4.2.1.2 Claimant has no link key

If the claimant does not have a link key associated with the verifier it shall send an LMP_not_accepted PDU with the error code *key missing* after receiving an LMP_au_rand PDU.



Sequence 21: Authentication fails. Claimant has no link key.

4.2.1.3 Repeated attempts

The scheme described in [Part H] Section 5.1 on page 775 shall be applied when an authentication fails. This will prevent an intruder from trying a large number of keys in a relatively short time.

4.2.2 Pairing

When two devices do not have a common link key an initialization key (K_{init}) shall be created based on a PIN, and a random number and a BD_ADDR. K_{init} shall be created as specified in [Part H] Section 6.3 on page 781. When both devices have calculated K_{init} the link key shall be created, and a mutual authentication is performed. The pairing procedure starts with a device sending an LMP_in_rand PDU; this device is referred to as the "initiating LM" or "initiator" in Section 4.2.2.1 on page 226 - Section 4.2.2.5 on page 228. The other device is referred to as the "responding LM" or "responder". The PDUs used in the pairing procedure are:

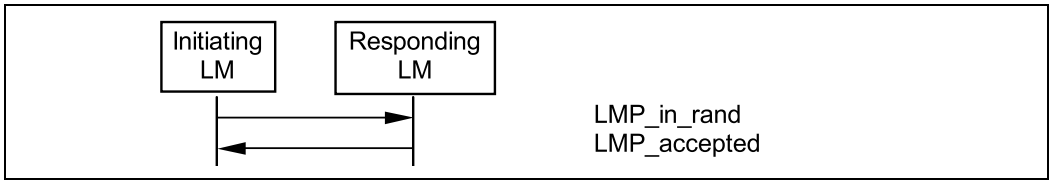
M/O	PDU	Contents
M	LMP_in_rand	random number
M	LMP_au_rand	random number
M	LMP_sres	authentication response
M	LMP_comb_key	random number
M	LMP_unit_key	key

Table 4.12: PDUs used for pairing

All sequences described in Section 4 on page 209, including the mutual authentication after the link key has been created, shall form a single transaction. The transaction ID from the first LMP_in_rand shall be used for all subsequent sequences.

4.2.2.1 Responder accepts pairing

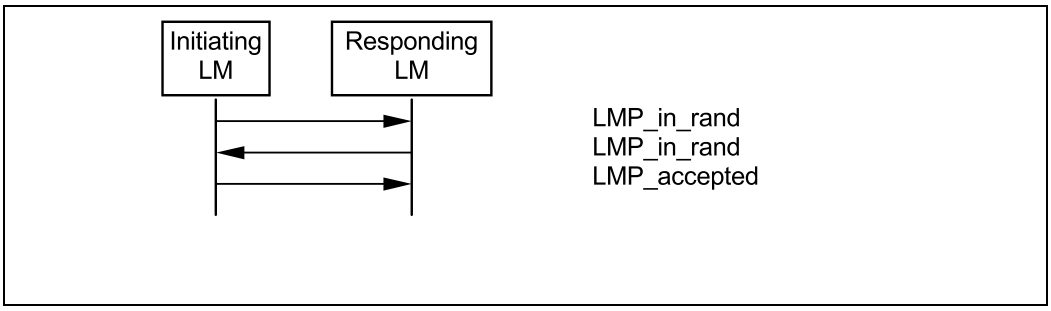
When the initiator sends an LMP_in_rand PDU and the responder shall reply with an LMP_accepted PDU. Both devices shall then calculate K_{init} based on the BD_ADDR of the responder and the procedure continues with creation of the link key; see Section 4.2.2.4 on page 228.



Sequence 22: Pairing accepted. Responder has a variable PIN. Initiator has a variable or fixed PIN.

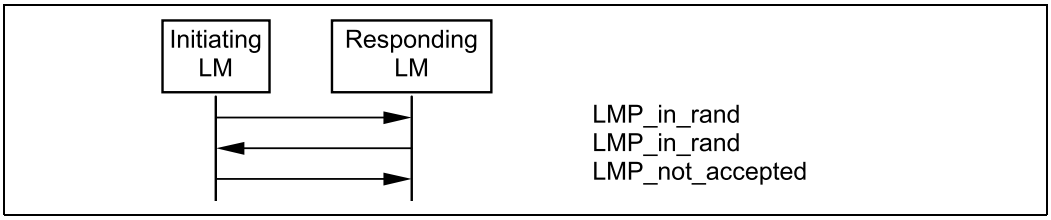
4.2.2.2 Responder has a fixed PIN

If the responder has a fixed PIN it shall generate a new random number and send it back in an LMP_in_rand PDU. If the initiator has a variable PIN it shall accept the LMP_in_rand PDU and shall respond with an LMP_accepted PDU. Both sides shall then calculate K_{init} based on the last IN_RAND and the BD_ADDR of the initiator. The procedure continues with creation of the link key; see Section 4.2.2.4 on page 228.



Sequence 23: Responder has a fixed PIN and initiator has a variable PIN.

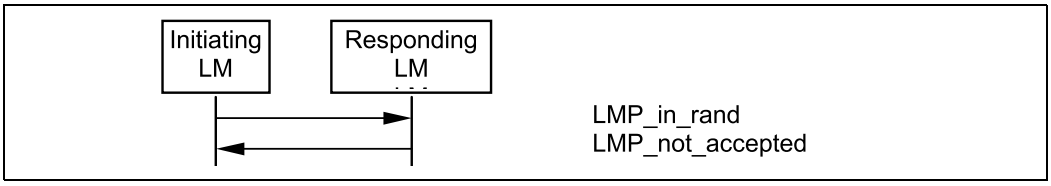
If the responder has a fixed PIN and the initiator also has a fixed PIN, the second LMP_in_rand shall be rejected by the initiator sending an LMP_not_accepted PDU with the error code *pairing not allowed*.



Sequence 24: Both devices have a fixed PIN.

4.2.2.3 Responder rejects pairing

If the responder rejects pairing it shall send an LMP_not_accepted PDU with the error code *pairing not allowed* after receiving an LMP_in_rand PDU.



Sequence 25: Responder rejects pairing.

4.2.2.4 Creation of the link key

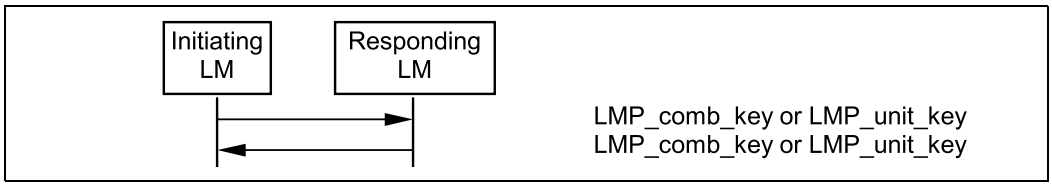
When K_{init} is calculated in both devices the link key shall be created. This link key will be used in the authentication between the two devices for all subsequent connections until it is changed; see Section 4.2.3 on page 229 and Section 4.2.4 on page 230. The link key created in the pairing procedure will either be a combination key or one of the device's unit keys. The following rules shall apply to the selection of the link key:

- if one device sends an LMP_unit_key PDU and the other device sends LMP_comb_key, the unit key will be the link key.
- if both devices send an LMP_unit_key PDU, the master's unit key will be the link key.
- if both devices send an LMP_comb_key PDU, the link key shall be calculated as described in [Part H] Section 3.2 on page 755.

The content of the LMP_unit_key PDU is the unit key bitwise XORed with K_{init}. The content of the LMP_comb_key PDU is LK_RAND bitwise XORed with K_{init}. Any device configured to use a combination key shall store the link key.

The use of unit keys is deprecated since it is implicitly insecure.

When the link key, combination or unit key, has been created mutual authentication shall be performed to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication is performed with the initiator as the verifier. When finalized an authentication in the reverse direction is performed.



Sequence 26: Creation of the link key.

4.2.2.5 Repeated attempts

When the authentication after creation of the link key fails because of an incorrect authentication response, the same scheme as in Section 4.2.1.3 on page 225 shall be used. This prevents an intruder from trying a large number of different PINs in a relatively short time.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 386

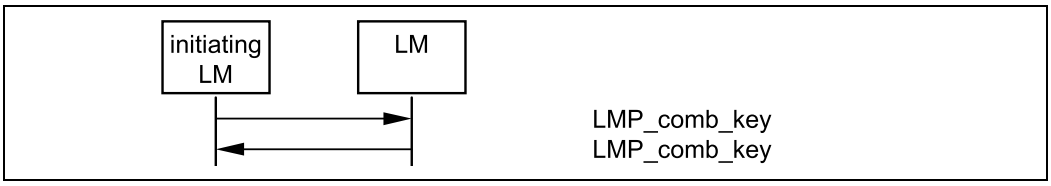
4.2.3 Change link key

If the link key is derived from combination keys and the current link is the semi-permanent link key, the link key can be changed. If the link key is a unit key, the devices shall go through the pairing procedure in order to change the link key. The contents of the LMP_comb_key PDU is protected by a bitwise XOR with the current link key.

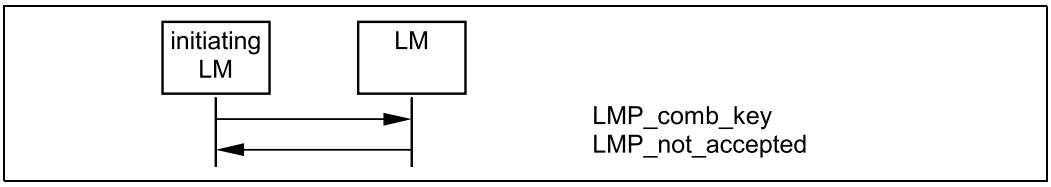
M/O	PDU	Contents
M	LMP_comb_key	random number

Table 4.13: PDUs used for change of link key.

All sequences described in Section 4.2.3 , including the mutual authentication after the link key has been changed, shall form a single transaction. The transaction ID from the first LMP_comb_key PDU shall be used for all subsequent sequences.



Sequence 27: Successful change of the link key.



Sequence 28: Change of the link key not possible since the other device uses a unit key.

If the change of link key is successful the new link key shall be stored and the old link key shall be discarded. The new link key shall be used as link key for all the following connections between the two devices until the link key is changed again. The new link key also becomes the current link key. It will remain the current link key until the link key is changed again, or until a temporary link key is created, see Section 4.2.4 on page 230.

When the new link key has been created mutual authentication shall be performed to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication is performed with the device that initiated change link key as verifier. When finalized an authentication in the reverse direction is performed.

4.2.4 Change current link key type

The current link key can be a semi-permanent link key or a temporary link key. It may be changed temporarily, but the change shall only be valid for the current connection, see [Part H] Section 3.1 on page 753. Changing to a temporary link key is necessary if the piconet is to support encrypted broadcast. The current link key may not be changed before the connection establishment procedure has completed. This feature is only supported if broadcast encryption is supported as indicated by the LMP features mask.

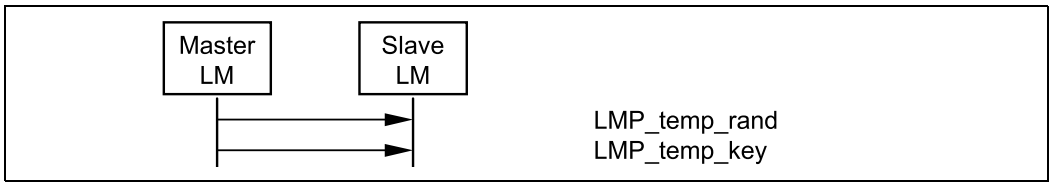
M/O	PDU	Contents
O(23)	LMP_temp_rand	random number
O(23)	LMP_temp_key	key
O(23)	LMP_use_semi_permanent_key	-

Table 4.14: PDUs used to change the current link key.

4.2.4.1 Change to a temporary link key

The master starts by creating the master key K_{master} as specified in Security Specification (EQ 4), on page 760. Then the master shall generate a random number, RAND, and shall send it to the slave in an LMP_temp_rand PDU. Both sides then calculate an overlay denoted OVL as OVL= E_{22}(current link key, RAND, 16). The master shall then send K_{master} protected by a modulo-2 addition with OVL to the slave in an LMP_temp_key PDU. The slave calculates K_{master}, based on OVL, that becomes the current link key. It shall be the current link key until the devices fall back to the semi-permanent link key, see section 4.2.4.2 on page 231.

Note: the terminology in this section is the same as used in [Part H] Section 3.2.8 on page 760.



Sequence 29: Change to a temporary link key.

All sequences described in Section 4.2.4.1 on page 230, including the mutual authentication after K_{master} has been created, shall form a single transaction. The transaction ID shall be set to 0.

When the devices have changed to the temporary key, a mutual authentication shall be made to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication shall be per-

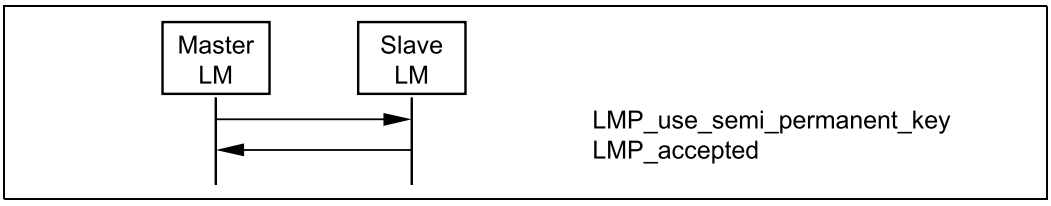
Petitioner Kiosoft Exhibit 1011 Part 1
Page 388

formed with the master as verifier. When finalized an authentication in the reverse direction is performed.

Should the mutual authentication fail at either side, the LM of the verifier should start the detach procedure. This will allow the procedure to succeed even though one of the devices may be erroneous.

4.2.4.2 Make the semi-permanent link key the current link key

After the current link key has been changed to K_{master}, this change can be undone and the semi-permanent link key becomes the current link key again. If encryption is used on the link, the procedure to go back to the semi-permanent link key shall be immediately followed by the master stopping encryption using the procedure described in Section 4.2.5.4 on page 235. Encryption may be restarted by the master according to the procedures in section 4.2.5.1 on page 232 subsection 3. This is to assure that encryption with encryption parameters known by other devices in the piconet is not used when the semi-permanent link key is the current link key.



Sequence 30: Link key changed to the semi-permanent link key.

4.2.5 Encryption

If at least one authentication has been performed encryption may be used. In order for the master to use the same encryption parameters for all slaves in the piconet it shall issue a temporary key, K_{master}. The master shall make this key the current link key for all slaves in the piconet before encryption is started, see Section 4.2.4 on page 230. This is required if broadcast packets are to be encrypted.

M/O	PDU	Contents
O	LMP_encryption_mode_req	encryption mode
O	LMP_encryption_key_size_req	key size
O	LMP_start_encryption_req	random number
O	LMP_stop_encryption_req	-

Table 4.15: PDUs used for handling encryption.

All sequences described in Section 4.2.5 shall form a single transaction. The transaction ID from the LMP_encryption_mode_req PDU shall be used for all subsequent sequences.

4.2.5.1 Encryption mode

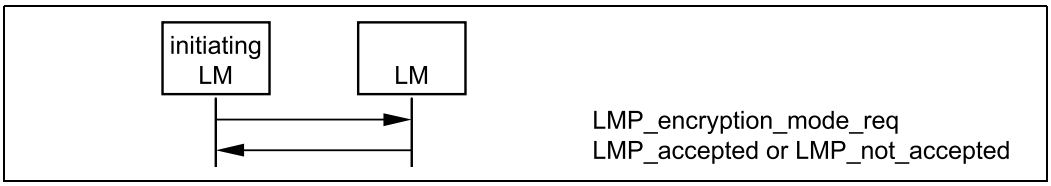
The master and the slave must agree upon whether to use encryption (encryption mode=1 in lmp_encryption_mode_req) or not (encryption mode=0). If the semi-permanent key is used (Key_Flag=0x00) encryption shall only apply to point-to-point packets. If the master link key is used (Key_Flag=0x01) encryption shall apply to both point-to-point packets and broadcast packets. If master and slave agree on the encryption mode, the master continues to give more detailed information about the encryption.

Devices should never send LMP_encryption_mode_req with an encryption mode value of 2 however for backwards compatibility if the LMP_encryption_mode_req is received with an encryption mode value of 2 then it should be treated the same as an encryption mode value of 1.

The initiating LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 96). The initiating device shall then send the LMP_encryption_mode_req PDU. If the responding device accepts the change in encryption mode then it shall complete the transmission of the current packet on the ACL logical transport and shall then suspend transmission on the ACL-U logical link. The responding device shall then send the LMP_accepted PDU.

ACL-U logical link traffic shall only be resumed after the attempt to encrypt or decrypt the logical transport is completed i.e. at the end of Sequence 31, 32 or 33.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 390

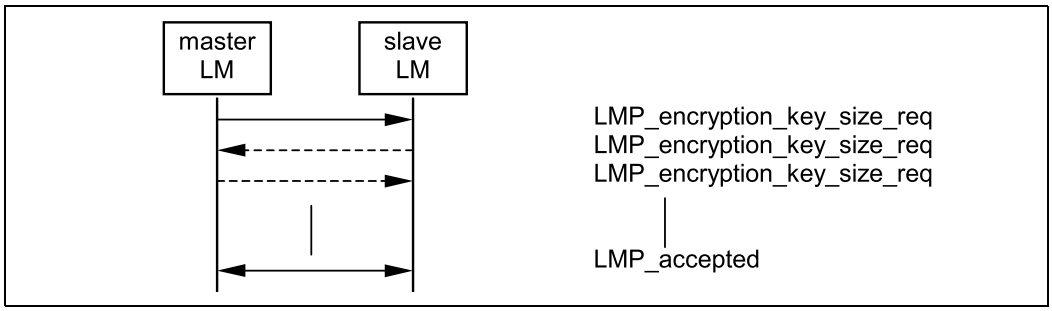


Sequence 31: Negotiation for encryption mode.

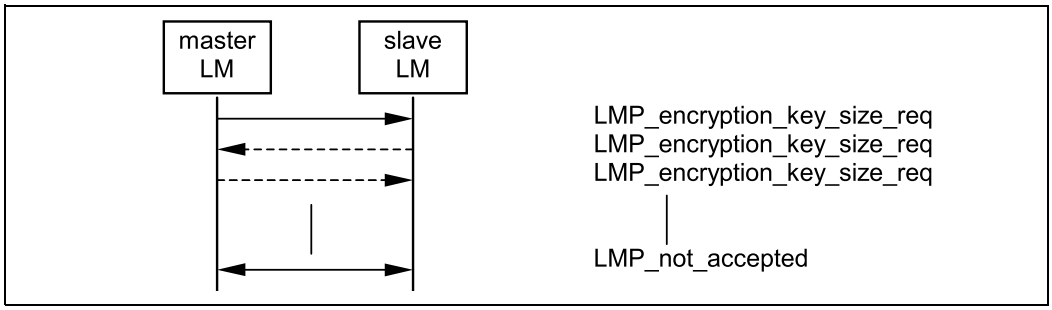
After a device has sent an LMP_encryption_mode_req PDU it shall not send an LMP_au_rand PDU before encryption is started. After a device has received an LMP_encryption_mode_req PDU and sent an LMP_accepted PDU it shall not send an LMP_au_rand PDU before encryption is started. If an LMP_au_rand PDU is sent violating these rules, the claimant shall respond with an LMP_not_accepted PDU with the error code *PDU not allowed*. This assures that devices will not have different ACOs when they calculate the encryption key. If the encryption mode is not accepted or the encryption key size negotiation results in disagreement the devices may send an LMP_au_rand PDU again.

4.2.5.2 Encryption key size

Note: this section uses the same terms as in [Part H] Section 4.1 on page 764. The master sends an LMP_encryption_key_size_req PDU including the suggested key size L_{sug}, m, that is initially equal to $L_{max, m}$. If $L_{min, s} \leq L_{sug, m}$ and the slave supports $L_{sug, m}$ it shall respond with an LMP_accepted PDU and $L_{sug, m}$ shall be used as the key size. If both conditions are not fulfilled the slave sends back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug, s}$. This value shall be the slave's largest supported key size that is less than $L_{sug, m}$. Then the master performs the corresponding test on the slave's suggestion. This procedure is repeated until a key size agreement is reached or it becomes clear that no such agreement can be reached. If an agreement is reached a device sends an LMP_accepted PDU and the key size in the last LMP_encryption_key_size_req PDU shall be used. After this, encryption is started; see Section 4.2.5.3 on page 234. If an agreement is not reached a device sends an LMP_not_accepted PDU with the error code *unsupported parameter value* and the devices shall not communicate using encryption.



Sequence 32: Encryption key size negotiation successful.



Sequence 33: Encryption key size negotiation failed.

4.2.5.3 Start encryption

To start encryption, the master issues the random number EN_RAND and calculates the encryption key. See [Part H] Section 3.2.5 on page 758. The random number shall be the same for all slaves in the piconet when broadcast encryption is used. The master then sends an LMP_start_encryption_req PDU, that includes EN_RAND. The slave shall calculate the encryption key when this message is received and shall acknowledge with an LMP_accepted PDU.



Sequence 34: Start of encryption.

Starting encryption shall be performed in three steps:

1. Master is configured to transmit unencrypted packets and to receive encrypted packets.
2. Slave is configured to transmit and receive encrypted packets.
3. Master is configured to transmit and receive encrypted packets.

Between step 1 and step 2, master-to-slave transmission is possible. This is when an LMP_start_encryption_req PDU is transmitted. Step 2 is triggered when the slave receives this message. Between step 2 and step 3, slave-to-master
transmission is possible. This is when an LMP_accepted PDU is transmitted. Step 3 is triggered when the master receives this message.

4.2.5.4 Stop encryption

To stop encryption a device shall send an LMP_encryption_mode_req PDU with the parameter encryption mode equal to 0 (no encryption). The other device responds with an LMP_accepted PDU or an LMP_not_accepted PDU (the procedure is described in Sequence 31 in section 4.2.5.1 on page 232). If accepted, encryption shall be stopped by the master sending an LMP_stop_encryption_req PDU and the slave shall respond with an LMP_accepted PDU according to Sequence 35.



Sequence 35: Stop of encryption.

Stopping encryption shall be performed in three steps, similar to the procedure for starting encryption.

1. Master is configured to transmit encrypted packets and to receive unencrypted packets.
2. Slave is configured to transmit and receive unencrypted packets.
3. Master is configured to transmit and receive unencrypted packets.

Between step 1 and step 2 master to slave transmission is possible. This is when an LMP_stop_encryption_req PDU is transmitted. Step 2 is triggered when the slave receives this message. Between step 2 and step 3 slave to master transmission is possible. This is when an LMP_accepted PDU is transmitted. Step 3 is triggered when the master receives this message.

4.2.5.5 Change encryption mode, key or random number

If the encryption key or encryption random number need to be changed or if the current link key needs to be changed according to the procedures in Section 4.2.4 on page 230, encryption shall be stopped and re-started after completion, using the procedures in Section 4.2.5 on page 232, subsections 3-4 for the new parameters to take effect.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 393

4.2.6 Request supported encryption key size

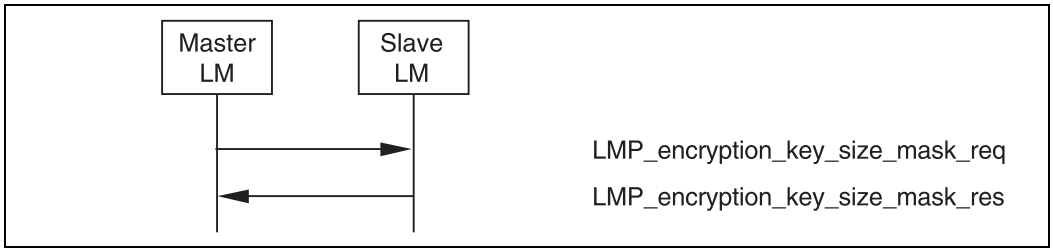
When broadcast encryption is supported via the LMP features mask, it is possible for the master to request a slave's supported encryption key sizes.

M/O	PDU	Contents
O(23)	LMP_encryption_key_size_mask_req	
O(23)	LMP_encryption_key_size_mask_res	key size mask

Table 4.16: PDUs used for encryption key size request

The master shall send an LMP_key_size_req PDU to the slave to obtain the slaves supported encryption key sizes.

The slave shall return a bit mask indicating all broadcast encryption key sizes supported. The least significant bit shall indicate support for a key size of 1, the next most significant bit shall indicate support for a key size of 2 and so on up to a key size of 16. In all cases a bit set to 1 shall indicate support for a key size; a bit set to 0 shall indicate that the key size is not supported.



Sequence 36: Request for supported encryption key sizes.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 394

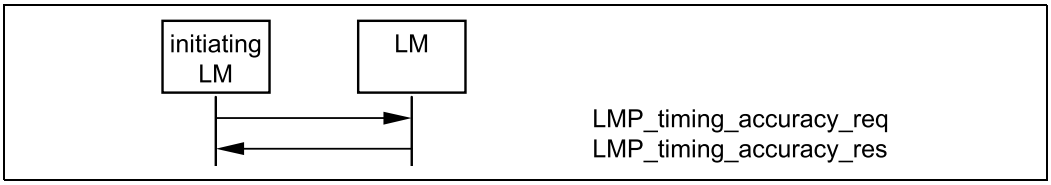
4.3 INFORMATIONAL REQUESTS

4.3.1 Timing accuracy

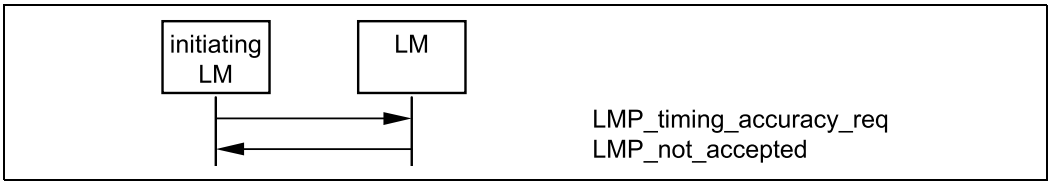
LMP supports requests for the timing accuracy. This information can be used to minimize the scan window during piconet physical channel re-synchronization (see Baseband Specification, Section 2.2.5.2, on page 62). The timing accuracy parameters returned are the long term drift measured in ppm and the long term jitter measured in μs of the worst case clock used. These parameters are fixed for a certain device and shall be identical when requested several times. Otherwise, the requesting device shall assume worst case values (drift=250ppm and jitter=10μs).

M/O	PDU	Contents
O(4)	LMP_timing_accuracy_req	-
O(4)	LMP_timing_accuracy_res	drift jitter

Table 4.17: PDUs used for requesting timing accuracy information.



Sequence 37: The requested device supports timing accuracy information.



Sequence 38: The requested device does not support timing accuracy information.

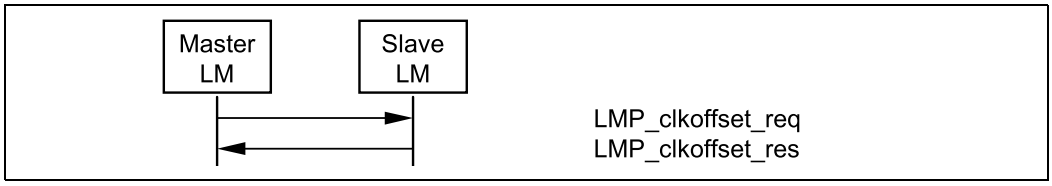
4.3.2 Clock offset

The clock offset can be used to speed up the paging time the next time the same device is paged. The master can request the clock offset at anytime following a successful baseband paging procedure (i.e., before, during or after connection setup). The clock offset shall be defined by the following equation:

$(CLKN_{16\text{-}2\ slave} - CLKN_{16\text{-}2\ master})$ mod 2**15.

M/O	PDU	Contents
M	LMP_clkoffset_req	-
M	LMP_clkoffset_res	clock offset

Table 4.18: PDUs used for clock offset request.



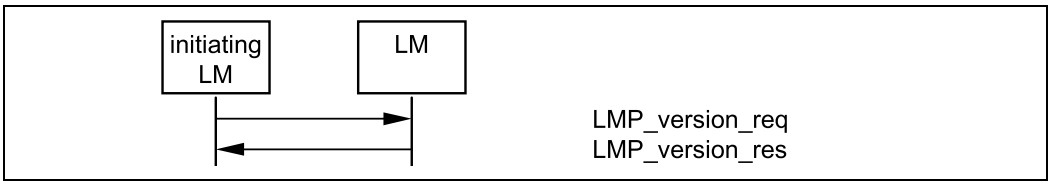
Sequence 39: Clock offset requested.

4.3.3 LMP version

LMP supports requests for the version of the LM protocol. The LMP_version_req and LMP_version_res PDUs contain three parameters: VersNr, CompId and SubVersNr. VersNr specifies the version of the Bluetooth LMP specification that the device supports. CompId is used to track possible problems with the lower Bluetooth layers. All companies that create a unique implementation of the LM shall have their own CompId. The same company is also responsible for the administration and maintenance of the SubVersNr. It is recommended that each company has a unique SubVersNr for each RF/BB/LM implementation. For a given VersNr and CompId, the values of the SubVersNr shall increase each time a new implementation is released. For both CompId and SubVersNr the value 0xFFFF means that no valid number applies. There is no ability to negotiate the version of the LMP. The sequence below is only used to exchange the parameters. LMP version can be requested at anytime following a successful baseband paging procedure.

M/O	PDU	Contents
M	LMP_version_req	VersNr CompId SubVersNr
M	LMP_version_res	VersNr CompId SubVersNr

Table 4.19: PDUs used for LMP version request.



Sequence 40: Request for LMP version.

4.3.4 Supported features

The supported features may be requested at anytime following a successful baseband paging procedure by sending the LMP_features_req PDU. Upon reception of an LMP_features_req PDU, the receiving device shall return an LMP_features_res PDU.

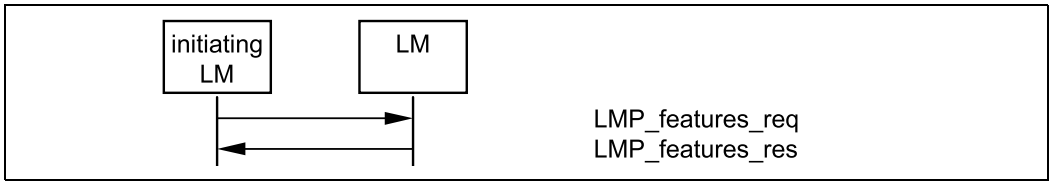
The number of features bits required will in the future exceed the size of a single page of features. An extended features mask is therefore provided to allow support for more than 64 features. Support for the extended features mask is indicated by the presence of the appropriate bit in the LMP features mask. The LMP_features_req_ext and LMP_features_res_ext PDUs operate in precisely the same way as the LMP_features_req and LMP_features_res PDUs except that they allow the various pages of the extended features mask to be requested. The LMP_features_req_ext may be sent at any time following the exchange of the LMP_features_req and LMP_features_rsp PDUs.

The LMP_features_req_ext PDU contains a feature page index that specifies which page is requested and the contents of that page for the requesting device. Pages are numbered from 0-255 with page 0 corresponding to the normal features mask. Each page consists of 64 bits. If a device does not support any page number it shall return a mask with every bit set to 0. It also contains the maximum features page number containing any non-zero bit for this device. The recipient of an LMP_features_req_ext PDU shall respond with an LMP_features_res_ext PDU containing the same page number and the appropriate features page along with its own maximum features page number.

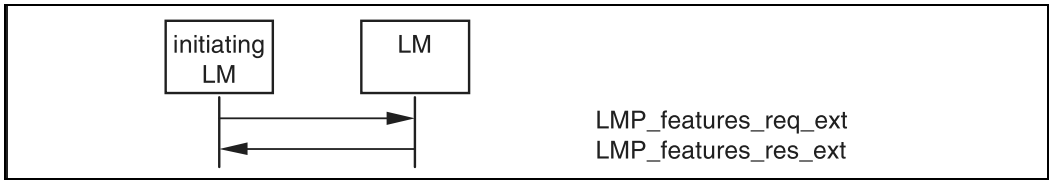
If the extended features request is not supported then all bits in all extended features pages for that device shall be assumed to be zero.

M/O	PDU	Contents
M	LMP_features_req	features
M	LMP_features_res	features
O(63)	LMP_features_req_ext	features page max supported page extended features
O(63)	LMP_features_res_ext	features page max supported page extended features

Table 4.20: PDUs used for features request.



Sequence 41: Request for supported features.



Sequence 42: Request for extended features.

4.3.5 Name request

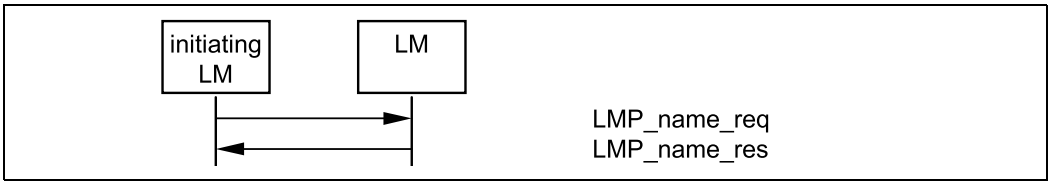
LMP supports name request to another device. The name is a user-friendly name associated with the device and consists of a maximum of 248 bytes coded according to the UTF-8 standard. The name is fragmented over one or more DM1 packets. When an LMP_name_req PDU is sent, a name offset indicates which fragment is expected. The corresponding LMP_name_res PDU carries the same name offset, the name length indicating the total number of bytes in the name of the device and the name fragment, where:

- name fragment(N) = name(N + name offset), if (N + name offset) < name length
- name fragment(N) = 0,otherwise.

Here $0 \leq N \leq 13$. In the first sent LMP_name_req PDU, name offset=0. Sequence 43 is then repeated until the initiator has collected all fragments of the name. The name request may be made at any time following a successful baseband paging procedure.

M/O	PDU	Contents
M	LMP_name_req	name offset
M	LMP_name_res	name offset name length name fragment

Table 4.21: PDUs used for name request.



Sequence 43: Device's name requested and it responses.

4.4 ROLE SWITCH

4.4.1 Slot offset

With LMP_slot_offset the information about the difference between the slot boundaries in different piconets is transmitted. The LMP_slot_offset PDU may be sent anytime after the baseband paging procedure has completed. This PDU carries the parameters slot offset and BD_ADDR. The slot offset shall be the time in microseconds between the start of a master transmission in the current piconet to the start of the next following master transmission in the piconet where the BD_ADDR device (normally the slave) is master at the time that the request is interpreted by the BD_ADDR device.

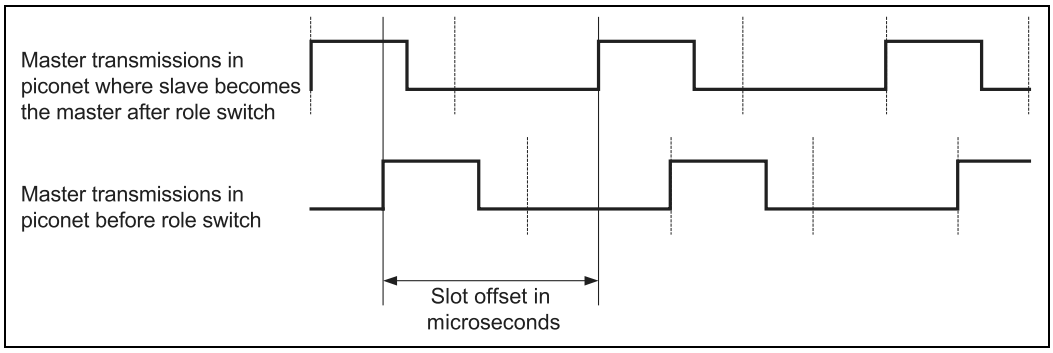
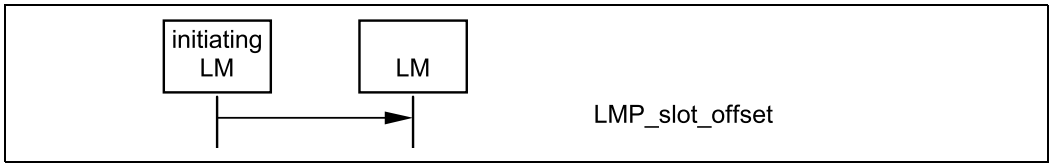


Figure 4.2: Slot offset for role switch.

See Section 4.4 on page 242 for the use of LMP_slot_offset in the context of the role switch. In the case of role switch the BD_ADDR is that of the slave device.

M/O	PDU	Contents
O(3)	LMP_slot_offset	slot offset BD_ADDR

Table 4.22: PDU used for slot offset information.



Sequence 44: Slot offset information is sent.

Petitioner Kiosoft Exhibit 1011 Part 1
Page 400