



# SPECIFICATION OF THE *BLUETOOTH SYSTEM*

Experience More

## Bluetooth Core Specification Addendum 4

Adoption Date: 12 February 2013



## Disclaimer and Copyright Notice

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

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

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

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

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

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

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

Copyright © 1999 - 2013

Ericsson AB,  
Lenovo, (Singapore) Pte. Ltd  
Intel Corporation,  
Microsoft Corporation,  
Motorola, Mobility Inc.,  
Nokia Corporation,  
Toshiba Corporation

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



## ABOUT ADDENDUM 4

---

This addendum provides an optional update to the Bluetooth® Core Specification. When the addendum is applied to an allowed Core Specification, the following parts of the specification shall be replaced, added, or appended with the revised versions:

- Volume 1**
- Part A Architecture
  - Part B Acronyms & Abbreviations
  - Part D Mixing of Specification Versions

- Volume 2**
- Part B Baseband Specification
  - Part C Link Manager Protocol Specification
  - Part E Host Controller Interface Functional Specification
  - Part F Message Sequence Charts
  - Part G Sample Data

- Volume 3**
- Part C Generic Access Profile



## TABLE OF CONTENTS

### MIXING OF SPECIFICATION VERSIONS

<b>Contents .....</b>	<b>6</b>
<b>1 Mixing of Specification Versions .....</b>	<b>7</b>
1.1 Features and their Types .....	8
1.2 Core Specification Addenda .....	10

### CONNECTIONLESS SLAVE BROADCAST

<b>Contents .....</b>	<b>14</b>
<b>1 Vol 1, Part A (Architecture) Changes: .....</b>	<b>18</b>
<b>2 Vol 1, Part B (Acronyms) Changes:.....</b>	<b>27</b>
<b>3 Vol 2, Part B (Baseband) Changes:.....</b>	<b>28</b>
<b>4 Vol 2, Part C (Link Manager Protocol) Changes: .....</b>	<b>62</b>
<b>5 Vol 2, Part E (Host Controller Interface) Changes: .....</b>	<b>64</b>
<b>6 Vol 2, Part F (Message Sequence Charts) Changes: .....</b>	<b>113</b>
<b>7 Vol 2, Part G (Sample Data) Changes: .....</b>	<b>117</b>
<b>8 Vol 3, Part C (Generic Access Profile) Changes: .....</b>	<b>119</b>

### UNENCRYPTED UNICAST CONNECTIONLESS DATA SUPPORT

<b>Contents .....</b>	<b>126</b>
<b>1 Change Instance .....</b>	<b>127</b>
1.1 Change #1 - Volume 3, Part C (GAP).....	127

### FAST ADVERTISING INTERVAL

<b>Contents .....</b>	<b>130</b>
<b>1 Change Instances .....</b>	<b>131</b>
1.1 Change #1 – Vol 3, Part C (GAP), Section 9.2.2, Non-Discoverable Mode .....	131
1.2 Change #2 – Vol 3, Part C (GAP), Section 9.3.2, Non-Connectable Mode.....	131
1.3 Change #3 – Vol 3, Part C (GAP), Section 9.3.11, Connection Establishment Timing Parameters.....	132
1.4 Change #4 – Vol 3, Part C (GAP), Appendix A (Normative): Timers and Constants .....	133

### eSCO RESERVED SLOTS CLARIFICATION

<b>Contents .....</b>	<b>135</b>
<b>1 Change Instance .....</b>	<b>136</b>
1.1 Change #1 – Vol 2, Part B, Section 8.6.3 eSCO .....	136

Part A

## **MIXING OF SPECIFICATION VERSIONS**

## CONTENTS

---

<b>1</b>	<b>Mixing of Specification Versions .....</b>	<b>7</b>
1.1	Features and their Types .....	8
1.2	Core Specification Addenda .....	10

## 1 MIXING OF SPECIFICATION VERSIONS

---

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

- All parts within a Primary Controller implementation shall be the same version of [Volume 2](#) and [Volume 6](#), and shall support at least one Type 1, Type 2, or Type 3 feature introduced in that version of the Core Specification from [Table 1.2](#).
- All parts within a Host implementation of [Volume 3](#) shall be the same version and shall support at least one Type 3 or Type 4 feature introduced in that version of the Core Specification from [Table 1.2](#).
- An AMP Controller implementation shall contain parts of [Volume 2](#) and [Volume 5](#) from the same version.
- The Primary Controller, AMP Controller, and Host may be different versions within a single implementation.

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

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

*Table 1.1: Feature type definitions.*

The outcome of mixing different Core System Packages are derived from the feature definitions in the table above:

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

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

See the [Bluetooth Brand Book](#) for specification naming requirements.

## 1.1 FEATURES AND THEIR TYPES

The following table lists the features, their types, and the version or addendum where the feature was first introduced.

Feature	Version or Addendum	Type
Basic AFH operation	1.2	1
Enhanced inquiry	1.2	1
Configuration of AFH (setting channels and enabling/disabling channel assessment)	1.2	2
Enhanced synchronization capability	1.2	2
Interlaced inquiry scan	1.2	2
Interlaced page scan	1.2	2
Broadcast encryption	1.2	2
Enhanced flow specification and flush time-out	1.2	3
Extended SCO links	1.2	3
Inquiry Result with RSSI	1.2	3
L2CAP flow and error control	1.2	4
2 Mbps EDR	2.0 + EDR	2
3 Mbps EDR	2.0 + EDR	2
3 slot packets in EDR	2.0 + EDR	2
5 slot packets in EDR	2.0 + EDR	2
2 Mbps eSCO	2.0 + EDR	2 <sup>1</sup>
3 Mbps eSCO	2.0 + EDR	2 <sup>1</sup>
3 slot packets for EDR eSCO	2.0 + EDR	2 <sup>1</sup>
Erroneous Data Reporting	2.1 + EDR	3

Table 1.2: Features and their types.

*Mixing of Specification Versions*

Feature	Version or Addendum	Type
Extended Inquiry Response	2.1 + EDR	3
Encryption Pause and Resume	2.1 + EDR	1
Link Supervision Timeout Changed Event	2.1 + EDR	3
Non-Flushable Packet Boundary Flag	2.1 + EDR	3
Sniff subrating	2.1+ EDR	3
Secure Simple Pairing	2.1.+ EDR	3
L2CAP Enhanced Retransmission Mode	Addendum 1/ 3.0 + HS	4
L2CAP Streaming Mode	Addendum 1/ 3.0 + HS	4
Enhanced Power Control	3.0 + HS	<u>2_1</u>
AMP Manager Protocol (A2MP)	3.0 + HS	4
L2CAP Enhancements for AMP	3.0 + HS	4
802.11 PAL	3.0 + HS	3
Generic Test Methodology	3.0 + HS	3
Unicast Connectionless Data	3.0 + HS	4
Low Energy Controller (PHY and LL)	4.0	3
Low Energy Host (L2CAP and Security Manager)	4.0	4
Attribute Protocol and Generic Attribute Profile	4.0	4
Appearance Data Type	Addendum 2	4
802.11n Enhancements to the 802.11 PAL	Addendum 2	3
MWS Coexistence Signaling	Addendum 3	2
<u>Connectionless Slave Broadcast</u>	<u>Addendum 4</u>	<u>3</u>
<u>Unencrypted UCD</u>	<u>Addendum 4</u>	<u>4</u>

Table 1.2: Features and their types.

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

## 1.2 CORE SPECIFICATION ADDENDA

A Core Specification Addendum (CSA) contains one or more parts of a single volume, one or more parts in multiple volumes, changes on one or more parts, or a combination of parts and changes. Addenda are used to supersede a part in a volume or may be used to add a part to a volume according to the rules in [Table 1.3](#).

Note: Each Change may contain changes and/or additions to one or more parts of the Core Specification.

Addendum	Volume and Part or Change Name	Addition/ Changes/ Replacement	Allowed Versions & Addendum	Mandatory / Optional / Conditional	Type
1	Volume 3, Part A	Replacement	2.0 + EDR, 2.1 + EDR	O	4
2	Audio Architecture HCI Changes	Change	2.1 + EDR, 3.0 + HS, 4.0	O	2
	Audio Architecture USB Changes	Change	2.1 + EDR, 3.0 + HS, 4.0	O	2
	LE Limited Discovery Time Changes	Change	4.0	C.1	4
	EIR and AD Data Types in GAP Changes	Change	4.0	C.1	4
	EIR and AD Data Types Specification	Addition	4.0	C.1	4
	Volume 5, Part A	Replacement	3.0 + HS, 4.0	O	3

*Table 1.3: Adopted core specification versions to use with addenda.*

*Mixing of Specification Versions*

Addendum	Volume and Part or Change Name	Addition/Changes/Replacement	Allowed Versions & Addendum	Mandatory / Optional / Conditional	Type
3	LE Errata	Change	4.0 with CSA2	C. <u>3</u> <u>2</u>	Multiple
	GAP Connection Parameters Changes	Change	4.0 with CSA2	C. <u>2</u> <u>1</u>	4
	GAP Authentication and Lost Bond Changes	Change	4.0 with CSA2	C. <u>2</u> <u>1</u>	4
	Common Profile and Services Error Code Range Changes	Change	4.0 with CSA2	C. <u>2</u> <u>1</u>	4
	Private Addressing Changes	Change	4.0 with CSA2	C. <u>2</u> <u>1</u>	4
	Dual Mode Addressing Changes	Change	4.0 with CSA2	C. <u>4</u> <u>3</u>	4
	MWS Coexistence Logical Signaling Specification	Addition	2.1 + EDR, 3.0 + HS, 4.0 with CSA2	O	2
	MWS Coexistence HCI	Addition	2.1 + EDR, 3.0 + HS, 4.0 with CSA2	C. <u>5</u> <u>4</u>	2
	Wireless Coexistence Interface 1 (WCI-1) Transport Layer Specification	Addition	2.1 + EDR, 3.0 + HS, 4.0 with CSA2	C. <u>5</u> <u>4</u>	2
	Wireless Coexistence Interface 2 (WCI-2) Transport Layer Specification	Addition	2.1 + EDR, 3.0 + HS, 4.0 with CSA2	C. <u>5</u> <u>4</u>	2

Table 1.3: Adopted core specification versions to use with addenda.

*Mixing of Specification Versions*

Addendum	Volume and Part or Change Name	Addition/Changes/Replacement	Allowed Versions & Addendum	Mandatory / Optional / Conditional	Type
<u>4</u>	<u>Connectionless Slave Broadcast</u>	<u>Change</u>	<u>3.0 + HS, 4.0 with CSA3</u>	<u>O</u>	<u>3</u>
	<u>Unencrypted UCD</u>	<u>Change</u>	<u>3.0 + HS, 4.0 with CSA3</u>	<u>O</u>	<u>4</u>
	<u>Fast Advertising Interval</u>	<u>Change</u>	<u>4.0 with CSA3</u>	<u>C.1</u>	<u>4</u>
	<u>eSCO Reserved Slot Clarification</u>	<u>Change</u>	<u>2.1 + EDR, 3.0 + HS, 4.0 with CSA3</u>	<u>O</u>	<u>1</u>

Table 1.3: Adopted core specification versions to use with addenda.

- C.1: Mandatory if either the Host Part of the Low Energy Core Configuration or the Host Part of the Basic Rate and Low Energy Combined Core Configuration is supported, otherwise Excluded.
- ~~C.2: Mandatory if either the Host Part of the Low Energy Core Configuration or the Host Part of the Basic Rate and Low Energy Combined Core Configuration is supported, otherwise Excluded.~~
- C.32: Mandatory if either the Host Part of the Low Energy Core Configuration, Controller Part of the Low Energy Core Configuration, Host Part of the Basic Rate and Low Energy Combined Core Configuration, or Controller Part of the Basic Rate and Low Energy Combined Core Configuration is supported, otherwise Excluded.
- C.43: Mandatory if the Host Part of the Basic Rate and Low Energy Combined Core Configuration is supported, otherwise Excluded.
- C.54: Optional if MWS Coexistence Logical Signaling is supported, otherwise Excluded.

Part B

## **CONNECTIONLESS SLAVE BROADCAST**

## CONTENTS

---

<b>1</b>	<b>Vol 1, Part A (Architecture) Changes:</b>	<b>18</b>
	<b>1.1 Overview of BR/EDR Operation</b>	[Updated] 18
	<b>1.4 Nomenclature</b>	[Updated] 20
	<b>3.1 Core Traffic Bearers</b>	[Updated] 21
	3.1.2 Unframed Data Traffic	[Updated] 21
	<b>3.2 Transport Architecture Entities</b>	[Updated] 22
	3.2.1 BR/EDR Generic Packet Structure	[Updated] 22
	3.3.1.2.1 Overview	[Updated] 22
	3.3.1.5 Synchronization Scan Channel	[New Section] 23
	3.3.1.5.1 Overview	[New Section] 23
	3.3.1.5.2 Characteristics	[New Section] 23
	3.3.1.5.3 Topology	[New Section] 23
	3.3.1.5.4 Supported Layers	[New Section] 23
	3.4.1 BR/EDR Links Supported by the Basic and Adapted Piconet Physical Channel	[Updated] 23
	3.4.1.3 Connectionless Slave Broadcast Physical Link	[New Section] 24
	<b>3.5 Logical Links and Logical Transports</b>	[Updated] 24
	3.5.4.8 Connectionless Slave Broadcast (CSB)	[New Section] 26
	3.5.5.1.4 Profile Broadcast Data (PBD) Logical Link	[New Section] 26
	4.2.1.9 Connectionless Slave Broadcast Mode	[New Section] 26
<b>2</b>	<b>Vol 1, Part B (Acronyms) Changes:</b>	<b>27</b>
<b>3</b>	<b>Vol 2, Part B (Baseband) Changes:</b>	<b>28</b>
	<b>1 GENERAL DESCRIPTION</b>	[Updated] 28
	<b>1.1 Bluetooth Clock</b>	[Updated] 28
	<b>1.3 Access Codes</b>	[Updated] 29
	<b>2 PHYSICAL CHANNELS</b>	[Updated] 29
	<b>2.1 Physical Channel Definition</b>	[Updated] 30
	2.2.5.2 Piconet physical channel re-synchronization	[Updated] 30
	<b>2.3 Adapted Piconet Physical Channel</b>	[Updated] 31
	2.3.1 Hopping Characteristics	31
	<b>2.6 Hop Selection</b>	[Updated] 32
	2.6.4.8 Synchronization train RF channels	[New Section] 32
	<b>2.7 Synchronization Scan Physical Channel</b>	[New Section] 33
	2.7.1 Hopping Characteristics	[New Section] 33
	2.7.2 Synchronization Train Procedure Timing	[New Section] 33
	2.7.3 Synchronization Scan Procedure Timing	[New Section] 34
	<b>3 PHYSICAL LINKS</b>	[Updated] 35
	<b>3.1 Link Supervision for Active and Parked Physical Links</b>	[Updated] 35
	<b>3.2 Link Supervision for Connectionless Slave Broadcast Physical Links</b>	[New Section] 36

<b>4 LOGICAL TRANSPORTS .....</b>	37
<b>4.1 General .....</b>	[Updated] 37
<b>4.2 Logical Transport Address (LT_ADDR) .....</b>	[Updated] 37
<b>4.4 Asynchronous Logical Transport .....</b>	[Updated] 38
<b>4.8 Connectionless Slave Broadcast Logical Transport .....</b>	[New Section] 38
<b>5 LOGICAL LINKS .....</b>	[Updated] 39
<b>5.6 Logical Link Priorities .....</b>	[Updated] 39
<b>5.7 Profile Broadcast Data Logical Link .....</b>	[New Section] 39
<b>6 PACKETS .....</b>	40
6.4.1 LT_ADDR .....	[Updated] 40
6.4.2 TYPE .....	[Updated] 40
6.4.3 FLOW .....	[Updated] 40
6.4.3 ARQN .....	[Updated] 41
6.4.5 SEQN .....	[Updated] 41
<b>6.5 Packet Types .....</b>	[Updated] 41
6.5.1.3 POLL Packet .....	[Updated] 43
6.5.1.4 FHS Packet .....	[Updated] 43
6.5.4 ACL Packets .....	[Updated] 44
6.5.4.7 AUX1 Packet .....	[Updated] 44
6.6.2 Asynchronous Data Field .....	[Updated] 44
<b>7 BITSTREAM PROCESSING .....</b>	[Updated] 47
<b>7.2 Data Whitening .....</b>	[Updated] 48
7.6.3 Flushing Payloads .....	[Updated] 48
7.6.5 Active Slave and Park Slave Broadcast Packets .....	[Updated] 48
<b>8 LINK CONTROLLER OPERATION .....</b>	50
<b>8.1 Overview of States .....</b>	[Updated] 50
<b>8.3 Connection Establishment Substates .....</b>	[Updated] 51
8.3.3 Page Response Substates .....	[Updated] 51
8.3.3.2 Master response substate .....	[Updated] 52
<b>8.5 Connection State .....</b>	[Updated] 52
8.6.4 Broadcast Scheme .....	[Updated] 53
8.6.8 Channel Classification and Channel Map Selection .....	[Updated] 54
<b>8.10 Connectionless Slave Broadcast Mode .....</b>	[New Section] 54
8.10.1 Connectionless Slave Broadcast Transmit Operation .....	[New Section] 54
8.10.2 Connectionless Slave Broadcast Receive Operation .....	[New Section] 56
8.10.3 AFH in Connectionless Slave Broadcast .....	[New Section] 56
<b>8.11 Synchronization Establishment Substates .....</b>	[New Section] 57
8.11.1 Synchronization Scan Substate .....	[New Section] 57
8.11.2 Synchronization Train Substate .....	[New Section] 57
<b>APPENDIX B: TIMERS .....</b>	[Updated] 60
<b>APPENDIX C: RECOMMENDATIONS FOR AFH OPERATION IN PARK, HOLD, SNIFF, AND CONNECTIONLESS SLAVE BROADCAST .....</b>	[Updated] 61

<b>AFH Operation in Connectionless Slave Broadcast .....</b>	<b>[New Section] 61</b>
<b>4 Vol 2, Part C (Link Manager Protocol) Changes:.....</b>	<b>62</b>
<b>3 DEVICE FEATURES .....</b>	<b>62</b>
<b>3.2 Feature Definitions .....</b>	<b>[Updated] 62</b>
<b>3.3 Feature Mask Definition .....</b>	<b>[Updated] 62</b>
<b>5 Vol 2, Part E (Host Controller Interface) Changes:.....</b>	<b>64</b>
<b>3 OVERVIEW OF COMMANDS AND EVENTS .....</b>	<b>[Updated] 64</b>
<b>3.6 Device Discovery .....</b>	<b>[Updated] 64</b>
<b>3.7 Connection Setup .....</b>	<b>[Updated] 65</b>
<b>3.15 Link Information .....</b>	<b>[Updated] 66</b>
<b>3.18 Alphabetical List of Commands and Events .....</b>	<b>[Updated] 68</b>
<b>3.20 Connectionless Slave Broadcast .....</b>	<b>[New Section] 69</b>
<b>5 HCI DATA FORMATS .....</b>	<b>71</b>
<b>5.3.1.1 Broadcast Connection Handles .....</b>	<b>[Updated] 71</b>
<b>6 HCI CONFIGURATION PARAMETERS .....</b>	<b>72</b>
<b>6.27 Supported Commands .....</b>	<b>[Updated] 72</b>
<b>6.36 Synchronization Train Interval .....</b>	<b>[New Section] 73</b>
<b>6.37 Synchronization Train Timeout .....</b>	<b>[New Section] 73</b>
<b>6.38 Service Data .....</b>	<b>[New Section] 73</b>
<b>7 HCI COMMANDS AND EVENTS .....</b>	<b>74</b>
<b>7.1 Link Control Commands .....</b>	<b>74</b>
<b>7.1.47 Truncated Page Command .....</b>	<b>[New Section] 74</b>
<b>7.1.48 Truncated Page Cancel Command .....</b>	<b>[New Section] 76</b>
<b>7.1.49 Set Connectionless Slave Broadcast Command .....</b>	<b>[New Section] 78</b>
<b>7.1.50 Set Connectionless Slave Broadcast Receive Command .....</b>	<b>[New Section] 82</b>
<b>7.1.51 Start Synchronization Train Command .....</b>	<b>[New Section] 86</b>
<b>7.1.52 Receive Synchronization Train Command .....</b>	<b>[New Section] 87</b>
<b>7.3 Controller &amp; Baseband Commands .....</b>	<b>89</b>
<b>7.3.69 Set Event Mask Page 2 Command .....</b>	<b>[Updated] 89</b>
<b>7.3.86 Set Reserved LT_ADDR Command .....</b>	<b>[New Section] 90</b>
<b>7.3.87 Delete Reserved LT_ADDR Command .....</b>	<b>[New Section] 92</b>
<b>7.3.88 Set Connectionless Slave Broadcast Data Command .....</b>	<b>[New Section] 93</b>
<b>7.3.89 Read Synchronization Train Parameters Command .....</b>	<b>[New Section] 95</b>
<b>7.3.90 Write Synchronization Train Parameters Command .....</b>	<b>[New Section] 97</b>
<b>7.5 Status Parameters .....</b>	<b>99</b>
<b>7.5.12 Set Triggered Clock Capture Command .....</b>	<b>[New Section] 99</b>

<b>7.7 Events .....</b>	<b>102</b>
7.7.66 Triggered Clock Capture Event .....	[New Section] 102
7.7.67 Synchronization Train Complete Event .....	[New Section] 103
7.7.68 Synchronization Train Received Event .....	[New Section] 104
7.7.69 Connectionless Slave Broadcast Receive Event .....	[New Section] 106
7.7.70 Connectionless Slave Broadcast Timeout Event .....	[New Section] 108
7.7.71 Truncated Page Complete Event .....	[New Section] 109
7.7.72 Slave Page Response Timeout Event .....	[New Section] 110
7.7.73 Connectionless Slave Broadcast Channel Map Change Event .....	[New Section] 111
7.7.74 Inquiry Response Notification Event .....	[New Section] 112
<b>6 Vol 2, Part F (Message Sequence Charts) Changes: ....</b>	<b>113</b>
<b>9 CONNECTIONLESS SLAVE BROADCAST SERVICES</b>	
.....	[New Section] 113
<b>7 Vol 2, Part G (Sample Data) Changes:.....</b>	<b>117</b>
<b>11 CONNECTIONLESS SLAVE BROADCAST SAMPLE DATA</b>	
.....	[New Section] 117
<b>8 Vol 3, Part C (Generic Access Profile) Changes:.....</b>	<b>119</b>
<b>4 MODES .....</b>	<b>[Updated] 119</b>
<b>4.4 Synchronizability Modes .....</b>	<b>[New Section] 120</b>
4.4.1 Non-synchronizable Mode .....	[New Section] 120
4.4.1.1 Definition .....	[New Section] 120
4.4.1.2 Term on UI-level .....	[New Section] 120
4.4.2 Synchronizable Mode .....	[New Section] 120
4.4.2.1 Definition .....	[New Section] 120
<b>7 ESTABLISHMENT PROCEDURES .....</b>	<b>[Updated] 121</b>
<b>7.5 Synchronization Establishment .....</b>	<b>[New Section] 122</b>
7.5.1 Purpose .....	[New Section] 122
7.5.2 Term on UI Level .....	[New Section] 122
7.5.3 Description .....	[New Section] 122
7.5.4 Conditions .....	[New Section] 123
<b>17 APPENDIX A (NORMATIVE): TIMERS AND CONSTANTS</b>	
.....	[Updated] 124

## 1 VOL 1, PART A (ARCHITECTURE) CHANGES:

### 1.1 OVERVIEW OF BR/EDR OPERATION [Updated]

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

During typical operation a physical radio channel is shared by a group of devices that are synchronized to a common clock and frequency hopping pattern. One device provides the synchronization reference and is known as the master. All other devices synchronized to a master's clock and frequency hopping pattern are known as slaves. A group of devices synchronized in this fashion form a piconet. This is the fundamental form of communication in the Bluetooth BR/EDR wireless technology.

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

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

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

Within a physical channel, a physical link is formed between a master device and slave devices. The physical link provides bidirectional packet transport between the master and slave devices, except in the case of a Connectionless Slave Broadcast physical link. In that case, the physical link provides a unidirectional packet transport from the master to a potentially unlimited number of

---

*Connectionless Slave Broadcast*

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

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

A control protocol for the baseband and physical layers is carried over logical links in addition to user data. This is the link manager protocol (LMP). Devices that are active in a piconet have a default asynchronous connection-oriented logical transport that is used to transport the LMP protocol signaling. For historical reasons this is known as the ACL logical transport. With the exception of Connectionless Slave Broadcast devices, The default ACL logical transport is the one that is created whenever a device joins a piconet. Connectionless Slave Broadcast devices may join the piconet purely to listen to Connectionless Slave Broadcast packets. In that case, a Connectionless Slave Broadcast logical transport is created (also called a CSB logical transport) and no ACL logical transport is required. For all devices, Additional logical transports may be created to transport synchronous data streams when ~~this is~~ required.

The Link Manager function uses LMP to control the operation of devices in the piconet and provide services to manage the lower architectural layers (radio layer and baseband layer). The LMP protocol is only carried on the default ACL logical transport and the default broadcast logical transport.

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

## 1.4 NOMENCLATURE [Updated]

[Insert the following new entries into Table 1.1: Nomenclature]

<u>Connectionless Slave Broadcast (CSB)</u>	A feature that enables a master to broadcast information to an unlimited number of slaves.
<u>Connectionless Slave Broadcast Receiver</u>	A Bluetooth device that receives broadcast information from a Connectionless Slave Broadcast Transmitter. The device is a slave of the piconet.
<u>Connectionless Slave Broadcast Transmitter</u>	A Bluetooth device that sends Connectionless Slave Broadcast messages for reception by one or more Connectionless Slave Broadcast receivers. The device is the master of the piconet.
<u>Profile Broadcast Data (PBD)</u>	A logical link that carries data from a Connectionless Slave Broadcast Transmitter to one or more Connectionless Slave Broadcast Receivers.
<u>Synchronization Scan Physical Channel</u>	A physical channel that enables a slave device to receive synchronization train packets from a master device.
<u>Synchronization Train</u>	A series of packets transmitted on a set of fixed frequencies that deliver sufficient information for a receiving device to start receiving corresponding Connectionless Slave Broadcast packets.

Table 1.1: Nomenclature.

### 3.1 CORE TRAFFIC BEARERS [Updated]

[Figure 3.2 updated]

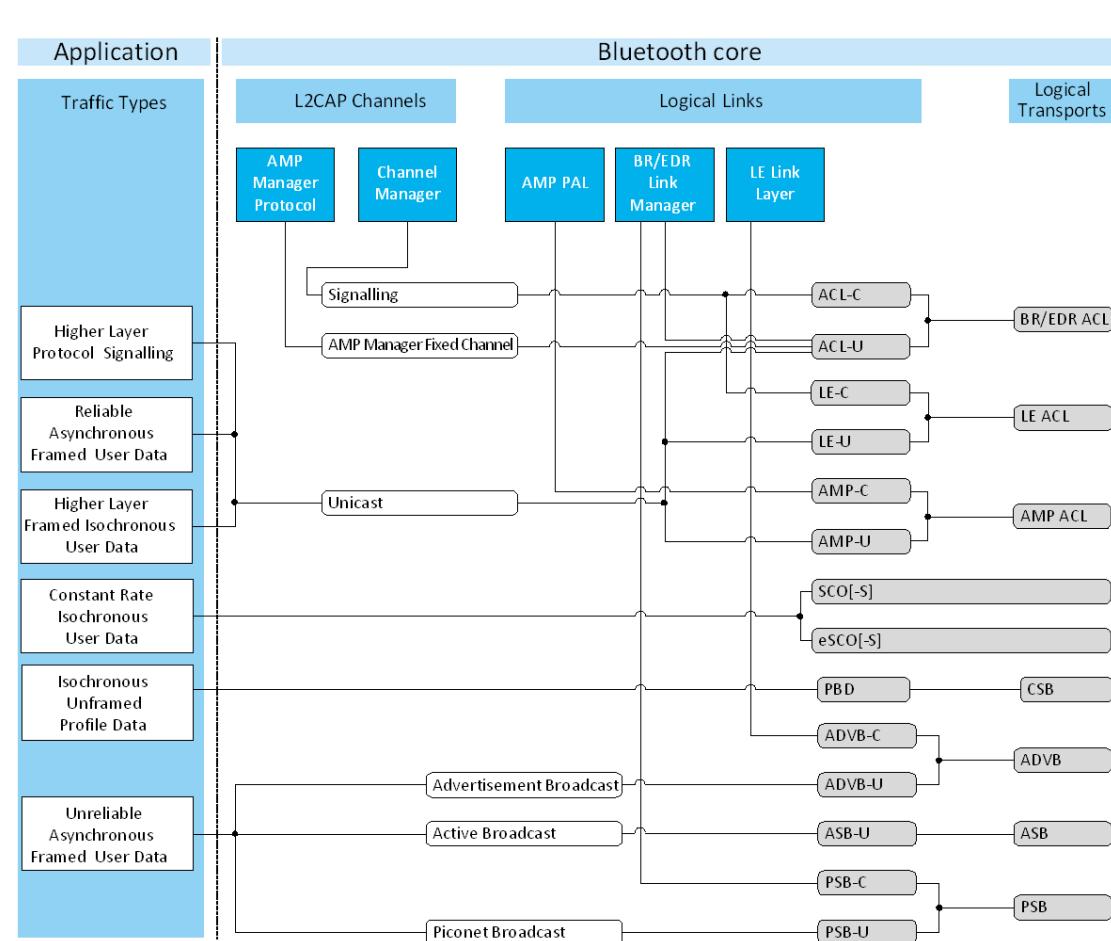


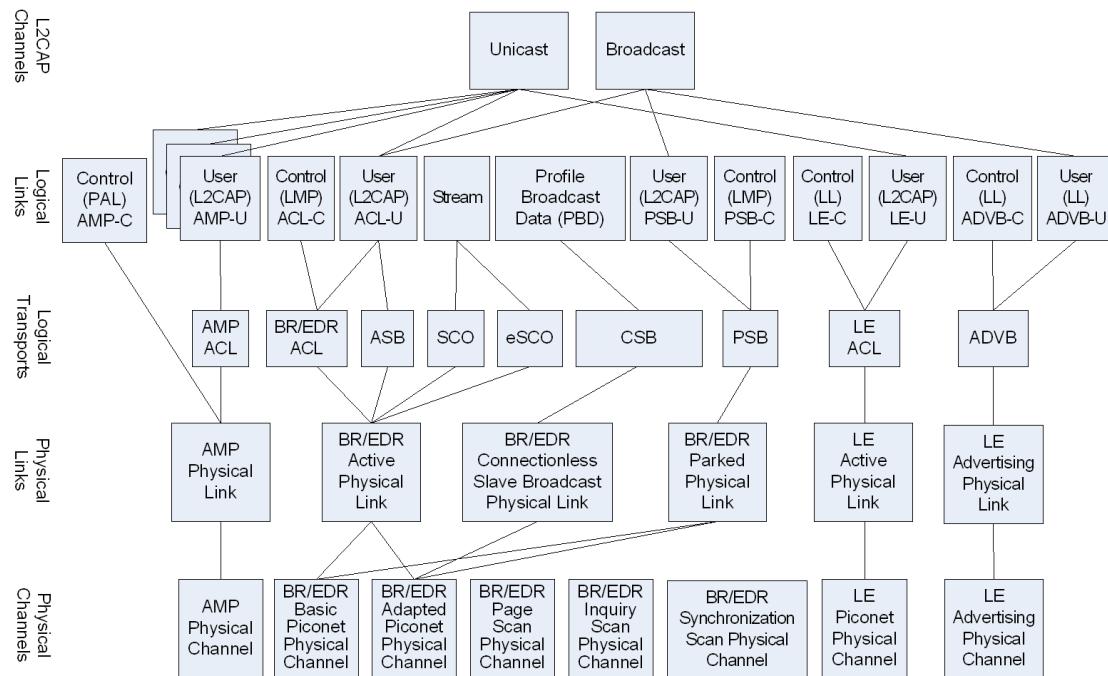
Figure 3.2: Bluetooth traffic bearers.

#### 3.1.2 Unframed Data Traffic [Updated]

[Insert new paragraph after the 2nd paragraph of this section]

..... 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 Bluetooth core system also supports the direct transport of application data using a Profile Broadcast Data (PBD) logical link. This logical link is similar to SCO-S and eSCO-S since it reserves physical channel bandwidth, provides a constant rate transport locked to the piconet clock, and transports data at fixed intervals. It does not support multiplexed logical links or any further layering within the Bluetooth core but, unlike SCO-S and eSCO-S, it supports broadcasting data from a single transmitter to many receivers.

*Connectionless Slave Broadcast***3.2 TRANSPORT ARCHITECTURE ENTITIES [Updated]****[Updated Figure 3.3]****Figure 3.3: Overview of transport architecture entities and hierarchy.****3.2.1 BR/EDR Generic Packet Structure [Updated]****[Change the last paragraph in this section as follows]**

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, and CSB logical transports the payload contains the user data for the logical link.

**3.3.1.2.1 Overview [Updated]****[Change to the last paragraph in this section as follows]**

The topology and supported layers of the adapted piconet physical channel are identical to the basic piconet physical channel with one exception: on the adapted piconet physical channel, it is possible for a single master to transmit data to an unlimited number of slaves using a single CSB logical transport. In this case, however, data is only transferred from master to slave and not from slave to master.

### 3.3.1.5 Synchronization Scan Channel [New Section]

#### **3.3.1.5.1 Overview [New Section]**

In order to receive packets sent on the CSB logical transport, a device must first obtain information about the timing and frequency channels of those packets. The synchronization scan channel is provided for that purpose. A scanning device listens for synchronization train packets on the synchronization scan channel. Once a synchronization train packet is received, the device may stop listening for synchronization train packets because it has the timing and frequency information necessary to start receiving packets sent on the CSB logical transport.

#### **3.3.1.5.2 Characteristics [New Section]**

The synchronization scan channel uses an access code derived from the Bluetooth device address of the synchronization train transmitter to identify synchronization train packets on the channel. Once a synchronization train packet is received, the scanning BR/EDR controller may start receiving packets sent on the CSB logical transport, depending on the needs of the Host and any applicable profile(s).

#### **3.3.1.5.3 Topology [New Section]**

The topology formed during this scan is transient and point-to-multipoint. There can be an unlimited number of scanning devices simultaneously receiving synchronization train packets from the same synchronization train transmitter.

#### **3.3.1.5.4 Supported Layers [New Section]**

There is a one-way flow of packets from the synchronization train transmitting device to the scanning device(s). This may be considered a temporary physical link that exists only until the scanning device receives the required information. No further architectural layers are considered to be supported.

### **3.4.1 BR/EDR Links Supported by the Basic and Adapted Piconet Physical Channel [Updated]**

The basic ~~and adapted~~-piconet physical channels supports a physical link which may be active or parked. The adapted piconet physical channel may support several physical links, including active, parked, and Connectionless Slave Broadcast. An active or parked physical link is a point-to-point link between the master and a slave. A Connectionless Slave Broadcast physical link is a point-to-multipoint link between the master and zero or more slaves. At least one physical link on the piconet physical channel ~~It~~ is always present when ~~the~~a slave is synchronized in the piconet.

### 3.4.1.3 Connectionless Slave Broadcast Physical Link [New Section]

A Connectionless Slave Broadcast physical link is present on a slave when the slave is synchronized in the piconet where a CSB logical transport exists. On a master, a Connectionless Slave Broadcast physical link is present when a CSB logical transport exists whether or not any slaves are synchronized. The Connectionless Slave Broadcast physical link is a point-to-multipoint unidirectional link between a master and zero or more slaves.

Connectionless Slave Broadcast physical links do not support power control because there is no feedback from slaves to the master. Traffic is always directed from a single master to zero or more slaves.

Connectionless Slave Broadcast packets are sent at regular intervals. To set the interval, the BR/EDR Controller selects an interval within a range requested by the Host.

## 3.5 LOGICAL LINKS AND LOGICAL TRANSPORTS [Updated]

[Add a new row to table 3.1]

Logical transport	Links supported	Supported by	Bearer	Overview
Asynchronous Connection-Oriented (ACL)	Control (LMP) ACL-C or (PAL) AMP-C  User (L2CAP) ACL-U or AMP-U	BR/EDR active physical link, BR/EDR basic or adapted piconet physical channel, AMP physical link, AMP physical channel.	BR/EDR, AMP	Reliable or time-bounded, bi-directional, point-to-point.
Synchronous Connection-Oriented (SCO)	Stream (unframed) SCO-S	BR/EDR active physical link, BR/EDR basic or adapted piconet physical channel	BR/EDR	Bi-directional, symmetric, point-to-point, AV channels. Used for 64Kb/s constant rate data.

Table 3.1: Logical transport types.

*Connectionless Slave Broadcast*

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

Table 3.1: Logical transport types.

### 3.5.4.8 Connectionless Slave Broadcast (CSB) [New Section]

The CSB logical transport is used to transport profile broadcast data to all devices connected to the Connectionless Slave Broadcast logical transport. There is no acknowledgement scheme and the traffic is unidirectional from a master to zero or more slaves. To improve reliability, profile broadcast data may be transmitted multiple times.

The CSB logical transport is created on the transmitter whenever the Connectionless Slave Broadcast is started. The CSB logical transport is created on the receiver whenever Connectionless Slave Broadcast reception is configured. The CSB logical transport is identified by a unique LT\_ADDR within the piconet that is reserved specifically for that purpose by the Connectionless Slave Broadcast Transmitter.

### 3.5.5.1.4 Profile Broadcast Data (PBD) Logical Link [New Section]

The PBD logical link is used to broadcast isochronous unframed data to multiple receivers and resides on the CSB logical transport.

### 4.2.1.9 Connectionless Slave Broadcast Mode [New Section]

Connectionless slave broadcast mode allows a piconet master to transmit profile broadcast data to any number of connected slave devices using the BR/EDR adapted piconet physical channel. To enter this mode, the master reserves a specific logical transport as the dedicated CSB logical transport and starts broadcasting data using the Connectionless Slave Broadcast physical link and the synchronization train procedure. A single Profile Broadcast Data logical link is defined, which carries profile broadcast data using the Connectionless Slave Broadcast logical transport. The profile broadcast data is unframed and bypasses L2CAP.

To receive the Connectionless Slave Broadcast packets, a device must connect with the Connectionless Slave Broadcast Transmitter which has already established a CSB logical transport. To connect, a device follows the Synchronization Scan procedure to obtain the time schedule of the physical link and then starts receiving the Connectionless Slave Broadcast packets. Once connected, Connectionless Slave Broadcast receivers can receive profile broadcast data on the dedicated CSB logical transport and PBD logical link.

## 2 VOL 1, PART B (ACRONYMS) CHANGES:

---

[Add the following new rows to Table 1.1]

Acronym or abbreviation	Writing out in full	Which means
<u>CSB</u>	<u>Connectionless Slave Broadcast</u>	<u>The logical transport enabled by the Connectionless Slave Broadcast feature.</u>
<u>PBD</u>	<u>Profile Broadcast Data</u>	<u>The name of the logical link enabled by the Connectionless Slave Broadcast feature.</u>
<u>STP</u>	<u>Synchronization Train Packet</u>	<u>Packets transmitted using the Synchronization Train feature.</u>

Table 1.1: Acronyms and Abbreviations.

## 3 VOL 2, PART B (BASEBAND) CHANGES:

---

### 1 GENERAL DESCRIPTION [Updated]

---

[Add changes to the 2<sup>nd</sup> paragraph as follows]

This part specifies the normal operation of a Bluetooth baseband.

The Bluetooth system provides a point-to-point connection or a point-to-multipoint connection, see (a) and (b) in [Figure 1.1 on page 65](#). 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, m~~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. An unlimited number of slaves can receive data on the physical channel carrying the Connectionless Slave Broadcast physical link.

#### 1.1 BLUETOOTH CLOCK [Updated]

[Add changes to the 4th paragraph of this section as follows]

CLKN is the native clock and shall be the reference to all other clock appearances. In STANDBY and in Park, Hold, ~~and Sniff, and Connectionless Slave Broadcast~~ modes 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.

## 1.3 ACCESS CODES [Updated]

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 110](#):

- 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**, **synchronization train substate**, and **synchronization scan substate**, 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 signaling.

## 2 PHYSICAL CHANNELS [Updated]

[Add changes to the 3rd paragraph as follows]

~~Four~~Five 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~~Other physical channels are used for discovering (the inquiry scan channel) and connecting (the page scan channel) Bluetooth devices. The synchronization scan physical channel is used by devices to obtain timing and frequency information about the Connectionless Slave Broadcast physical link.

## 2.1 PHYSICAL CHANNEL DEFINITION [Updated]

Physical channels, apart from the synchronization scan physical channel, 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 synchronization scan physical channel uses a set of fixed RF channels. 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 83](#)). The maximum hop rate is 1600 hops/s in the **CONNECTION** state, the synchronization train substate, and the synchronization scan substate 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
- synchronization scan physical channel

### 2.2.5.2 Piconet physical channel re-synchronization [Updated]

[Add changes to the 1st paragraph of this section as follows]

In the piconet physical channel, a slave can lose 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 can occur in sniff, hold, ~~and Park, and Connectionless Slave Broadcast~~, 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 (except for a Connectionless Slave Broadcast slave device which shall listen for the master but does not send information). In this case, the length of the search window in the slave device may be increased from 20  $\mu$ s to a larger value X  $\mu$ s as illustrated in [Figure 2.5 on page 76](#). 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.

## 2.3 ADAPTED PICONET PHYSICAL CHANNEL [Updated]

For devices that enter connectionless slave broadcast mode, the device that transmits Connectionless Slave Broadcast packets is the master of the piconet and any device that receives Connectionless Slave Broadcast packets is a slave of the piconet.

### 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 83](#).

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 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.

## 2.6 HOP SELECTION [Updated]

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.

In addition, a set of synchronization train RF channels with 3 fixed frequencies is defined.

### 2.6.4.8 Synchronization train RF channels [New Section]

The synchronization train and synchronization scan use RF channels from the set of three fixed RF Channels with indices 0, 24, and 78.

## 2.7 SYNCHRONIZATION SCAN PHYSICAL CHANNEL [New Section]

The synchronization scan physical channel enables devices to receive synchronization train packets.

### 2.7.1 Hopping Characteristics [New Section]

When a device enters the **synchronization scan** substate, it shall scan the synchronization train RF channels using the timing defined in [section 2.7.3](#). Each individual scan window shall use a different RF channel than the previous two scan windows.

The synchronization scan physical channel shall use all of the synchronization train RF channels defined in [section 2.6.4.8](#).

### 2.7.2 Synchronization Train Procedure Timing [New Section]

The CSB master shall use the synchronization train procedure only when connectionless slave broadcast mode is enabled. During the synchronization train procedure, the master shall attempt to transmit synchronization train packets on all of the RF Channels specified in [section 2.6.4.8](#). The transmission of synchronization train packets on each RF Channel is independent of the transmission on other RF Channels. For each RF Channel, the master shall:

1. Establish synchronization train events that are separated by  $T_{Sync\_Train\_Interval}$  where  $T_{Sync\_Train\_Interval}$  is selected by the Controller from a range provided by the Host ([see HCI \[Part E\] section 7.3.90 Write Synchronization Train Parameters Command \[New Section\]](#)).
2. Attempt to send a synchronization train packet between each pair of synchronization train events.
3. In the absence of scheduling conflicts, delay the start of the synchronization train packet transmission by  $T_{Sync\_Train\_Delay}$  slots from the synchronization train event, where  $T_{Sync\_Train\_Delay}$  is a pseudorandom number between 0 and  $T_{Sync\_Train\_Delay\_Max}$  inclusive. Each value of  $T_{Sync\_Train\_Delay}$  shall be an even integer.  $T_{Sync\_Train\_Delay\_Max}$  shall equal 16 slots. Each synchronization packet transmission shall start where  $CLK[1:0]=00b$ .
4. If the transmission of the synchronization train packet conflicts with the timing of higher priority packets, the actual delay may be adjusted to avoid the conflict. The actual delay should stay within the range of 0 to  $T_{Sync\_Train\_Delay\_Max}$  inclusive and shall not be greater than or equal to  $T_{Sync\_Train\_Interval}$ . If it is not possible to transmit the complete packet before the next synchronization train event, it shall not be transmitted. Note that increasing the delay beyond the recommended range (0 to

$T_{Sync\_Train\_Delay\_Max}$ ) increases the chance that the scanning device will miss the packet.

5. There shall be no more than one synchronization train packet transmitted between any two consecutive synchronization train events.
6. The actual delays used shall not all be the same for any three consecutive synchronization train packets (though any two may be).

Note: Subject to the above requirements, synchronization train events on different RF Channels may be managed separately or may be co-ordinated using the same value of  $T_{Sync\_Train\_Delay}$ .

**Figure 2.21** shows the timing relationship of consecutive synchronization train packets on a single RF Channel.

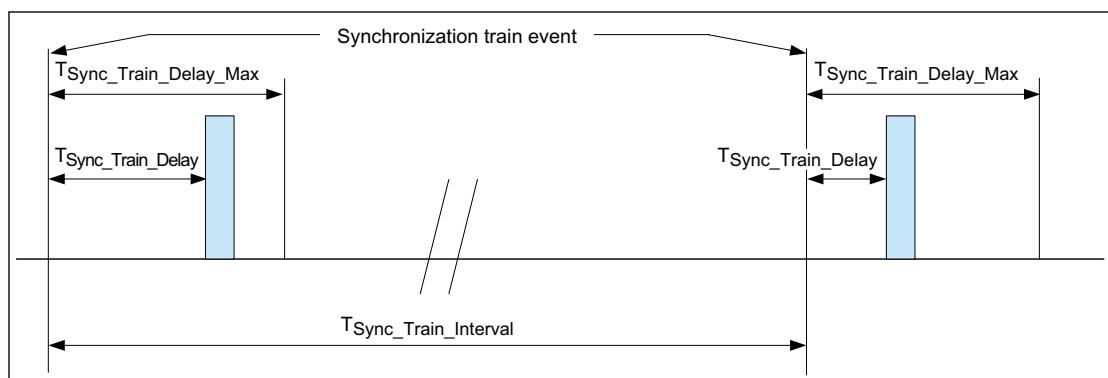


Figure 2.21: Synchronization train packet timing on a single channel.

### 2.7.3 Synchronization Scan Procedure Timing [New Section]

During the Synchronization Scan procedure, a device performs synchronization scans on the synchronization train RF channels. During each scan window, the device listens for the duration of  $T_{Sync\_Scan\_Window}$  (see [HCI \[Part E\] section 7.1.52 Receive Synchronization Train Command \[New Section\]](#)). The RF Channel for each scan window shall be selected as specified in [section 2.7.1](#). Each scan window should be continuous and not interrupted by other activities. The interval between the start of consecutive scan windows shall be  $\leq T_{Sync\_Scan\_Interval}$  (see [HCI \[Part E\] section 7.1.52 Receive Synchronization Train Command \[New Section\]](#)). This timing relationship is shown in [Figure 2.22](#).

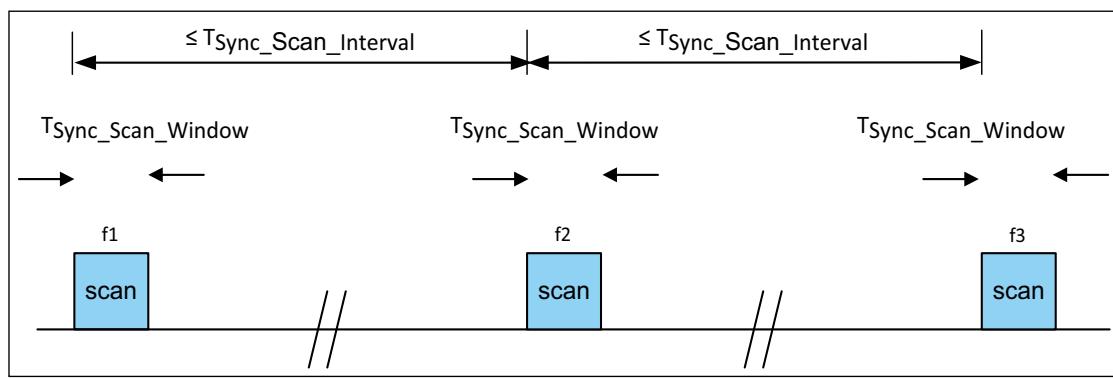
*Connectionless Slave Broadcast*

Figure 2.22: Synchronization scan timing.

### 3 PHYSICAL LINKS [Updated]

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 Connectionless Slave Broadcast physical link is associated with the BR/EDR adapted piconet physical channel, a single logical transport (the CSB logical transport), and does not support Link Manager Protocol. For link supervision on Connectionless Slave Broadcast physical links, see Section 3.2. Multi-slot packets on Connectionless Slave Broadcast physical links are controlled by the Host and specified at the profile level.

TheOther than the Connectionless Slave Broadcast physical link, common properties of physical links are:

- Power control (see Link Manager Protocol [Section 4.1.3 on page 230](#))
- Link supervision (see [Section on page 35](#) and Link Manager Protocol [Section 4.1.6 on page 239](#))
- Encryption (see Security [\[Part H\] Section 4 on page 1072](#) and Link Manager Protocol [\[Part C\] Section 4.2.5 on page 255](#))
- Channel quality-driven data rate change (see Link Manager Protocol [Section 4.1.7 on page 240](#))
- Multi-slot packet control (see Link Manager Protocol [Section 4.1.10 on page 244](#))

#### 3.1 LINK SUPERVISION FOR ACTIVE AND PARKED PHYSICAL LINKS [Updated]

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 can 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 97](#)) or parked member address (see [Section 4.7.1 on page 105](#)) is reassigned to another slave.

To be able to detect link loss, both the master and the slave shall use a link supervision timer,  $T_{\text{supervision}}$ . Upon reception of a valid packet header with one of the slave's addresses (see [Section 4.2 on page 97](#)) on the physical link, the timer shall be reset. If at any time in **CONNECTION** state, the timer reaches the  $\text{supervisionTO}$  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,  $\text{supervisionTO}$ , 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.

### 3.2 LINK SUPERVISION FOR CONNECTIONLESS SLAVE BROADCAST PHYSICAL LINKS [New Section]

For Connectionless Slave Broadcast physical links, only the slave side monitors the link. To detect link loss, the slave shall use a link supervision timer,  $T_{\text{CSB\_Supervision}}$ . Each Connectionless Slave Broadcast slave shall reset the timer upon reception of a Connectionless Slave Broadcast packet with a valid packet header. If at any time in connectionless slave broadcast mode of the **CONNECTION** state, the timer reaches the  $\text{CSB\_supervisionTO}$  value, the connection shall be considered disconnected.

For each slave, the timeout period,  $\text{CSB\_supervisionTO}$ , shall be provided by the Host.

## 4 LOGICAL TRANSPORTS

---

### 4.1 GENERAL [Updated]

Between master and slave(s), different types of logical transports may be established. ~~Five~~<sub>Six</sub> 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
- Connectionless Slave Broadcast (CSB) 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.

The CSB logical transport is used by a master to send profile broadcast data to zero or more slaves.

### 4.2 LOGICAL TRANSPORT ADDRESS (LT\_ADDR) [Updated]

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 ASB and PSB broadcast messages. The CSB logical transport uses a single non-zero LT\_ADDR. 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 115](#)). 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.

The primary LT\_ADDR shall be assigned by the master to the slave when the slave is activated. This is either at connection establishment, or 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.

At any given time an LT\_ADDR (other than the special case of the all-zero LT\_ADDR) is either unused or is used for exactly one of the three purposes of the primary address for a slave, a secondary address for eSCO traffic, or for a CSB logical transport. Therefore allocating a secondary LT\_ADDR for an eSCO logical transport, or reserving an LT\_ADDR for the CSB logical transport, reduces the maximum number of active slaves possible in the piconet.

## 4.4 ASYNCHRONOUS LOGICAL TRANSPORT [Updated]

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 (LT\_ADDR=0) are considered as broadcast packets and should be read by every slave except slaves with only a CSB logical transport. If there is no data to be sent on the ACL logical transport and no polling is required, no transmission is required.

## 4.8 CONNECTIONLESS SLAVE BROADCAST LOGICAL TRANSPORT [New Section]

The CSB logical transport is used to transport profile broadcast data from a master to multiple slaves. There is no acknowledgement scheme and the traffic is unidirectional. Note: The CSB logical transport is not used for L2CAP connection-oriented channels, L2CAP control signaling, or LMP control signaling.

The CSB logical transport is unreliable. To improve reliability each packet payload may be transmitted a number of times.

## 5 LOGICAL LINKS [Updated]

FiveSix 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)
- Profile Broadcast Data (PBD)

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 PBD logical link is used to carry profile broadcast data. 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, it 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. The PBD logical link is carried by the CSB logical transport.

### 5.6 LOGICAL LINK PRIORITIES [Updated]

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. The ACL-C, SCO-S, and eSCO-S logical links should have priority over traffic on the PBD logical link.

### 5.7 PROFILE BROADCAST DATA LOGICAL LINK [New Section]

The PBD logical link carries profile broadcast data. Messages are not fragmented and always use LLID start code 10.

## 6 PACKETS

---

### 6.4.1 LT\_ADDR [Updated]

The 3-bit LT\_ADDR field contains the logical transport address for the packet (see [Section 4.2 on page 97](#)). This field indicates the destination slave ([or slaves in the case of a broadcast](#)) 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 [Updated]

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, or a CSB logical transport](#). Second, it shall be determined whether Enhanced Data Rate has been enabled for the logical transport (ACL or eSCO) indicated by LT\_ADDR. It can then 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](#)). 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 117](#), each packet type is described in more detail.

### 6.4.3 FLOW [Updated]

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. [The FLOW bit is not used on the CSB logical transport and shall be set to zero on transmission and ignored upon receipt](#).

### 6.4.3 ARQN [Updated]

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 141](#) for initialization and usage of this bit.

The ARQN bit is not used on the CSB logical transport and shall be set to zero on transmission and ignored upon receipt.

### 6.4.5 SEQN [Updated]

The SEQN bit provides a sequential numbering scheme to order the data packet stream. See [Section 7.6.2 on page 144](#) for initialization and usage of the SEQN bit. For broadcast packets, a modified sequencing method is used, see [Section 7.6.5 on page 148](#).

The SEQN bit is not used on the CSB logical transport and shall be set to zero on transmission and ignored upon receipt.

## 6.5 PACKET TYPES [Updated]

The packets used on the piconet are related to the logical transports they are used in. ~~Three~~<sup>Four</sup> logical transports with distinct packet types are defined (see [Section 4 on page 97](#)): the SCO logical transport, the eSCO logical transport, ~~and~~ the ACL logical transport, and the CSB 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](#) summarizes the packets defined for the SCO, eSCO, ~~and~~ ACL, and CSB logical transport types.

All packet types with a payload shall use GFSK modulation unless specified otherwise in the following sections.

ACL logical transports Enhanced Data Rate packet types are explicitly selected via LMP using the *packet\_type\_table* (ptt) parameter. eSCO Enhanced Data Rate packet types are selected when the eSCO logical transport is established. Enhanced Data Rate packet types for the CSB logical transport are selected when the CSB logical transport is established.

*Connectionless Slave Broadcast*

<u>Segment</u>	<u>TYPE code</u> $b_3b_2b_1b_0$	<u>Slot occupancy</u>	<u>SCO logical transport (1 Mbps)</u>	<u>eSCO logical transport (1 Mbps)</u>	<u>eSCO logical transport (2-3 Mbps)</u>	<u>ACL logical transport (1 Mbps) ptt=0</u>	<u>ACL logical transport (2-3 Mbps) ptt=1</u>	<u>CSB logical transport (1 Mbps)</u>	<u>CSB logical transport (2-3 Mbps)</u>
<b>1</b>	0000	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	0001	1	POLL	POLL	POLL	POLL	POLL	reserved	reserved
	0010	1	FHS	reserved	reserved	FHS	FHS	reserved	reserved
	0011	1	DM1	reserved	reserved	DM1	DM1	DM1	DM1
	0100	1	undefined	undefined	undefined	DH1	2-DH1	DH1	2-DH1
<b>2</b>	0101	1	HV1	undefined	undefined	undefined	undefined	undefined	undefined
	0110	1	HV2	undefined	2-EV3	undefined	undefined	undefined	undefined
	0111	1	HV3	EV3	3-EV3	undefined	undefined	undefined	undefined
	1000	1	DV	undefined	undefined	undefined	3-DH1	undefined	3-DH1
	1001	1	undefined	undefined	undefined	AUX1	AUX1	undefined	undefined
<b>3</b>	1010	3	undefined	undefined	undefined	DM3	2-DH3	DM3	2-DH3
	1011	3	undefined	undefined	undefined	DH3	3-DH3	DH3	3-DH3
	1100	3	undefined	EV4	2-EV5	undefined	undefined	undefined	undefined
	1101	3	undefined	EV5	3-EV5	undefined	undefined	undefined	undefined
	1110	5	undefined	undefined	undefined	DM5	2-DH5	DM5	2-DH5
<b>4</b>	1111	5	undefined	undefined	undefined	DH5	3-DH5	DH5	3-DH5

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

### **6.5.1.3 POLL Packet [Updated]**

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. POLL packets shall not be sent on a Connectionless Slave Broadcast logical transport.

### **6.5.1.4 FHS Packet [Updated]**

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 6.9** 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 frequency hop synchronization before the piconet channel has been established, or when an existing piconet changes to a new piconet. However, the FHS packet is not used by Connectionless Slave Broadcast slaves for frequency hop synchronization with Connectionless Slave Broadcast masters.

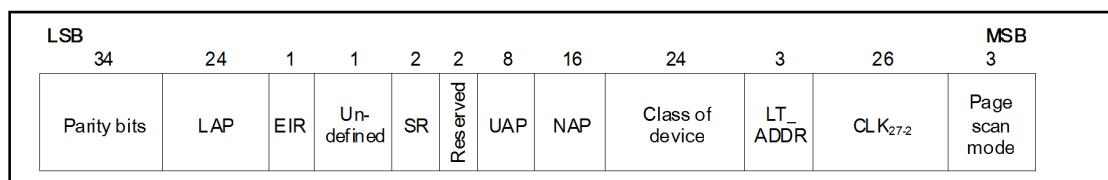


Figure 6.9: Format of the FHS payload.

### 6.5.4 ACL Packets [Updated]

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

Seven packet types are defined for Basic Rate operation: DM1, DH1, DM3, DH3, DM5, DH5 and AUX1. Six additional packets are defined for Enhanced Data Rate operation: 2-DH1, 3-DH1, 2-DH3, 3-DH3, 2-DH5 and 3-DH5.

#### 6.5.4.7 AUX1 Packet [Updated]

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 or PBD logical links. An AUX1 packet may be discarded.

### 6.6.2 Asynchronous Data Field [Updated]

[Add changes to “# 3. Payload header” (beginning with Table 6.6) as follows]

LLID code $b_1 b_0$	Logical Link	Information
00	NA	undefined
01	ACL-U	Continuation fragment of an L2CAP message
10	ACL-U PBD	Start of an L2CAP message or no fragmentation <u>Profile broadcast data, no fragmentation</u>
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. The first fragment of an L2CAP message sent over HCI is identified by having a Packet\_Boundary\_Flag value of 00 or 10 both of which are mapped to LLID code 10. If there is no fragmentation, code 10 shall be used for every packet. ACL packets used over the PBD logical link do not use fragmentation. For ACL packets used over the PBD logical link, LLID code 10 shall be used by the transmitter. Any ACL packet used over the PBD logical link with LLID not equal to 10 shall be

*Connectionless Slave Broadcast*

ignored by the receiver. 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-U 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 ( $7 \times 256$  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.

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](#)). 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), see [Table 6.7](#). On the ACL-C logical link, no flow control is applied and the payload FLOW bit shall always be set to one. On the PBD logical link, no flow control is applied and the payload FLOW bit shall always be set to zero.

LLID code $b_1b_0$	Usage and semantics of the ACL payload header FLOW bit
00	Not defined, reserved for future use.

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

*Connectionless Slave Broadcast*

LLID code $b_1b_0$	<b>Usage and semantics of the ACL payload header FLOW bit</b>
01 <del>or 10</del>	Flow control of the ACL-U channel (L2CAP messages)
10	<u>For the ACL-U logical link, flow control of the ACL-U channel (L2CAP messages)</u> <u>For the PBD logical link, always set FLOW=0 on transmission and ignore the bit on reception</u>
11	Always set FLOW=1 on transmission and ignore the bit on reception

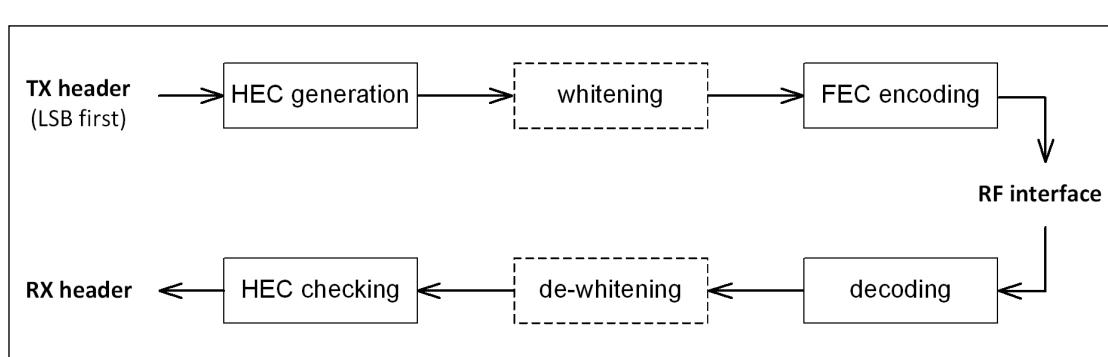
Table 6.7: Use of payload header flow bit on the logical links.

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 6.12](#) and [Figure 6.13](#), 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 to the left from the right-most end of the second byte in the payload header.

## 7 BITSTREAM PROCESSING [Updated]

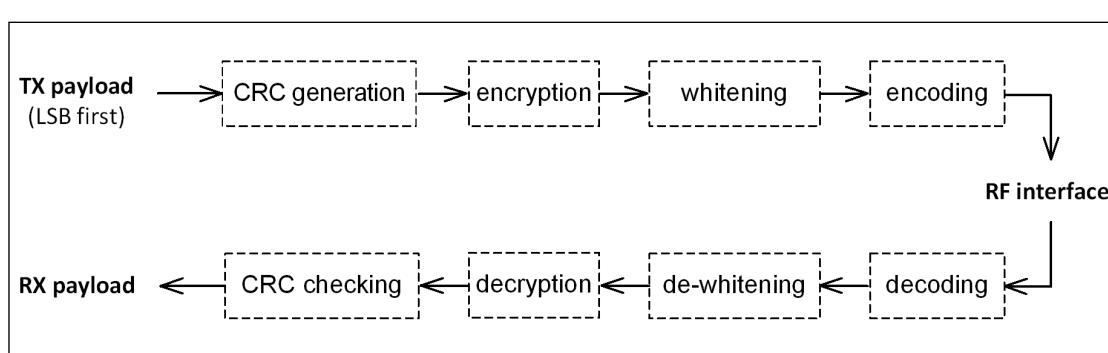
Bluetooth devices shall use the bitstream processing schemes as defined in the following sections.

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](#) shows the processes carried out for the packet header both at the transmit and the receive side. All header bit processes are mandatory [except that whitening and de-whitening shall not be performed on synchronization train packets](#). In [Figure 7.1](#) processes not performed on all packet types are indicated by dashed blocks.



[Figure 7.1: Header bit processes.](#)

[Figure 7.2](#) 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 138, are mandatory for every payload; except synchronization train payloads, where they are forbidden.](#) All other processes are optional and depend on the packet type (see [Section 6.6 on page 127](#)) and whether encryption is enabled. In [Figure 7.2](#), optional processes are indicated by dashed blocks.



[Figure 7.2: Payload bit processes.](#)

## 7.2 DATA WHITENING [Updated]

[Add changes to the first two paragraphs of this section as follows]

Synchronization train packets shall not be scrambled with a data whitening word. Before transmission of all other packets, 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 of scrambled packets shall be descrambled using the same whitening word generated in the recipient. The descrambling shall be performed after FEC decoding.

### 7.6.3 Flushing Payloads [Updated]

[Add changes to the first paragraph of this section as follows]

In ACL logical transports, 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.

[Add changes to the final paragraph of this section as follows]

In eSCO, packets shall be automatically flushed at the end of the eSCO window. Connectionless Slave Broadcast packets are transmitted at each scheduled Connectionless Slave Broadcast Instant. If no new data is available, the Connectionless Slave Broadcast Transmitter shall transmit a packet with the last available data in the payload.

### 7.6.5 Active Slave and Park Slave Broadcast Packets [Updated]

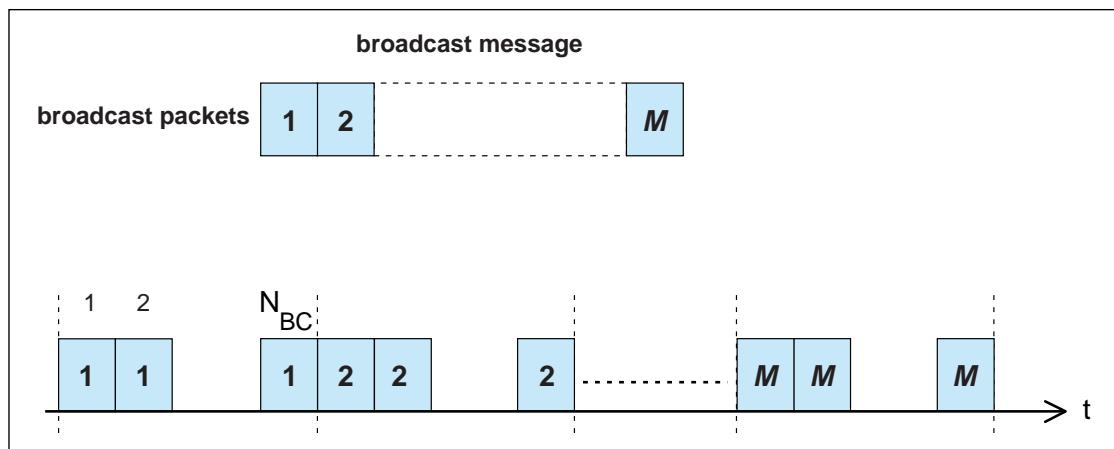
ASB and PSB broadcast packets are packets transmitted by the master to all the slaves simultaneously (see [Section 8.6.4](#)) If multiple hop sequences are being used each transmission may only be received by some of the slaves. In this case the master shall repeat the transmission on each hop sequence. An ASB or PSB broadcast packet shall be indicated by the all-zero LT\_ADDR (note; the FHS packet and the extended inquiry response packet are the only packets which may have an all-zero LT\_ADDR but are not broadcast packets). Broadcast packets shall not be acknowledged (at least not at the LC level).

*Connectionless Slave Broadcast*

Since broadcast messages are not acknowledged, each broadcast packet is transmitted at least a fixed number of times. An ASB or PSB 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 149](#). Optionally, an ASB or PSB 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](#).

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.

ASB and PSB Bbroadcast 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. ASB and PSB Bbroadcast 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 [Table 6.6 on page 130](#). Only repetitions of the L2CAP continuation packets shall be filtered out.



## 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 [Updated]

Figure 8.1 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~~nine substates, **page**, **page scan**, **inquiry**, **inquiry scan**, **synchronization train**, **synchronization scan**, **master response**, **slave response**, and **inquiry response**. Note that the **master response**, **slave response**, and **inquiry response** substates are not shown in the simplified figure below. 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).

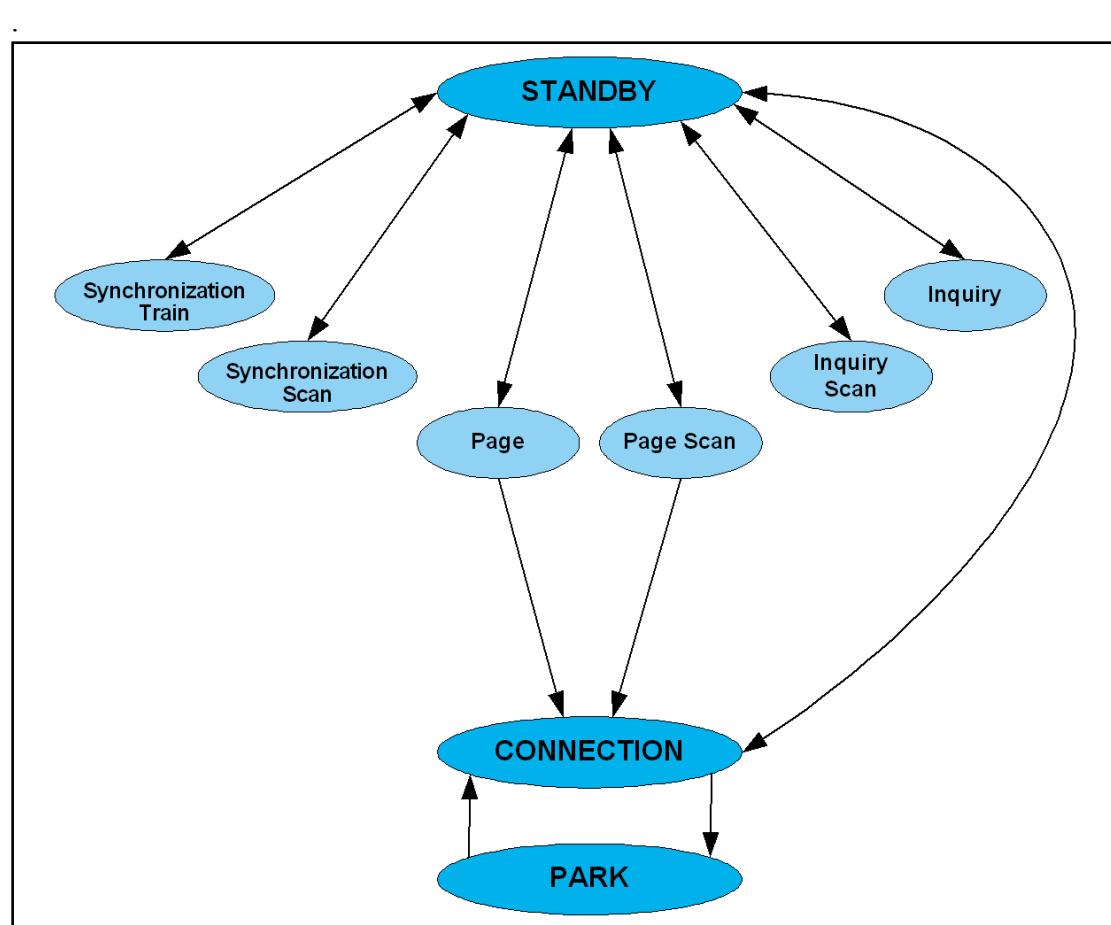


Figure 8.1: State diagram of link controller.

## 8.3 CONNECTION ESTABLISHMENT SUBSTATES [Updated]

In order to establish new connections the paging procedure or the synchronization scan procedure is used. Only the Bluetooth device address is required to set up a connection using the paging procedure. Knowledge about the clock, obtained from the inquiry procedure (see [Section 8.4 on page 160](#)) or from a previous connection with this device, and the page scanning mode of the other device will accelerate the setup paging procedure. A device that establishes a connection carries out using a page procedure and will automatically become the master of the connection. A device that establishes a connection using a synchronization scan procedure does so in two steps. First, from within the synchronization scan substate, the BR/EDR Controller follows the synchronization scan procedure to collect clock, packet timing, and AFH channel map information from the master and then transitions to the STANDBY state. Second, as commanded by the Host, the BR/EDR Controller transitions from the STANDBY state to the connectionless slave broadcast mode of the CONNECTION state.

The truncated page procedure is a modification of the paging procedure and is used to deliberately generate a page response timeout on the slave device.

### 8.3.3 Page Response Substates [Updated]

[Add following text at the end of the section, after Figure 8.4]

In the case of the truncated page procedure, the messaging stops after step 2 in the sequence shown in [Table 8.3](#). This procedure is illustrated in [Figure 8.5](#) and [Figure 8.6](#).

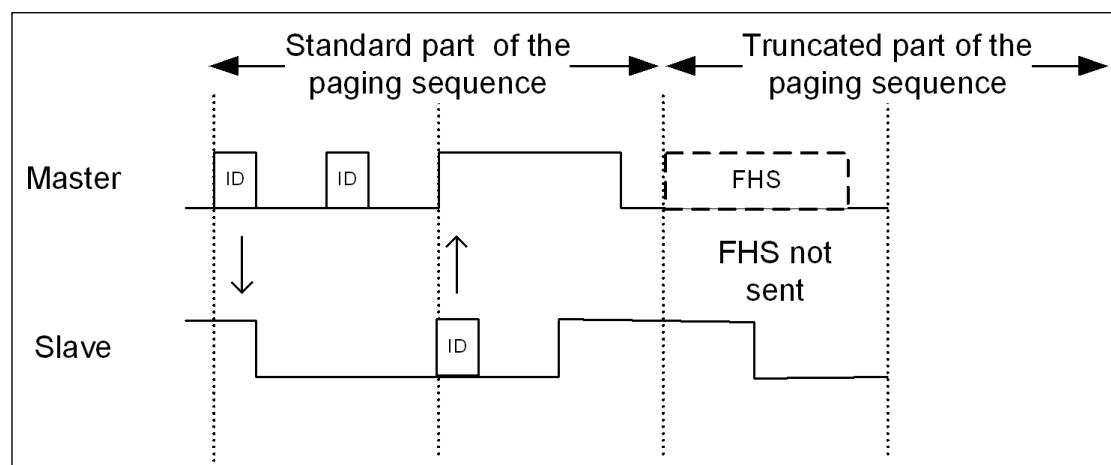


Figure 8.5: First half slot truncated page.

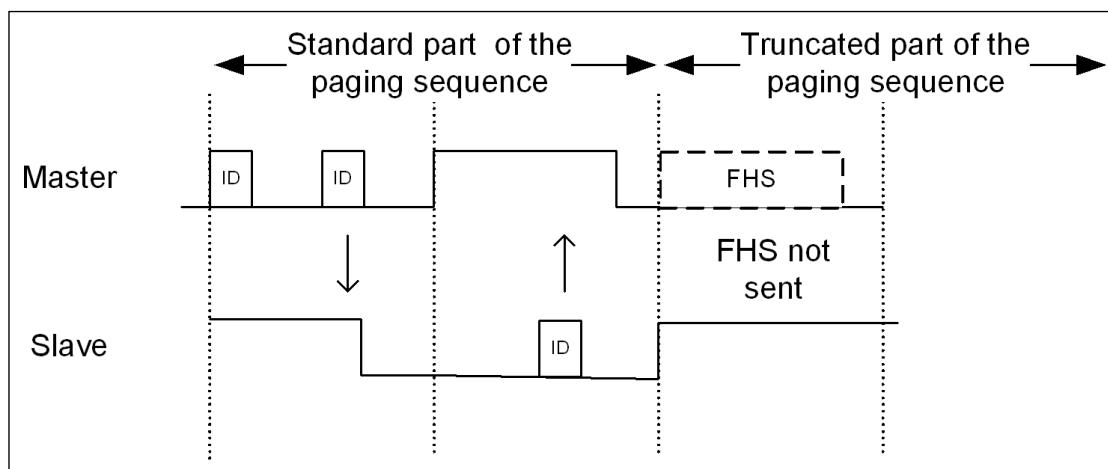


Figure 8.6: [Second half slot truncated page](#).

#### 8.3.3.2 Master response substate [Updated]

When the master has received a slave page response message in step 2, [and the master is not executing the truncated page procedure](#), 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 157](#) 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 79](#).

## 8.5 CONNECTION STATE [Updated]

In the **CONNECTION** state, the connection has been established and packets can be sent back and forth, [with the exception of connectionless slave broadcast mode, in which case packets are only sent from the master](#). 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*.

[There are two ways to enter the CONNECTION state. A device can transition from the page/page scan substates to the CONNECTION state or directly from the STANDBY state to the connectionless slave broadcast mode of the CONNECTION state.](#)

If a device enters from the page/page scan substates, 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 564.

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 8.7).

If a device enters the connectionless slave broadcast mode of the **CONNECTION** state, the master starts by sending packets on the CSB logical transport and the slave starts by receiving packets sent on the CSB logical transport.

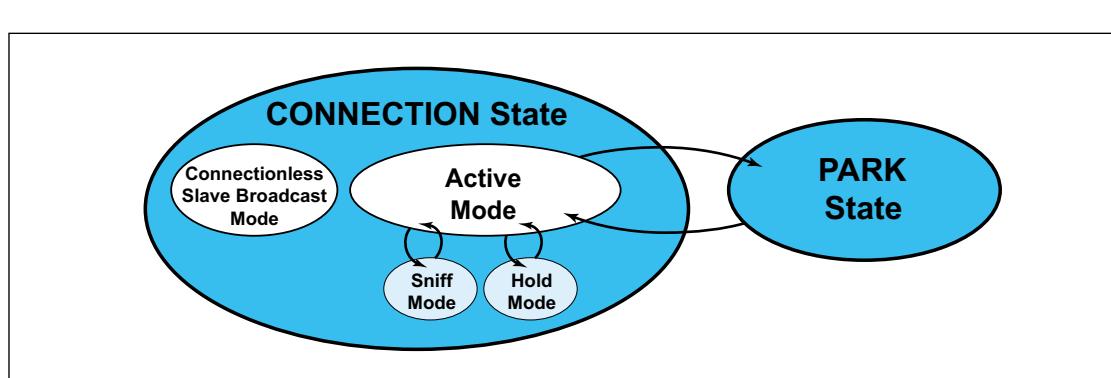


Figure 8.7: Connection state.

#### 8.6.4 Broadcast Scheme [Updated]

The master of the piconet can broadcast messages to all slaves on the ASB-U, PSB-C, and PSB-U logical transports. An ASB or PSB 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).

The Broadcast LT\_ADDR shall use a ptt=0.

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 148](#).

In order to support the **PARK** state (as described in [Section 8.9 on page 183](#)), 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 183](#).

The master can broadcast messages to multiple slaves in connectionless slave broadcast mode. For this purpose, the master shall reserve an LT\_ADDR for use only by Connectionless Slave Broadcast traffic. In connectionless slave broadcast mode, the master transmits packets at specified intervals. If the Host has not provided new data in time for a particular interval, then the master shall resend the most recently available data from the Host.

### 8.6.8 Channel Classification and Channel Map Selection [Updated]

[Add the following text to the end of this section]

For all devices that support Synchronizable mode, the RF channel indices used for the Synchronization Train (see [Section 2.6.4.8](#)) shall be marked as unused in the AFH\_channel\_map for all logical links.

## 8.10 CONNECTIONLESS SLAVE BROADCAST MODE [New Section]

In connectionless slave broadcast mode, the master broadcasts packets to zero or more slaves.

### 8.10.1 Connectionless Slave Broadcast Transmit Operation [New Section]

In connectionless slave broadcast mode, the master transmits packets on the CSB logical transport at intervals requested by the Host in master-to-slave transmission slots.

If the Host has not yet provided the BR/EDR Controller with any data packets to transmit since enabling the broadcast, or if the length of the data is zero, the BR/EDR Controller shall transmit NULL packets. This is different than the case where the Host has provided data since enabling the broadcast but has not provided new data since the previous broadcast packet. In that case, the BR/EDR Controller resends the most recent data.

The Host may provide Connectionless Slave Broadcast data through HCI commands. Because HCI commands are limited to 255 bytes, a single command cannot carry the maximum payloads allowed by larger packets, such

*Connectionless Slave Broadcast*

as DH5. Therefore, HCI commands for Connectionless Slave Broadcast allow fragmentation of large payloads at the HCI level. The BR/EDR Controller shall assemble all HCI fragments of a packet before transmission and shall not transmit incomplete packets. Until such assembly is complete, the controller shall continue to transmit the previous data (if any).

**Figure 8.21** shows the timing of Connectionless Slave Broadcast packets.

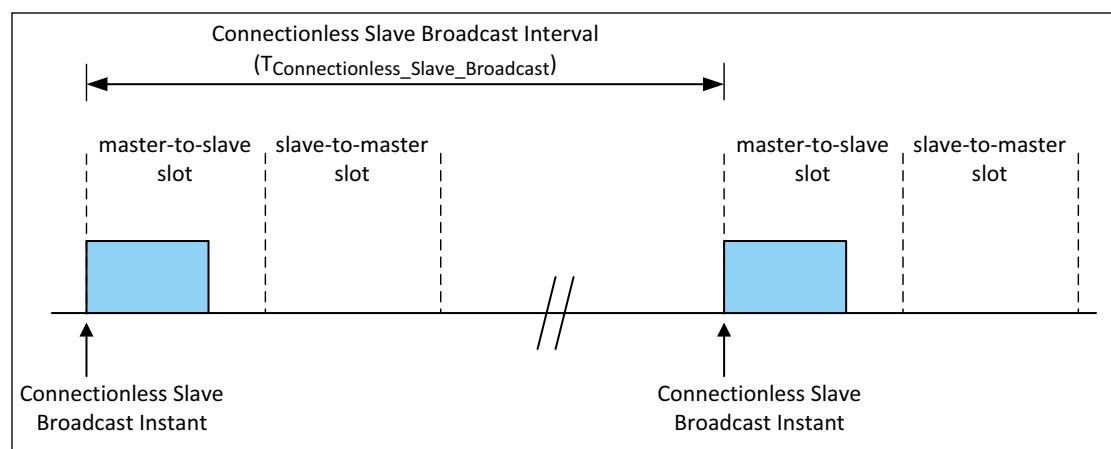


Figure 8.21: Connectionless Slave Broadcast Timing.

Connectionless Slave Broadcast Instants are separated by the Connectionless Slave Broadcast Interval ( $T_{\text{Connectionless\_Slave\_Broadcast}}$ ). The Connectionless Slave Broadcast interval can be any even value from 0x0002 to 0xFFFF slots and is negotiated between the BR/EDR Controller and the Host during CSB setup. At the start of each Connectionless Slave Broadcast Instant, the Connectionless Slave Broadcast packet is transmitted using the adapted piconet physical channel. Connectionless Slave Broadcast packets shall not be encrypted.

The master shall stop transmitting Connectionless Slave Broadcast packets after  $CSB\_supervisionTO$  slots have passed without transmission of a Connectionless Slave Broadcast packet.  $CSB\_supervisionTO$  may be any even value from 0x0002 to 0xFFFF slots. Connectionless Slave Broadcast packets can fail to transmit for a certain continuous period of time due to scheduling conflicts from higher priority traffic.

### 8.10.2 Connectionless Slave Broadcast Receive Operation [New Section]

In connectionless slave broadcast mode, the slave is synchronized to a Connectionless Slave Broadcast transmitter and is receiving profile broadcast data on the LT\_ADDR reserved by the master for the CSB logical transport.

When requesting synchronization establishment, the Host provides the Controller with the master's clock offset and the next Connectionless Slave Broadcast Instant received from the synchronization train. Because of delays in the Host, it is possible that the Connectionless Slave Broadcast Instant occurs in the past. The BR/EDR Controller shall support Connectionless Slave Broadcast instants at least 1 second in the past. The slave should determine whether the Connectionless Slave Broadcast Instant is in the past or the future by using the master's clock offset and its own clock to estimate the current master clock and start listening from the next broadcast instant.

The Skip parameter is set by the Host and specifies the number of consecutive Connectionless Slave Broadcast instants which the receiver may skip after successfully receiving a Connectionless Slave Broadcast packet. Skip is configured by the slave's Host. If the Connectionless Slave Broadcast packet is received successfully, the payload data shall be forwarded to the Host. If no Connectionless Slave Broadcast packet is received by a slave during a Connectionless Slave Broadcast Instant, the slave shall ignore Skip and instead listen at every scheduled Connectionless Slave Broadcast Instant until it is able to successfully receive a Connectionless Slave Broadcast packet or until CSB\_supervisionTO number of slots have passed.

The BR/EDR Controller shall stop listening for Connectionless Slave Broadcast packets after CSB\_supervisionTO slots have passed without receiving a Connectionless Slave Broadcast packet.

The BR/EDR Controller may transfer Connectionless Slave Broadcast data to the Host through HCI events. Because HCI events are limited to 255 bytes a received packet may not fit into a single HCI event. The BR/EDR Controller shall fragment and transfer such packets using multiple HCI events.

### 8.10.3 AFH in Connectionless Slave Broadcast [New Section]

Connectionless Slave Broadcast packets shall be transmitted on the adapted piconet channel and the synchronization train shall always contain the current AFH channel map for the PBD logical link.

## 8.11 SYNCHRONIZATION ESTABLISHMENT SUBSTATES [New Section]

### 8.11.1 Synchronization Scan Substate [New Section]

The **synchronization scan** substate is used by the Connectionless Slave Broadcast Receiver to receive synchronization train packets from the Connectionless Slave Broadcast Transmitter. The **synchronization scan** substate can be entered from the **STANDBY** state. In the **synchronization scan** substate, a device uses the Synchronization Scan procedure (see [section 2.7.3](#)). During each synchronization scan window, the device shall listen on a single frequency with its correlator matched to the Connectionless Slave Broadcast Transmitter's channel access code (CAC).

If the correlator exceeds the trigger threshold during the Synchronization Scan procedure, the scanning device shall receive the synchronization train packet, whose contents are defined in [table 8.4](#). Upon reception of the synchronization train packet the device shall exit the **synchronization scan** substate and return to the **STANDBY** state. If the packet is not received the device should stay in the **synchronization scan substate**. If the *synchronization\_scanTO* expires before reception of a Synchronization Train packet, the device shall return to the **STANDBY** state.

The synchronization scan may be interrupted by higher priority traffic. In particular, the reserved synchronous slots should have higher priority than the synchronization scan.

### 8.11.2 Synchronization Train Substate [New Section]

The **synchronization train** substate is used by the Connectionless Slave Broadcast Transmitter (which also shall be the master of the piconet) to transmit synchronization train packets. The **synchronization train** substate can be entered from the **STANDBY** state. In the **synchronization train** substate, a device uses the Synchronization Train procedure (see [section 2.7.2](#)). During the Synchronization Train procedure, the Connectionless Slave Broadcast Transmitter repeatedly transmits synchronization train packets on the channels specified in [section 2.6.4.8](#). While in the **synchronization train** substate, the master shall continue to transmit Connectionless Slave Broadcast packets in addition to synchronization train packets.

Reserved SCO and eSCO slots should have higher priority than the synchronization train. Once the master enters the **synchronization train** substate, it shall remain in the **synchronization train** substate until *synchronization\_trainTO* expires or the Host directs otherwise. The master shall transition to the **STANDBY** state when exiting the **synchronization train** substate.

*Connectionless Slave Broadcast*

The synchronization train packet is a basic rate ACL packet with type DM3 and LT\_ADDR of zero. FLOW, ARQN and SEQN shall all be set to zero upon transmission and ignored upon receipt. The HEC LFSR shall be initialized with the master UAP. In the payload header, the LLID shall contain the value 10b (start of an L2CAP message or no fragmentation). FLOW in the payload header shall be set to zero upon transmission and ignored upon reception. The length of the payload body (LENGTH) shall be 28 bytes. The CRC LFSR shall be initialized with the master UAP. Data whitening is not used. Encryption is not used.

The format of the payload portion of the DM3 packet used in the synchronization train is shown in [Table 8.8](#).

Bytes	Field	Length	Description
0-3	CLK	4 Bytes	Current piconet clock (CLK)
4-7	Future Connectionless Slave Broadcast Instant	4 Bytes	CLK[27:1] at the start of a future Connectionless Slave Broadcast instant
8-17	AFH Channel Map	10 Bytes	The current AFH_Channel_Map used for the PBD logical link. The format is the same as the Link Manager AFH_Channel_Map parameter, see <a href="#">Vol 2, Part C, Section 5.2, Table 5.2</a> .
18-23	Master BD_ADDR	6 Bytes	
24-25	Connectionless Slave Broadcast Interval	2 Bytes	Time duration in slots from the start of a Connectionless Slave Broadcast packet to the start of the next Connectionless Slave Broadcast packet. Must be even.
26	Connectionless Slave Broadcast LT_ADDR	1 Byte	LT_ADDR of the CSB logical transport
27	Service Data	1 Byte	Defined by the service using the Connectionless Slave Broadcast feature.

*Table 8.8: Synchronization Train Packet Payload Body.*

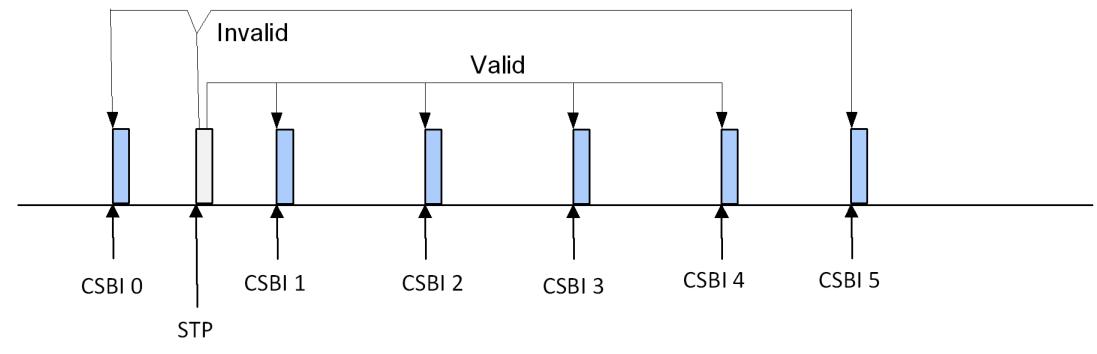
The Host provides the Service Data for the synchronization train packet payload body. The BR/EDR Controller provides the data for all other fields.

All devices in the **synchronization scan** substate shall accept payloads of any length from 28 to 121 bytes inclusive and shall ignore bytes 28 (counting from 0) and beyond.

The Future Connectionless Slave Broadcast Instant in the synchronization train packet payload shall be set to the CLK corresponding to one of the next 4 broadcast instants. [Figure 8.22](#) shows an example of valid and invalid values for this field.

*Connectionless Slave Broadcast*

STP = Synchronization Train Packet  
CSBI = Connectionless Slave Broadcast Instant



*Figure 8.22: Examples of Synchronization Train Pointing to Connectionless Slave Broadcast Instant.*

The master shall not transmit synchronization train packets in a Connectionless Slave Broadcast Instant.

## APPENDIX B: TIMERS [Updated]

---

[Add the following timers to the end of this section]

### **CSB\_supervisionTO**

The *CSB\_supervisionTO* is used by the slave to monitor connection loss and by the master to monitor scheduling conflicts and has a range of 0x0002 to 0xFFFF with only even values valid. At the baseband level a default value equivalent to 5.12 seconds should be used. The Host can change the value of *CSB\_supervisionTO*.

### **synchronization\_trainTO**

The *synchronization\_trainTO* is used by the master to determine how long to continue broadcasting Synchronization Train packets and has a range of 0x00000002 to 0x07FFFFFF with only even values valid. At the baseband level a default value equivalent to 120 seconds should be used. The Host can change the value of *synchronization\_trainTO*.

### **synchronization\_scanTO**

The *synchronization\_scanTO* is used by the slave to determine when to stop scanning for synchronization train packets. It has a range of 0x0022 to 0xFFFF with only even values valid. At the baseband level a default value equivalent to 5.12 seconds should be used. The Host can change the value of *synchronization\_scanTO*.

## APPENDIX C: RECOMMENDATIONS FOR AFH OPERATION IN PARK, HOLD, SNIFF, AND CONNECTIONLESS SLAVE BROADCAST [Updated]

[Add changes to the end of this section as follows]

### AFH Operation in Connectionless Slave Broadcast [New Section]

After the master's BR/EDR Controller notifies the master's Host that the AFH channel map has changed for the PBD logical link, the master's Host may restart the synchronization train to allow slaves to obtain the updated AFH channel map. This is shown in the following MSC.

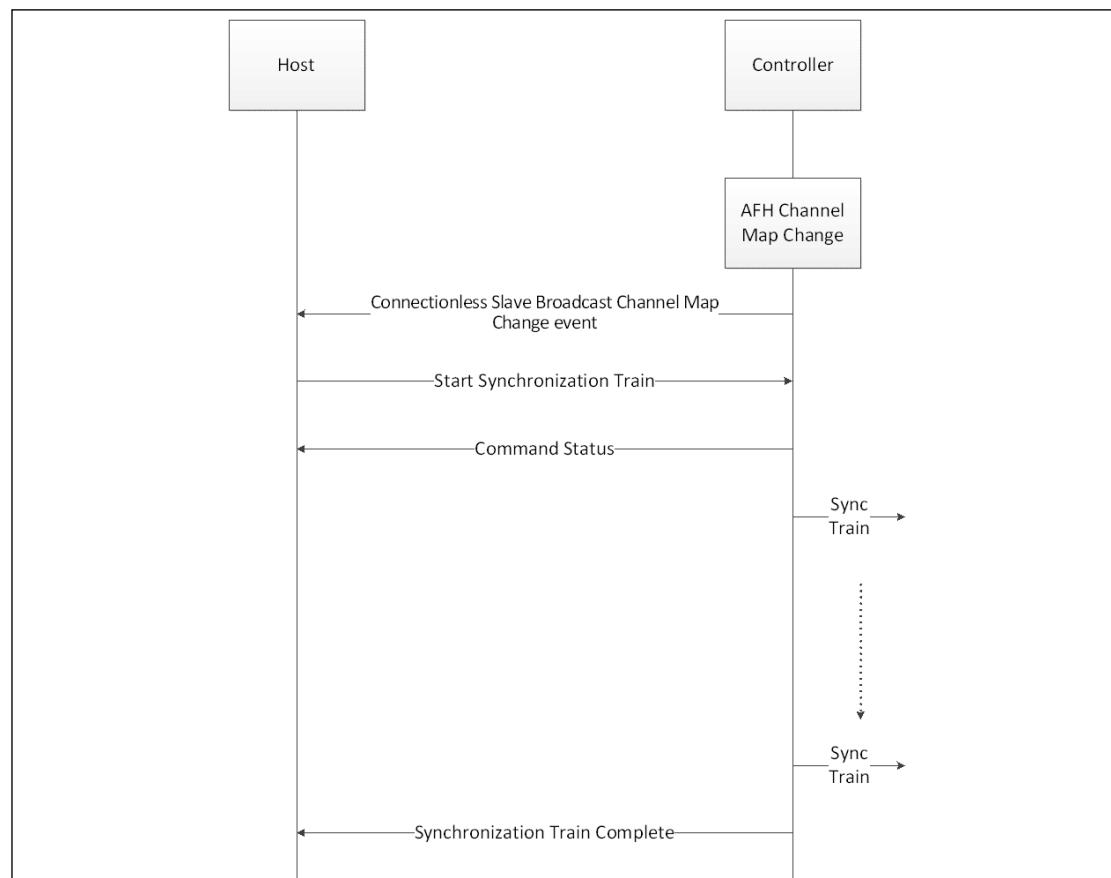


Figure 11.5: AFH Map Change – Connectionless Slave Broadcast transmitter.

Modification of the AFH channel map will affect slaves using a different AFH channel map. Depending on the overlap between a slave's current AFH channel map and the current AFH channel map of the master, the slave may miss some Connectionless Slave Broadcast instants and, in the case of disjoint or nearly disjoint AFH channel maps, the slave may lose synchronization. Slaves should monitor the Connectionless Slave Broadcast reception rate and obtain the current AFH channel map of the master (via the synchronization train) if degradation exceeds desired thresholds.

## 4 VOL 2, PART C (LINK MANAGER PROTOCOL) CHANGES:

### 3 DEVICE FEATURES

#### 3.2 FEATURE DEFINITIONS [Updated]

[Add the following rows to the end of Table 3.1: Feature Definitions]

Feature	Definition
Simultaneous LE and BR/EDR to Same Device Capable (Host)	<p>This feature indicates that the Host supports simultaneous LE and BR/EDR links to the same remote device.</p> <p>This bit shall only be set if the Simultaneous LE and BR/EDR to Same Device Capable (Controller) bit is set on the local Controller.</p> <p>The local Host sets this feature bit to indicate to a remote device that the local device is capable of supporting simultaneous LE and BR/EDR connections to the peer device.</p> <p>A remote Host uses this feature to determine whether simultaneous LE and BR/EDR connections are possible to the peer device.</p>
<u>Connectionless Slave Broadcast – Master Operation</u>	<u>This feature indicates whether the device supports Connectionless Slave Broadcast as master.</u>
<u>Connectionless Slave Broadcast – Slave Operation</u>	<u>This feature indicates whether the device supports Connectionless Slave Broadcast as slave.</u>
<u>Synchronization Train</u>	<u>This feature indicates whether the device supports Synchronization Train as master.</u>
<u>Synchronization Scan</u>	<u>This feature indicates whether the device supports Synchronization Scan as slave.</u>
<u>Inquiry Response Notification Event</u>	<u>This feature bit indicates whether the device supports sending the HCI Inquiry Response Notification Event to the Host.</u>

Table 3.1: Feature Definitions.

#### 3.3 FEATURE MASK DEFINITION [Updated]

[Add new table “Table 3.4” to the end of the section]

No.	Supported Feature	Byte	Bit
128	Connectionless Slave Broadcast – Master Operation	0	0
129	Connectionless Slave Broadcast – Slave Operation	0	1

Table 3.4: Extended feature mask definition (page 2).

*Connectionless Slave Broadcast*

No.	Supported Feature	Byte	Bit
130	Synchronization Train	0	2
131	Synchronization Scan	0	3
132	Inquiry Response Notification Event	0	4

Table 3.4: Extended feature mask definition (page 2).

## 5 VOL 2, PART E (HOST CONTROLLER INTERFACE) CHANGES:

### 3 OVERVIEW OF COMMANDS AND EVENTS [Updated]

[Add an entry at the end of Table 3.1 as follows]

<b>Testing</b>	The testing commands and events allow a device to be placed into test mode.
<b>Connectionless Slave Broadcast</b>	<u>The Connectionless Slave Broadcast commands and events allow setup and teardown of a Connectionless Slave Broadcast physical link.</u>

Table 3.1: Overview of commands and events.

### 3.6 DEVICE DISCOVERY [Updated]

[Add an entry at the end of Table 3.7 as follows]

Name	Vers.	Summary description	Supported Controllers
LE Set Scan Parameters Command	4.0	The LE Set Scan Parameters Command will set the parameters used for scanning.	LE
<u>Inquiry Response Notification Event</u>	<u>CSA4</u>	<u>The Inquiry Response Notification event indicates to the Host that the BR/EDR Controller responded to an inquiry message.</u>	<u>BR/EDR</u>

Table 3.7: Device discovery.

### 3.7 CONNECTION SETUP [Updated]

[Add rows at the end of Table 3.8 as follows]

Name	Vers.	Summary description	Supported Controllers
LE Create Connection Command	4.0	The LE Create Connection Command is used to create a new connection.	LE
<u>Slave Page Response Timeout Event</u>	<u>CSA4</u>	<u>The Slave Page Response Timeout event indicates to the Host that the <code>pagerespTO</code> has been exceeded on the BR/EDR Controller after the Controller responded to an ID packet.</u>	<u>BR/EDR</u>
<u>Truncated Page Command</u>	<u>CSA4</u>	<u>The Truncated Page command will cause the BR/EDR Controller to page the BR/EDR Controller with the <code>BD_ADDR</code> specified by the command parameters and abort the page sequence after receiving the ID response packet.</u>	<u>BR/EDR</u>
<u>Truncated Page Cancel Command</u>	<u>CSA4</u>	<u>The Truncated Page Cancel Command is used to cancel an ongoing Truncated Page.</u>	<u>BR/EDR</u>
<u>Truncated Page Complete Event</u>	<u>CSA4</u>	<u>The Truncated Page Complete event indicates to the Host that a Truncated Page has completed.</u>	<u>BR/EDR</u>

Table 3.8: Connection setup.

### 3.15 LINK INFORMATION [Updated]

[Insert the following rows to *Table 3.16*, after Read Clock Command]

Name	Vers.	Summary description	Supported Controllers
Read LMP Handle Command	1.2	The Read LMP Handle command will read the current LMP Handle associated with the Connection_Handle.	BR/EDR
Read Transmit Power Level Command	1.1	The Read Transmit Power Level command will read the values for the Transmit Power Level parameter for the specified Connection_Handle.	BR/EDR, LE
Read Link Quality Command	1.1	The Read Link Quality command will read the value for the Link Quality for the specified Connection_Handle.	BR/EDR, AMP
Read RSSI Command	1.1	The Read RSSI command will read the value for the Received Signal Strength Indication (RSSI) for a Connection_Handle to another Controller.	All
Read Clock Offset Command	1.1	The Read Clock Offset command allows the Host to read the clock offset of remote BR/EDR Controllers.	BR/EDR
Read Clock Offset Complete Event	1.1	The Read Clock Offset Complete event is used to indicate the completion of the process of the LM obtaining the Clock offset information.	BR/EDR
Read Clock Command	1.2	The Read Clock command will read an estimate of a piconet or the local Bluetooth Clock.	BR/EDR
<u>Set Triggered Clock Capture Command</u>	CSA4	<u>The Set Triggered Clock Capture command is used to configure the Controller to return events containing an estimate of a piconet or the local Bluetooth clock.</u>	<u>BR/EDR</u>
<u>Triggered Clock Capture event</u>	CSA4	<u>The Triggered Clock Capture event reports the Bluetooth clock when an external trigger occurred.</u>	<u>BR/EDR</u>
Read AFH Channel Map Command	1.2	The Read AFH Channel Map command will read the current state of the channel map for a connection.	BR/EDR
Read Local AMP Info Command	3.0 + HS	The Read Local AMP Info command returns information about this AMP Controller.	AMP

Table 3.16: Link information.

*Connectionless Slave Broadcast*

Name	Vers.	Summary description	Supported Controllers
AMP Status Change Event	3.0 + HS	The AMP Status Change event can occur at any time, after initial Read Local AMP_Info request, the Controller detects a change has occurred regarding status.	AMP
Read Enhanced Transmit Power Level Command	3.0 + HS	The Read Enhanced Transmit Power Level command will read the values for the GFSK, DQPSK and 8DPSK Transmit Power Level parameters for the specified Connection_Handle.	BR/EDR
LE Read Advertising Channel TX Power Command	4.0	The LE Read Advertising Channel TX Power Command will read the transmit power level that will be used for advertising.	LE
LE Read Channel Map Command	4.0	The LE Read Channel Map Command will read the current state of the channel map for a connection.	LE

Table 3.16: Link information.

### 3.18 ALPHABETICAL LIST OF COMMANDS AND EVENTS

[Updated]

[Add the following items to Table 3.19]

Commands/Events	Group
<u>Connectionless Slave Broadcast Channel Map Change Event</u>	<u>Connectionless Slave Broadcast</u>
<u>Connectionless Slave Broadcast Receive Event</u>	<u>Connectionless Slave Broadcast</u>
<u>Connectionless Slave Broadcast Timeout Event</u>	<u>Connectionless Slave Broadcast</u>
<u>Delete Reserved LT_ADDR Command</u>	<u>Connectionless Slave Broadcast</u>
<u>Inquiry Response Notification Event</u>	<u>Device Discovery</u>
<u>Read Synchronization Train Parameters Command</u>	<u>Connectionless Slave Broadcast</u>
<u>Receive Synchronization Train Command</u>	<u>Connectionless Slave Broadcast</u>
<u>Set Connectionless Slave Broadcast Command</u>	<u>Connectionless Slave Broadcast</u>
<u>Set Connectionless Slave Broadcast Data Command</u>	<u>Connectionless Slave Broadcast</u>
<u>Set Connectionless Slave Broadcast Receive Command</u>	<u>Connectionless Slave Broadcast</u>
<u>Set Reserved LT_ADDR Command</u>	<u>Connectionless Slave Broadcast</u>
<u>Set Triggered Clock Capture Command</u>	<u>Link Information</u>
<u>Slave Page Response Timeout Event</u>	<u>Connection Setup</u>
<u>Start Synchronization Train Command</u>	<u>Connectionless Slave Broadcast</u>
<u>Synchronization Train Complete Event</u>	<u>Connectionless Slave Broadcast</u>
<u>Synchronization Train Received Event</u>	<u>Connectionless Slave Broadcast</u>
<u>Triggered Clock Capture Event</u>	<u>Link Information</u>
<u>Truncated Page Cancel Command</u>	<u>Connection Setup</u>
<u>Truncated Page Command</u>	<u>Connection Setup</u>
<u>Truncated Page Complete Event</u>	<u>Connection Setup</u>
<u>Write Synchronization Train Parameters Command</u>	<u>Connectionless Slave Broadcast</u>

Table 3.19: Alphabetical list of commands and events.

<Note from editor: Items in the above table will be sorted automatically when added to the table in the main specification document>

### 3.20 CONNECTIONLESS SLAVE BROADCAST [New Section]

The Connectionless Slave Broadcast Group of commands and events allows setup and teardown of a Connectionless Slave Broadcast physical link.

Name	Vers.	Summary description	Supported Controllers
Set Reserved LT_ADDR Command	CSA4	The Set_Reserved_LT_ADDR command requests that the BR/EDR Controller reserve a specific LT_ADDR for the purposes of Connectionless Slave Broadcast.	BR/EDR
Delete Reserved LT_ADDR Command	CSA4	The Delete Reserved LT_ADDR command requests that the BR/EDR Controller cancel the reservation of a specific LT_ADDR reserved for the purposes of Connectionless Slave Broadcast.	BR/EDR
Set Connectionless Slave Broadcast Data Command	CSA4	The Set Connectionless Slave Broadcast Data command is used by the Host to set Connectionless Slave Broadcast data in the BR/EDR Controller.	BR/EDR
Set Connectionless Slave Broadcast Command	CSA4	The Set Connectionless Slave Broadcast command controls Connectionless Slave Broadcast functionality (for transmission) in the BR/EDR Controller including enabling and disabling the broadcast.	BR/EDR
Set Connectionless Slave Broadcast Receive Command	CSA4	The Set Connectionless Slave Broadcast Receive command enables and disables Connectionless Slave Broadcast reception in the BR/EDR Controller.	BR/EDR
Write Synchronization Train Parameters Command	CSA4	The Write Synchronization Train Parameters command configures the Synchronization Train functionality in the BR/EDR Controller.	BR/EDR
Read Synchronization Train Parameters Command	CSA4	The Read Synchronization Train Parameters command returns the currently configured values for the Synchronization Train functionality in the BR/EDR Controller.	BR/EDR
Start Synchronization Train Command	CSA4	The Start Synchronization Train command enables the Synchronization Train on the BR/EDR Controller using the currently configured Synchronization Train parameters.	BR/EDR

Table 3.21: Connectionless Slave Broadcast.

*Connectionless Slave Broadcast*

Name	Vers.	Summary description	Supported Controllers
Receive Synchronization Train Command	CSA4	The Receive_Synchronization_Train command requests synchronization with the specified Connectionless Slave Broadcast transmitter.	BR/EDR
Synchronization Train Complete Event	CSA4	The Synchronization Train Complete event indicates that the Synchronization Train has completed.	BR/EDR
Synchronization Train Received Event	CSA4	The Synchronization Train Received event provides the status of Synchronization Train packets received from the device with the given BD_ADDR.	BR/EDR
Connectionless Slave Broadcast Receive Event	CSA4	The Connectionless Slave Broadcast Receive event provides the Host with the data received from a Connectionless Slave Broadcast packet.	BR/EDR
Connectionless Slave Broadcast Timeout Event	CSA4	On the Connectionless Slave Broadcast Receiver, the Connectionless Slave Broadcast Timeout event indicates to the Host that the BR/EDR Controller has lost synchronization with the Connectionless Slave Broadcast Transmitter.  On the Connectionless Slave Broadcast Transmitter, the Connectionless Slave Broadcast Timeout event indicates to the Host that the BR/EDR Controller has been unable to transmit a Connectionless Slave Broadcast packet for the timeout interval specified in the Set Connectionless Slave Broadcast command.	BR/EDR
Connectionless Slave Broadcast Channel Map Change Event	CSA4	The Connectionless Slave Broadcast Channel Map Change event indicates to the Host that the BR/EDR Controller has moved to a new AFH channel map for the PBD logical link.	BR/EDR

Table 3.21: Connectionless Slave Broadcast.

## 5 HCI DATA FORMATS

---

### 5.3.1.1 Broadcast Connection Handles [Updated]

[Add a new paragraph to the end of this section as follows]

For Connectionless Slave Broadcast, no Connection Handle is assigned.

## 6 HCI CONFIGURATION PARAMETERS

### 6.27 SUPPORTED COMMANDS [Updated]

[Add the following rows to the table in Section 6.27 in order]

Octet	Bit	Command Supported
29	0	Reserved
	1	Reserved
	2	Reserved
	3	Enhanced Setup Synchronous Connection
	4	Enhanced Accept Synchronous Connection
	5	Read Local Supported Codecs
	6	Set MWS Channel Parameters Command
	7	Set External Frame Configuration Command
30	0	Set MWS Signaling Command
	1	Set Transport Layer Command
	2	Set MWS Scan Frequency Table Command
	3	Get Transport Layer Configuration Command
	4	Set MWS PATTERN Configuration Command
	5	<u>Set Triggered Clock Capture</u>
	6	<u>Truncated Page</u>
	7	<u>Truncated Page Cancel</u>
31	0	<u>Set Connectionless Slave Broadcast</u>
	1	<u>Set Connectionless Slave Broadcast Receive</u>
	2	<u>Start Synchronization Train</u>
	3	<u>Receive Synchronization Train</u>
	4	<u>Set Reserved LT_ADDR</u>
	5	<u>Delete Reserved LT_ADDR</u>
	6	<u>Set Connectionless Slave Broadcast Data</u>
	7	<u>Read Synchronization Train Parameters</u>
32	0	<u>Write Synchronization Train Parameters</u>

## 6.36 SYNCHRONIZATION TRAIN INTERVAL [New Section]

The Sync\_Train\_Interval configuration parameter defines the time between Synchronization Train transmit events on a single transmit RF channel.

Value	Parameter Description
N = 0xFFFF	Size: 2 Octets Range: 0x0002 to 0xFFFF; only even values are valid Default: 0x0080 Mandatory Range: 0x0012 to 0x1000 $Time = N * 0.625 \text{ msec}$ Time Range: 1.25 msec to 40.9 sec Time Default: 80 msec

## 6.37 SYNCHRONIZATION TRAIN TIMEOUT [New Section]

The synchronization\_trainTO configuration parameter is used by the controller to terminate the Synchronization Train after it has been started via the HC\_Start\_Synchronization\_Train command.

Value	Parameter Description
N = 0xFFFFFFFF	Size: 4 Octets Range: 0x00000002 to 0x07FFFFFF; only even values are valid Default: 0x0002EE00 Mandatory Range: 0x00000002 to 0x07FFFFFF $Time = N * 0.625 \text{ msec}$ Time Range: 1.25 msec to 23.3 hours Time Default: 120 sec

## 6.38 SERVICE DATA [New Section]

The Service\_Data configuration parameter defines the value of the service data field in the Synchronization Train.

Value	Parameter Description
N = 0xXX	Size: 1 Octet Range: 0x00 to 0xFF Default: 0x00

## 7 HCI COMMANDS AND EVENTS

### 7.1 LINK CONTROL COMMANDS

#### 7.1.47 Truncated Page Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Truncated_Page	0x003F	BD_ADDR, Page_Scan_Repetition_Mode, Clock_Offset	

#### Description:

The Truncated\_Page command is used to page the BR/EDR Controller with the specified BD\_ADDR and then abort the paging sequence after an ID response has been received. See [Vol 2, Part B, section 8.3.3](#) for additional information.

The Page\_Scan\_Repetition\_Mode parameter specifies the page scan repetition mode supported by the remote BR/EDR Controller with the BD\_ADDR. This is the information that was acquired during the inquiry process.

The Clock\_Offset parameter is the difference between the local BR/EDR Controller's own clock and the clock of the remote BR/EDR Controller with BD\_ADDR. Only bits 2 through 16 of the difference are used, and they are mapped to this parameter as bits 0 through 14 respectively. A Clock\_Offset\_Valid\_Flag, located in bit 15 of the Clock\_Offset parameter, indicates if the Clock Offset is valid or not.

#### Command Parameters:

*BD\_ADDR:* Size: 6 Octets

Value	Parameter Description
0xFFFFFFFFXXXXXX	BD_ADDR of the Device to page

*Page\_Scan\_Repetition\_Mode:* Size: 1 Octet

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

*Connectionless Slave Broadcast****Clock\_Offset:******Size: 2 Octets***

<b>Value</b>	<b>Parameter Description</b>
Bit 14-0	Bit 16-2 of CLKslave-CLKmaster
Bit 15	0 = Clock_Offset is invalid 1 = Clock_Offset is valid

**Return Parameters:**

None.

**Event(s) generated (unless masked away):**

When the BR/EDR Controller receives the Truncated\_Page command the BR/EDR Controller shall send the Command Status event to the Host. In addition, when the Truncated Page procedure has completed, the BR/EDR Controller shall send a Truncated Page Complete event to the Host.

Note: No Command Complete event will be sent by the BR/EDR Controller to indicate that this command has been completed. Instead, the Truncated Page Complete event will indicate that this command has been completed.

### 7.1.48 Truncated Page Cancel Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Truncated_Page_Cancel	0x0040	BD_ADDR	Status, BD_ADDR

#### Description:

The Truncated\_Page\_Cancel command is used to request cancellation of an ongoing Truncated\_Page process previously started by a Truncated\_Page command.

#### Command Parameters:

*BD\_ADDR*: Size: 6 Octets

Value	Parameter Description
0xXXXXXXXXXXXXX	BD_ADDR of the device to which the Truncated Page command was previously issued and that is the subject of the cancellation request.

#### Return Parameters:

*Status*: Size: 1 Octet

Value	Parameter Description
0x00	Truncated_Page_Cancel command succeeded.
0x01-0xFF	Truncated_Page_Cancel command failed. <a href="#">See Part D, Error Codes</a> , for error codes and descriptions

*BD\_ADDR*: Size: 6 Octets

Value	Parameter Description
0xXXXXXXXXXXXXX	BD_ADDR of the device to which the Truncated Page command was previously issued before and that is the subject of the cancellation request.

**Event(s) generated (unless masked away):**

When the Truncated\_Page\_Cancel command has completed, a Command Complete event shall be generated.

If the truncated page procedure has already completed, but the BR/EDR Controller has not yet sent the Truncated Page Complete event, then the local device shall return a Command Complete event with status “Success”.

If the Truncated Page Cancel command is sent to the BR/EDR Controller without an outstanding Truncated Page Command to the same device, the BR/EDR Controller shall return a Command Complete event with the error code Unknown Connection Identifier (0x02). Note that from the BR/EDR Controller perspective this is identical to the situation where the Truncated Page Command has already completed and the Truncated Page Complete event already sent.

### 7.1.49 Set Connectionless Slave Broadcast Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connectionless_Slave_Broadcast	0x0041	Enable, LT_ADDR, LPO_Allowed, Packet_Type, Interval_Min, Interval_Max, CSB_supervisionTO	Status, LT_ADDR, Interval

#### Description:

The Set\_Connectionless\_Slave\_Broadcast command controls the Connectionless Slave Broadcast functionality in the BR/EDR Controller. Connectionless slave broadcast mode may be enabled or disabled by the Enable parameter. If this command is issued with Enable = 0, and the **synchronization train** substate is active, then the Controller shall also exit the **synchronization** train substate.

The LT\_ADDR indicated in the Set\_Connectionless\_Slave\_Broadcast shall be pre-allocated using the HCI\_Set\_Reserved\_LT\_ADDR command. If the LT\_ADDR has not been reserved, the *Unknown Connection Identifier* (0x02) error code shall be returned. If the controller is unable to reserve sufficient bandwidth for the requested activity, the *Connection Rejected Due to Limited Resources* (0x0D) error code shall be returned.

The LPO\_Allowed parameter informs the BR/EDR Controller whether it is allowed to sleep.

The Packet\_Type parameter specifies which packet types are allowed. The Host shall either enable BR packet types only, or shall enable EDR and DM1 packet types only.

The Interval\_Min and Interval\_Max parameters specify the range from which the BR/EDR Controller must select the Connectionless Slave Broadcast Interval. The selected Interval is returned.

#### Command Parameters:

*Enable*:

Size: 1 Octet

Value	Parameter Description
0x00	Disabled
0x01	Enabled
0x02-0xFF	Reserved for future use

*Connectionless Slave Broadcast**LT\_ADDR:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x01-0x07	LT_ADDR used for Connectionless Slave Broadcast
0x00, 0x08-0xFF	Reserved for future use

*LPO\_Allowed:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	BR/EDR Controller shall not sleep (that is, clock accuracy shall be equal to or better than $\pm 20$ ppm)
0x01	BR/EDR Controller may sleep (that is, clock accuracy shall be equal to or better than $\pm 250$ ppm)
0x02-0xFF	Reserved for future use

*Packet\_Type:**Size: 2 Octets*

<b>Value</b>	<b>Parameter Description</b>
0x0001	Reserved for future use
0x0002	2-DH1 may not be used
0x0004	3-DH1 may not be used
0x0008	DM1 may be used
0x0010	DH1 may be used
0x0020	Reserved for future use
0x0040	Reserved for future use
0x0080	Reserved for future use
0x0100	2-DH3 may not be used
0x0200	3-DH3 may not be used
0x0400	DM3 may be used
0x0800	DH3 may be used
0x1000	2-DH5 may not be used
0x2000	3-DH5 may not be used
0x4000	DM5 may be used
0x8000	DH5 may be used

*Connectionless Slave Broadcast**Interval\_Min:*

Size: 2 Octets

Value	Parameter Description
N = 0XXXX	Minimum interval between Connectionless Slave Broadcast packets in slots. Range: 0x0002-0xFFFF; only even values are valid

*Interval\_Max:*

Size: 2 Octets

Value	Parameter Description
N = 0XXXX	Maximum interval between Connectionless Slave Broadcast packets in slots. Range: 0x0002-0xFFFF; only even values are valid

*CSB\_supervisionTO:*

Size: 2 Octets

Value	Parameter Description
N = 0XXXX	Duration in slots after which the BR/EDR Controller reports a Connectionless Slave Broadcast Timeout event if it is unable to transmit a Connectionless Slave Broadcast packet. Range: 0x0002-0xFFFF; only even values are valid

**Return Parameters:***Status:*

Size: 1 Octet

Value	Parameter Description
0x00	Set_Connectionless_Slave_Broadcast command succeeded.
0x01-0xFF	Set_Connectionless_Slave_Broadcast command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions.

*LT\_ADDR:*

Size: 1 Octet

Value	Parameter Description
0x01-0x07	LT_ADDR used for Connectionless Slave Broadcast
0x00, 0x08-0xFF	Reserved for future use

*Interval:*

Size: 2 Octets

Value	Parameter Description
N = 0XXXX	Actual interval between Connectionless Slave Broadcast packets in slots. Range: 0x0002-0xFFFF; only even values are valid

**Event(s) generated (unless masked away):**

When the Set\_Connectionless\_Slave\_Broadcast command has completed, a Command Complete event shall be generated.

If the BR/EDR Controller is unable to transmit a Connectionless Slave Broadcast packet for  $CSB\_supervisionTO$  slots, it shall generate a Connectionless Slave Broadcast Timeout event.

### 7.1.50 Set Connectionless Slave Broadcast Receive Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connectionless_Slave_Broadcast_Receive	0x0042	Enable, BD_ADDR, LT_ADDR, Interval, Clock_Offset, Next_Connectionless_Slave_Broadcast_Clock, CSB_supervisionTO, Remote_Timing_Accuracy, Skip, Packet_Type, AFH_Channel_Map	Status, BD_ADDR, LT_ADDR

#### Description:

The Set\_Connectionless\_Slave\_Broadcast\_Receive command controls the reception of Connectionless Slave Broadcast packets in the BR/EDR Controller of a Connectionless Slave Broadcast slave. If the Enable parameter is set to Disabled, the BR/EDR Controller does not attempt to receive Connectionless Slave Broadcast packets. If the Enable parameter is set to Enabled, the BR/EDR Controller starts attempting to receive Connectionless Slave Broadcast packets on the specified LT\_ADDR.

The Interval parameter specifies the interval of the Connectionless Slave Broadcast to be used by the BR/EDR Controller.

The Skip parameter specifies the number of consecutive Connectionless Slave Broadcast instants which the receiver may skip after successfully receiving a Connectionless Slave Broadcast packet.

The Packet\_Type parameter specifies which packet types are allowed. The Host shall either enable BR packet types only or shall enable EDR and DM1 packet types only.

The AFH\_Channel\_Map parameter is the AFH channel map used by the master for the PBD logical link, and is obtained by the slave's Host from the Synchronization Train Received event.

*Connectionless Slave Broadcast***Command Parameters:***Enable:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	Disabled
0x01	Enabled
0x02-0xFF	Reserved for future use

*BD\_ADDR:**Size: 6 Octets*

<b>Value</b>	<b>Parameter Description</b>
0xXXXXXXXXXXXXX	BD_ADDR of the Connectionless Slave Broadcast transmitter

*LT\_ADDR:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x01-0x07	LT_ADDR to use for receiving Connectionless Slave Broadcast messages
0x00, 0x08-0xFF	Reserved for future use

*Interval:**Size: 2 Octets*

<b>Value</b>	<b>Parameter Description</b>
N = 0xFFFF	Interval between Connectionless Slave Broadcast packets instants in slots. Range: 0x0002-0xFFFF; only even values are valid

*Clock\_Offset:**Size: 4 Octets (28 bits meaningful)*

<b>Value</b>	<b>Parameter Description</b>
0xXXXXXXXX	(CLKslave – CLKmaster) modulo $2^{27}$

*Next\_Connectionless\_Slave\_Broadcast\_Clock:* *Size: 4 Octets (28 bits meaningful)*

<b>Value</b>	<b>Parameter Description</b>
0xXXXXXXXX	CLK for next Connectionless Slave Broadcast instant

*CSB\_supervisionTO:**Size: 2 Octets*

<b>Value</b>	<b>Parameter Description</b>
N = 0xFFFF	Duration in slots to continue listening for Connectionless Slave Broadcast packets after the last successfully received Connectionless Slave Broadcast packet. Range: 0x0002-0xFFFF; only even values are valid

*Connectionless Slave Broadcast**Remote\_Timing\_Accuracy:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
N = 0xXX	Timing accuracy of the master in ppm. Typical values are 20ppm and 250ppm.

*Skip:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
N = 0xXX	Number of Connectionless Slave Broadcast instants to skip after successfully receiving a Broadcast packet.

*Packet\_Type:**Size: 2 Octets*

<b>Value</b>	<b>Parameter Description</b>
0x0001	Reserved for future use
0x0002	2-DH1 may not be used
0x0004	3-DH1 may not be used
0x0008	DM1 may be used
0x0010	DH1 may be used
0x0020	Reserved for future use
0x0040	Reserved for future use
0x0080	Reserved for future use
0x0100	2-DH3 may not be used
0x0200	3-DH3 may not be used
0x0400	DM3 may be used
0x0800	DH3 may be used
0x1000	2-DH5 may not be used
0x2000	3-DH5 may not be used
0x4000	DM5 may be used
0x8000	DH5 may be used

*AFH\_Channel\_Map:**Size: 10 Octets (79 Bits Meaningful)*

<b>Value</b>	<b>Parameter Description</b>
0XXXXXXXXX XXXXXX XX	This parameter contains 80 1-bit fields. The $n^{th}$ such field (in the range 0 to 78) contains the value for channel $n$ : Channel $n$ is bad = 0 Channel $n$ is unknown = 1 The most significant bit (bit 79) is reserved and shall be set to 0

*Connectionless Slave Broadcast***Return Parameters:***Status:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	Set_Connectionless_Slave_Broadcast_Receive command succeeded.
0x01-0xFF	Set_Connectionless_Slave_Broadcast_Receive command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions.

*BD\_ADDR:**Size: 6 Octets*

<b>Value</b>	<b>Parameter Description</b>
0xXXXXXXXXXXXXX	BD_ADDR of the Connectionless Slave Broadcast transmitter

*LT\_ADDR:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x01-0x07	LT_ADDR used for receiving Connectionless Slave Broadcast messages
0x00, 0x08-0xFF	Reserved for future use

**Event(s) generated (unless masked away):**

When the Set\_Connectionless\_Slave\_Broadcast\_Receive command has been received, a Command Complete event shall be generated. Completion of the Set\_Connectionless\_Slave\_Broadcast\_Receive command does not require reception of a Connectionless Slave Broadcast packet. If the BR/EDR Controller does not receive a Connectionless Slave Broadcast packet for CSB\_supervisionTO slots, it shall generate a Connectionless Slave Broadcast Timeout event.

### 7.1.51 Start Synchronization Train Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Start_Synchronization_Train	0x0043	None	

#### Description:

The Start\_Synchronization\_Train command controls the Synchronization Train functionality in the BR/EDR Controller. Connectionless slave broadcast mode must be enabled on the BR/EDR Controller before this command may be used. If connectionless slave broadcast mode is not enabled, the Command Disallowed (0x0C) error code shall be returned. After receiving this command and returning a Command Status event, the Baseband starts attempting to send synchronization train packets containing information related to the enabled Connectionless Slave Broadcast packet timing.

Note: The AFH\_Channel\_Map used in the synchronization train packets is configured by the Set\_AFH\_Channel\_Classification command and the local channel classification in the BR/EDR Controller.

The synchronization train packets will be sent using the parameters specified by the latest Write\_Synchronization\_Train\_Parameters command. The Synchronization Train will continue until *synchronization\_trainTO* slots (as specified in the last Write\_Synchronization\_Train command) have passed or until the Host disables the Connectionless Slave Broadcast logical transport.

#### Command Parameters:

None.

#### Return Parameters:

None

#### Event(s) generated (unless masked away):

When the BR/EDR Controller receives the Start\_Synchronization\_Train command, it shall send a Command Status event to the Host.

Note: No Command Complete event shall be sent by the BR/EDR Controller to indicate that this command has been completed. Instead the Synchronization Train Complete event will indicate that this command has been completed.

### 7.1.52 Receive Synchronization Train Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Receive_Synchronization_Train	0x0044	BD_ADDR, synchronization_scanTO, Sync_Scan_Window, Sync_Scan_Interval	

#### Description:

The Receive\_Synchronization\_Train command requests synchronization with the specified Connectionless Slave Broadcast Transmitter. The Sync\_Scan\_Window parameter specifies the duration of each scan and the Sync\_Scan\_Interval parameter specifies the interval between the start of consecutive scan windows. A Synchronization\_Train\_Received event shall be sent if a synchronization train packet is received or if the BR/EDR Controller fails to receive a synchronization train packet for *synchronization\_scanTO slots*.

#### Command Parameters:

*BD\_ADDR*: Size: 6 Octets

Value	Parameter Description
0xxxxxxxxxxxxxx	BD_ADDR of the Connectionless Slave Broadcast transmitter

*synchronization\_scanTO*: Size: 2 Octets

Value	Parameter Description
N = 0xFFFF	Duration in slots to search for the synchronization train Range: 0x0022-0xFFFF; only even values are valid

*Sync\_Scan\_Window*: Size: 2 Octets

Value	Parameter Description
N = 0xFFFF	Duration in slots to listen for a synchronization train packet on a single frequency Range: 0x0022-0xFFFF; only even values are valid

*Sync\_Scan\_Interval*: Size: 2 Octets

Value	Parameter Description
N = 0xFFFF	Duration in slots between the start of consecutive scan windows Range: 0x0002-0xFFFF; only even values are valid

**Return Parameters:**

None.

**Event(s) generated (unless masked away):**

When the BR/EDR Controller receives the Receive\_Synchronization\_Train command, it shall send a Command Status event to the Host. In addition, when the BR/EDR Controller receives, or fails to receive within the duration specified by *synchronization\_scanTO*, the synchronization train from the Connectionless Slave Broadcast transmitter, it shall send a Synchronization Train Received event to the Host.

Note: No Command Complete event shall be sent by the BR/EDR Controller to indicate that this command has been completed. Instead the Synchronization Train Received event will indicate that this command has been completed.

## 7.3 CONTROLLER & BASEBAND COMMANDS

### 7.3.69 Set Event Mask Page 2 Command [Updated]

#### Command Parameters:

*Event\_Mask\_Page\_2:* Size: 8 Octets

Value	Parameter Description
0x0000000000000000	No events specified ( <b>default</b> )
0x0000000000000001	Physical Link Complete Event
0x0000000000000002	Channel Selected Event
0x0000000000000004	Disconnection Physical Link Event
0x0000000000000008	Physical Link Loss Early Warning Event
0x0000000000000010	Physical Link Recovery Event
0x0000000000000020	Logical Link Complete Event
0x0000000000000040	Disconnection Logical Link Complete Event
0x0000000000000080	Flow Spec Modify Complete Event
0x0000000000000100	Number of Completed Data Blocks Event
0x0000000000000200	AMP Start Test Event
0x0000000000000400	AMP Test End Event
0x0000000000000800	AMP Receiver Report Event
0x0000000000001000	Short Range Mode Change Complete Event
0x0000000000002000	AMP Status Change Event
<u>0x0000000000004000</u>	<u>Triggered Clock Capture Event</u>
<u>0xFFFFFFFFFFF8000</u>	<u>Reserved for future use</u>
<u>0x0000000000008000</u>	<u>Synchronization Train Complete Event</u>
<u>0x00000000000010000</u>	<u>Synchronization Train Received Event</u>
<u>0x00000000000020000</u>	<u>Connectionless Slave Broadcast Receive Event</u>
<u>0x00000000000040000</u>	<u>Connectionless Slave Broadcast Timeout Event</u>
<u>0x00000000000080000</u>	<u>Truncated Page Complete Event</u>
<u>0x00000000000100000</u>	<u>Slave Page Response Timeout Event</u>
<u>0x00000000000200000</u>	<u>Connectionless Slave Broadcast Channel Map Change Event</u>
<u>0x00000000000400000</u>	<u>Inquiry Response Notification Event</u>
<u>0xFFFFFFFFFFF800000</u>	<u>Reserved for future use</u>

### 7.3.86 Set Reserved LT\_ADDR Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Reserved_LT_ADDR	0x0074	LT_ADDR	Status, LT_ADDR

#### Description:

The Set\_Reserved\_LT\_ADDR command allows the Host to request that the BR/EDR Controller reserve a specific LT\_ADDR for Connectionless Slave Broadcast.

If the LT\_ADDR indicated in the LT\_ADDR parameter is already in use by the BR/EDR Controller, it shall return the ACL Connection Already Exists (0x0B) error code. If the LT\_ADDR indicated in the LT\_ADDR parameter is out of range, the controller shall return the Invalid HCI Command Parameters (0x12) error code. If the command succeeds, then the reserved LT\_ADDR shall be used when issuing subsequent Set Connectionless Slave Broadcast Data and Set Connectionless Slave Broadcast commands.

To ensure that the reserved LT\_ADDR is not already allocated, it is recommended that this command be issued at some point after HCI\_Reset is issued but before page scanning is enabled or paging is initiated.

#### Command Parameters:

*LT\_ADDR:* Size: 1 Octet

Value	Parameter Description
0x01-0x07	LT_ADDR to reserve for Connectionless Slave Broadcast
0x00, 0x08-0xFF	Reserved for future use

#### Return Parameters:

*Status:* Size: 1 Octet

Value	Parameter Description
0x00	Set_Reserved_LT_ADDR command succeeded.
0x01-0xFF	Set_Reserved_LT_ADDR command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions.

*Connectionless Slave Broadcast**LT\_ADDR:**Size: 1 Octet*

Value	Parameter Description
0x01-0x07	LT_ADDR reserved for Connectionless Slave Broadcast. This parameter shall have the same value as the Command Parameter LT_ADDR.
0x00, 0x08-0xFF	Reserved for future use

**Event(s) generated (unless masked away):**

When the Set\_Reserved\_LT\_ADDR command has completed, a Command Complete event shall be generated.

### 7.3.87 Delete Reserved LT\_ADDR Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Delete_Reserved_LT_ADDR	0x0075	LT_ADDR	Status, LT_ADDR

#### Description:

The Delete\_Reserved\_LT\_ADDR command requests that the BR/EDR Controller cancel the reservation for a specific LT\_ADDR reserved for the purposes of Connectionless Slave Broadcast.

If the LT\_ADDR indicated in the LT\_ADDR parameter is not reserved by the BR/EDR Controller, it shall return the Unknown Connection Identifier (0x02) error code.

If connectionless slave broadcast mode is still active, then the Controller shall return the Command Disallowed (0x0C) error code.

#### Command Parameters:

*LT\_ADDR*: Size: 1 Octet

Value	Parameter Description
0x01-0x07	LT_ADDR currently reserved for Connectionless Slave Broadcast and for which reservation is to be cancelled
0x00, 0x08-0xFF	Reserved for future use

#### Return Parameters:

*Status*: Size: 1 Octet

Value	Parameter Description
0x00	Set_Reserved_LT_ADDR command succeeded.
0x01-0xFF	Set_Reserved_LT_ADDR command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions.

*LT\_ADDR*: Size: 1 Octet

Value	Parameter Description
0x01-0x07	LT_ADDR whose reservation the Host has requested to cancel.
0x00, 0x08-0xFF	Reserved for future use

#### Event(s) generated (unless masked away):

When the Delete\_Reserved\_LT\_ADDR command has completed, a Command Complete event shall be generated.

### 7.3.88 Set Connectionless Slave Broadcast Data Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connectionless_Slave_Broadcast_Data	0x0076	LT_ADDR, Fragment, Data_Length, Data	Status, LT_ADDR

#### Description:

The Set\_Connectionless\_Slave\_Broadcast\_Data command provides the ability for the Host to set Connectionless Slave Broadcast data in the BR/EDR Controller. This command may be issued at any time after an LT\_ADDR has been reserved regardless of whether connectionless slave broadcast mode has been enabled or disabled by the Enable parameter in the Set\_Connectionless\_Slave\_Broadcast command. If the command is issued without the LT\_ADDR reserved, the Unknown Connection Identifier (0X02) error code shall be returned.

If connectionless slave broadcast mode is disabled, this data shall be kept by the BR/EDR Controller and used once connectionless slave broadcast mode is enabled. If connectionless slave broadcast mode is enabled, and this command is successful, this data will be sent starting with the next Connectionless Slave Broadcast instant.

The Data\_Length field may be zero, in which case no data needs to be provided.

The Host may fragment the data using the Fragment field in the command. If the combined length of the fragments exceeds the capacity of the largest allowed packet size specified in the Set Connectionless Slave Broadcast command, all fragments associated with the data being assembled shall be discarded and the Invalid HCI Command Parameters error (0x12) shall be returned.

#### Command Parameters:

*LT\_ADDR*:

Size: 1 Octet

Value	Parameter Description
0x01-0x07	LT_ADDR on which to send Connectionless Slave Broadcast data
0x00, 0x08-0xFF	Reserved for future use

*Connectionless Slave Broadcast***Fragment:****Size: 1 Octet**

<b>Value</b>	<b>Parameter Description</b>
0x00	Continuation fragment
0x01	Starting fragment
0x02	Ending fragment
0x03	No fragmentation (single fragment)
0x04-0xFF	Reserved for future use

**Data\_Length:****Size: 1 Octet**

<b>Value</b>	<b>Parameter Description</b>
0xXX	Length of the Data field

**Data:****Size: Data\_Length Octets**

<b>Value</b>	<b>Parameter Description</b>
Variable	Data to send in future Connectionless Slave Broadcast packets. This data will be repeated in future Connectionless Slave Broadcast instants until new data is provided

**Return Parameters:****Status:****Size: 1 Octet**

<b>Value</b>	<b>Parameter Description</b>
0x00	Set_Connectionless_Slave_Broadcast_Data command succeeded.
0x01-0xFF	Set_Connectionless_Slave_Broadcast_Data command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions

**LT\_ADDR:****Size: 1 Octet**

<b>Value</b>	<b>Parameter Description</b>
0x01-0x07	LT_ADDR on which Connectionless Slave Broadcast data will be sent
0x00, 0x08-0xFF	Reserved for future use

**Event(s) generated (unless masked away):**

When the Set\_Connectionless\_Slave\_Broadcast\_Data command has completed, a Command Complete event shall be generated.

### 7.3.89 Read Synchronization Train Parameters Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Synchronization_Train_Parameters	0x0077	None	Status, Sync_Train_Interval, synchronization_trainTO, Service_Data

#### Description:

The Read\_Synchronization\_Train\_Parameters command returns the currently configured values for the Synchronization Train functionality in the master's BR/EDR Controller. This command may be issued at any time.

#### Command Parameters:

None.

#### Return Parameters:

##### Status:

Size: 1 Octet

Value	Parameter Description
0x00	Read_Synchronization_Train_Parameters command succeeded.
0x01-0xFF	Read_Synchronization_Train_Parameters command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions.

##### Sync\_Train\_Interval:

Size: 2 Octets

Value	Parameter Description
N = 0XXXXXX	Interval in slots between consecutive Synchronization Train events on the same channel. Range: 0x0020-0xFFFF; only even values are valid

##### synchronization\_trainTO:

Size: 4 Octets

Value	Parameter Description
N = 0XXXXXXXX	Duration in slots to continue sending the synchronization train Range: 0x00000002 - 0x07FFFFFF; only even values are valid

##### Service\_Data:

Size: 1 Octet

Value	Parameter Description
0xXX	Host provided value included in Synchronization Train packet, octet 27; see <a href="#">Table 8.4 (Volume 2, Part B, Section 8.3.5)</a>

**Event(s) generated (unless masked away):**

When the BR/EDR Controller receives the Read\_Synchronization\_Train\_Parameters command, it shall send a Command Complete event to the Host.

### 7.3.90 Write Synchronization Train Parameters Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Synchronization_Train_Parameters	0x0078	Interval_Min, Interval_Max, synchronization_trainTO, Service_Data	Status, Sync_Train_Interval

#### Description:

The Write\_Synchronization\_Train\_Parameters command configures the Synchronization Train functionality in the BR/EDR Controller. This command may be issued at any time.

Note: The AFH\_Channel\_Map used in the Synchronization Train packets is configured by the Set\_AFH\_Channel\_Classification command and the local channel classification in the BR/EDR Controller.

Interval\_Min and Interval\_Max specify the allowed range of Sync\_Train\_Interval. Refer to [\[Vol. 2\], Part B, section 2.7.2](#) for a detailed description of Sync\_Train\_Interval. The BR/EDR Controller shall select an interval from this range and return it in Sync\_Train\_Interval. If the Controller is unable to select a value from this range, it shall return the Invalid HCI Command Parameters (0x12) error code.

Once started (via the Start\_Synchronization\_Train Command) the Synchronization Train will continue until synchronization\_trainTO slots have passed or Connectionless Slave Broadcast has been disabled.

#### Command Parameters:

##### Interval\_Min:

Size: 2 Octets

Value	Parameter Description
N = 0xFFFF	Minimum value allowed for the interval Sync_Train_Interval in slots. Range: 0x0020-0xFFFF; only even values are valid

##### Interval\_Max:

Size: 2 Octets

Value	Parameter Description
N = 0xFFFF	Maximum value allowed for the interval Sync_Train_Interval in slots. Range: 0x0020-0xFFFF; only even values are valid

*Connectionless Slave Broadcast**synchronization\_trainTO :**Size: 4 Octets*

<b>Value</b>	<b>Parameter Description</b>
N = 0XXXXXXXXX	Duration in slots to continue sending the synchronization train Range: 0x0000 0002-0x07FF FFFE; only even values are valid

*Service\_Data:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0xXX	Host provided value to be included in octet 27 of the Synchronization Train packet payload body; see <a href="#">Table 8.8 (Volume 2, Part B, Section 8.11.2)</a> .

**Return Parameters:***Status:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	Write_Synchronization_Train_Parameters command succeeded.
0x01-0xFF	Write_Synchronization_Train_Parameters command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions.

*Sync\_Train\_Interval:**Size: 2 Octets*

<b>Value</b>	<b>Parameter Description</b>
N = 0XXXXX	Interval in slots between consecutive Synchronization Train packets on the same channel. Range: 0x0020-0xFFFF; only even values are valid

**Event(s) generated (unless masked away):**

When the BR/EDR Controller receives the Write\_Synchronization\_Train\_Parameters command, it shall send a Command Complete event to the Host.

## 7.5 STATUS PARAMETERS

### 7.5.12 Set Triggered Clock Capture Command [New Section]

Command	OCF	Command Parameters	Return Parameters
HCI_Set_triggered_Clock_Capture	0x000D	Connection_Handle, Enable, Which_Clock, LPO_Allowed, Num_Clock_Captures_To_Filter	Status

#### Description:

The Set\_triggered\_Clock\_Capture command configures the BR/EDR Controller for triggered clock capturing.

Triggered clock capturing is enabled or disabled by the Enable parameter.

If the Which\_Clock value is 0, then the Connection\_Handle shall be ignored. If the Which\_Clock value is 1, then the Connection\_Handle shall be a valid ACL Connection\_Handle.

The LPO\_Allowed parameter informs the BR/EDR Controller whether the low power oscillator is allowed to be used, or not.

The Num\_Clock\_Captures\_To\_Filter parameter is used to filter triggered clock captures between sending Triggered Clock Capture events to the Host. When set to zero, all triggered clock captures shall result in a Triggered Clock Capture event sent to the Host. When set to a non-zero value, after every Triggered Clock Capture event, Num\_Clock\_Captures\_To\_Filter triggered clock captures in a row shall not trigger an event to be sent to the Host.

Note: An implementation should ensure that the rate of triggered clock captures does not overwhelm the HCI event queue and processing.

Note: See [\[Vol 2\] Part B, Section 1.1, Bluetooth Clock](#) for more information about the Bluetooth Clock.

#### Command Parameters:

*Connection\_Handle:* *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xFFFF	Connection Handle to be used to identify a connection. Range: 0x0000 – 0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

*Connectionless Slave Broadcast**Enable:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	Disable triggered clock capturing on the specified Connection Handle (Default)
0x01	Enable triggered clock capturing on the specified Connection Handle
0x02-0xFF	Reserved for future use

*Which\_Clock:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0xXX	0x00 = Local Clock (Connection_Handle does not have to be valid) 0x01 = Piconet Clock (Connection_Handle shall be valid) 0x02-0xFF = Reserved

*LPO\_Allowed:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	Controller shall not sleep (that is, clock accuracy shall be equal to or better than $\pm 20$ ppm)
0x01	Controller may sleep (that is, clock accuracy shall be equal to or better than $\pm 250$ ppm)
0x02-0xFF	Reserved for future use

*Num\_Clock\_Captures\_To\_Filter:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	All triggered clock captures result in a Triggered Clock Capture event sent to the Host
0x01-0xFF	Number of triggered clock captures filtered between sending a Triggered Clock Capture event to the Host.

**Return Parameters:***Status:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	Set_triggered_clock_capture command succeeded.
0x01-0xFF	Set_triggered_clock_capture command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions.

**Event(s) generated (unless masked away):**

When the Set\_Triggered\_Clock\_Capture command has completed, a Command Complete event shall be sent to the Host.

When Triggered Clock Capturing is enabled, Triggered Clock Capture events are returned until Triggered Clock Capturing is disabled.

## 7.7 EVENTS

### 7.7.66 Triggered Clock Capture Event [New Section]

Event	Event Code	Event Parameters
Triggered Clock Capture	0x4E	Connection_Handle, Which_Clock, Clock, Slot_Offset

#### Description:

The Triggered Clock Capture event is sent to indicate that a triggering event has occurred at the specified clock and offset value. The Which\_Clock parameter indicates whether the clock is local or a piconet clock. The Connection\_Handle parameter is used when the clock is a piconet clock to indicate which piconet's clock was reported.

The Clock parameter indicates the value of the selected clock at the instant of the triggering event, with bits 1 and 0 set to 0b.

The Slot\_Offset parameter indicates the number of microseconds (from 0 to 1249) from the instant at which the selected clock took the value Clock until the triggering event.

#### Event Parameters:

*Connection\_Handle:* *Size: 2 Octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000 – 0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

*Which\_Clock:* *Size: 1 Octet*

Value	Parameter Description
0xXX	0x00 = Local Clock (Connection_Handle does not have to be valid) 0x01 = Piconet Clock (Connection_Handle shall be valid) 0x02-0xFF = Reserved

*Clock:* *Size: 4 Octets (28 bits meaningful)*

Value	Parameter Description
0XXXXXXXXX	Bluetooth clock of the device requested with bits 1 and 0 set to 0b.

*Connectionless Slave Broadcast**Slot\_Offset:**Size: 2 Octets*

<b>Value</b>	<b>Parameter Description</b>
N = 0XXXX	Number of microseconds from the selected clock attaining the value Clock until the triggering event. Range: 0 to 1249.

**7.7.67 Synchronization Train Complete Event [New Section]**

<b>Event</b>	<b>Event Code</b>	<b>Event Parameters</b>
Synchronization Train Complete	0x4F	Status

**Description:**

The Synchronization Train Complete event indicates that the Start Synchronization Train command has completed.

**Event Parameters:***Status:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x00	Start Synchronization Train command completed successfully.
0x01-0xFF	Start Synchronization Train command failed. See <a href="#">Part D, Error Codes</a> , for error codes and descriptions.

### 7.7.68 Synchronization Train Received Event [New Section]

Event	Event Code	Event Parameters
Synchronization Train Received	0x50	Status, BD_ADDR, Clock_Offset, AFH_Channel_Map, LT_ADDR, Next_Broadcast_Instant, Connectionless_Slave_Broadcast_Interval, Service_Data

#### Description:

The Synchronization Train Received event provides information received from a synchronization train packet transmitted by a Connectionless Slave Broadcast transmitter with the given BD\_ADDR. If synchronization was successful, it provides the clock offset, AFH channel map, LT\_ADDR, next broadcast instant, broadcast interval, and service data as received from the synchronization train payload. If the command returns a status of 0x01-0xFF, then all other parameters are undefined and shall be ignored.

#### Event Parameters:

##### Status:

Size: 1 Octet

Value	Parameter Description
0x00	Received Synchronization Train command completed successfully.
0x01-0xFF	Received Synchronization Train command failed. See Part D, Error Codes, for error codes and descriptions.

##### BD\_ADDR:

Size: 6 Octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the Connectionless Slave Broadcast transmitter.

##### Clock\_Offset:

Size: 4 Octets(28 bits meaningful)

Value	Parameter Description
0XXXXXXXX	(CLKslave - CLKmaster) modulo $2^{27}$

*Connectionless Slave Broadcast**AFH\_Channel\_Map:**Size: 10 Octets (79 Bits Meaningful)*

<b>Value</b>	<b>Parameter Description</b>
0xXXXXXXXXXX XXXXXXXXXXXX XX	This parameter contains 80 1-bit fields. The $n^{th}$ such field (in the range 0 to 78) contains the value for channel $n$ : Channel $n$ is bad = 0 Channel $n$ is unknown = 1 The most significant bit (bit 79) is reserved and shall be set to 0 At least $N_{min}$ channels shall be marked as unknown. (See <a href="#">Baseband Specification, Section 2.3.1, on page 76</a> )

*LT\_ADDR:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0x01-0x07	LT_ADDR of Connectionless Slave Broadcast channel.
0x00, 0x08-0xFF	Reserved for future use

*Next\_Broadcast\_Instant:**Size: 4 Octets (28 bits meaningful)*

<b>Value</b>	<b>Parameter Description</b>
0xXXXXXXXXXX	CLK of a future broadcast on this channel

*Connectionless\_Slave\_Broadcast\_Interval:**Size: 2 Octets*

<b>Value</b>	<b>Parameter Description</b>
0xXXXX	Interval between Connectionless Slave Broadcast instants in slots. Range: 0x0002-0xFFFF; only even values are valid

*Service\_Data:**Size: 1 Octet*

<b>Value</b>	<b>Parameter Description</b>
0xXX	Value from octet 27 of the Synchronization Train packet; see <a href="#">Table 8.4 (Volume 2, Part B, Section 8.3.5)</a>

### 7.7.69 Connectionless Slave Broadcast Receive Event [New Section]

Event	Event Code	Event Parameters
Connectionless Slave Broadcast Receive	0x51	BD_ADDR, LT_ADDR, CLK, Offset, Receive_Status, Fragment, Data_Length, Data

#### Description:

The Connectionless Slave Broadcast Receive event shall be sent by the BR/EDR Controller every Connectionless Slave Broadcast Instant on which the BR/EDR Controller is scheduled to receive a Connectionless Slave Broadcast packet. If the packet is not received successfully, the event returns a Receive\_Status of 0x01. Otherwise, the event returns the payload Data along with the Piconet Clock and the offset from the local CLKN when the packet was received.

The BR/EDR Controller shall send multiple Connectionless Slave Broadcast Receive events if the length of the received data exceeds the capacity of a single Connectionless Slave Broadcast Receive event. The fragments shall be marked as starting, continuation, or ending to allow the Host to reassemble the received packet. Only a single event shall be generated for a Connectionless Slave Broadcast instant on which a Connectionless Slave Broadcast packet was scheduled for reception but the BR/EDR Controller failed to successfully receive it.

#### Event Parameters:

##### BD\_ADDR:

Size: 6 Octets

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the broadcasting master device

##### LT\_ADDR:

Size: 1 Octet

Value	Parameter Description
0x01-0x07	LT_ADDR of the Connectionless Slave Broadcast
0x00, 0x08-0xFF	Reserved for future use

##### CLK:

Size: 4 Octets (28 bits meaningful)

Value	Parameter Description
0XXXXXXXXXX	CLK when Connectionless Slave Broadcast data was received

*Connectionless Slave Broadcast***Offset:****Size: 4 Octets (28 bits meaningful)**

<b>Value</b>	<b>Parameter Description</b>
0xXXXXXXXX	(CLKslave – CLKmaster) modulo $2^{27}$

**Receive Status:****Size: 1 Octet**

<b>Value</b>	<b>Parameter Description</b>
0x00	Packet received successfully
0x01	Packet not received successfully (Fragment, Data_Length, and Data fields invalid)
0x02-0xFF	Reserved for future use

**Fragment:****Size: 1 Octet**

<b>Value</b>	<b>Parameter Description</b>
0x00	Continuation fragment
0x01	Starting fragment
0x02	Ending fragment
0x03	No fragmentation (single fragment)
0x04-0xFF	Reserved for future use

**Data\_Length:****Size: 1 Octet**

<b>Value</b>	<b>Parameter Description</b>
0XX	Length of Data field

**Data:****Size: Data\_Length Octets**

<b>Value</b>	<b>Parameter Description</b>
Variable	Data received from a Connectionless Slave Broadcast packet.

### 7.7.70 Connectionless Slave Broadcast Timeout Event [New Section]

Event	Event Code	Event Parameters
Connectionless Slave Broadcast Timeout	0x52	BD_ADDR, LT_ADDR

#### Description:

On the Connectionless Slave Broadcast Receiver, the Connectionless Slave Broadcast Timeout event indicates to the Host that the BR/EDR Controller has lost synchronization with the Connectionless Slave Broadcast because no Connectionless Slave Broadcast packets have been received for the timeout interval,  $CSB\_supervisionTO$ , specified in the Set Connectionless Slave Broadcast Receive command.

On the Connectionless Slave Broadcast Transmitter, the Connectionless Slave Broadcast Timeout event indicates to the Host that the BR/EDR Controller has been unable to transmit a Connectionless Slave Broadcast packet for the timeout interval,  $CSB\_supervisionTO$ , specified in the Set Connectionless Slave Broadcast command.

#### Event Parameters:

*BD\_ADDR*: Size: 6 Octets

Value	Parameter Description
0xXXXXXXXXXXXXX	BD_ADDR of the broadcasting master device

*LT\_ADDR*: Size: 1 Octet

Value	Parameter Description
0x01-0x07	LT_ADDR of the Connectionless Slave Broadcast
0x00, 0x08-0xFF	Reserved for future use

### 7.7.71 Truncated Page Complete Event [New Section]

Event	Event Code	Event Parameters
Truncated Page Complete	0x53	Status, BD_ADDR

#### Description:

The Truncated Page Complete event indicates to the Host that a Truncated Page command completed. Truncated Paging is considered to be successful when a slave page response ID packet has been received by the local BR/EDR Controller. See [Vol 2, Part B, section 8.3.3](#) for more information.

A Truncated Page Complete event shall always be sent for each Truncated Page Command. If the Host issues a Truncated Page Cancel command before the Controller returns the Truncated Page Complete event, then the Truncated Page Complete event shall be sent after the Command Complete event for the Truncated Page Cancel command. If the cancellation was successful, the Truncated Page Complete event shall be generated with the error code Unknown Connection Identifier (0x02).

#### Event Parameters:

*Status:* *Size: 1 Octet*

Value	Parameter Description
0x00	Truncated_Page command completed successfully.
0x01-0xFF	Truncated_Page command failed. See <a href="#">Part D, Error Codes</a> for a list of error codes and descriptions.

*BD\_ADDR:* *Size: 6 Octets*

Value	Parameter Description
0XXXXXXXXXXXXX	BD_ADDR of the paged (slave) device

### 7.7.72 Slave Page Response Timeout Event [New Section]

Event	Event Code	Event Parameters
Slave Page Response Timeout	0x54	

#### Description:

The Slave Page Response Timeout event indicates to the Host that a slave page response timeout has occurred in the BR/EDR Controller. Note: this event will be generated if the slave BR/EDR Controller responds to a page but does not receive the master FHS packet (see [Baseband, Section 8.3.3](#)) within  $\text{pagerespTO}$ .

#### Event Parameters:

None

*Connectionless Slave Broadcast*

### 7.7.73 Connectionless Slave Broadcast Channel Map Change Event [New Section]

Event	Event Code	Event Parameters
Connectionless Slave Broadcast Channel Map Change	0x55	Channel_Map

#### Description:

The Connectionless Slave Broadcast Channel Map Change event is sent by the master's BR/EDR Controller to the master's Host to indicate that the master's BR/EDR Controller has moved to a new AFH channel map for the PBD logical link.

After an AFH channel map change takes effect for the PBD logical link, the Connectionless Slave Broadcast Transmitter BR/EDR Controller shall send this event to the Host. Upon reception of this event, the Host may restart the synchronization train to allow receivers to obtain the updated AFH channel map.

This event shall also be sent if the Host issues a Set\_AFH\_Channel\_Classification command which causes the Connectionless Slave Broadcast Channel Map to change.

#### Event Parameters:

*Channel\_Map:* Size: 10 Octets (79 Bits Meaningful)

Value	Parameter Description
0xFFFFFFFFXX	This parameter contains 80 1-bit fields.
XXXXXXXXXX	The $n^{th}$ (starting from 0) such field (in the range 0 to 78) contains the value for channel $n$ . Bit 79 is reserved (set to 0 when transmitted and ignored when received)
XX	The 1-bit field is interpreted as follows: 0: channel $n$ is unused 1: channel $n$ is used

### 7.7.74 Inquiry Response Notification Event [New Section]

Event	Event Code	Event Parameters
Inquiry Response Notification	0x56	LAP, RSSI

#### Description:

The Inquiry Response Notification event indicates to the Host that the BR/EDR Controller responded to an Inquiry message. The LAP parameter in the event indicates the LAP used to create the access code received. The parameter may be used by the Host to determine which access code was used in cases where the BR/EDR Controller is performing inquiry scan on multiple inquiry access codes using parallel scanning or sequential scanning. See [Vol. 3, Part C, section 4.1.2.1](#) for details on sequential and parallel scanning.

The LAP parameter returned by the BR/EDR Controller shall be one of the LAPs currently enabled. LAPs are enabled via the `Write_Current_IAC_LAP` command.

The RSSI parameter indicates the signal strength of the received ID packet.

#### Event Parameters:

*LAP:* *Size: 3 Octets*

Value	Parameter Description
0x9E8B00– 0x9E8B3F	LAP from which the IAC was derived; see <a href="#">Bluetooth Assigned Numbers</a> .

*RSSI:* *Size: 1 Octet*

Value	Parameter Description
N = 0XX Signed Integer	Size: 1 Octet (signed integer) Range: -100 ≤ N ≤ 20, +127 indicates unknown RSSI Units: dBm

## 6 VOL 2, PART F (MESSAGE SEQUENCE CHARTS) CHANGES:

### 9 CONNECTIONLESS SLAVE BROADCAST SERVICES [New Section]

Figure 9.1 illustrates the Truncated Page procedure.

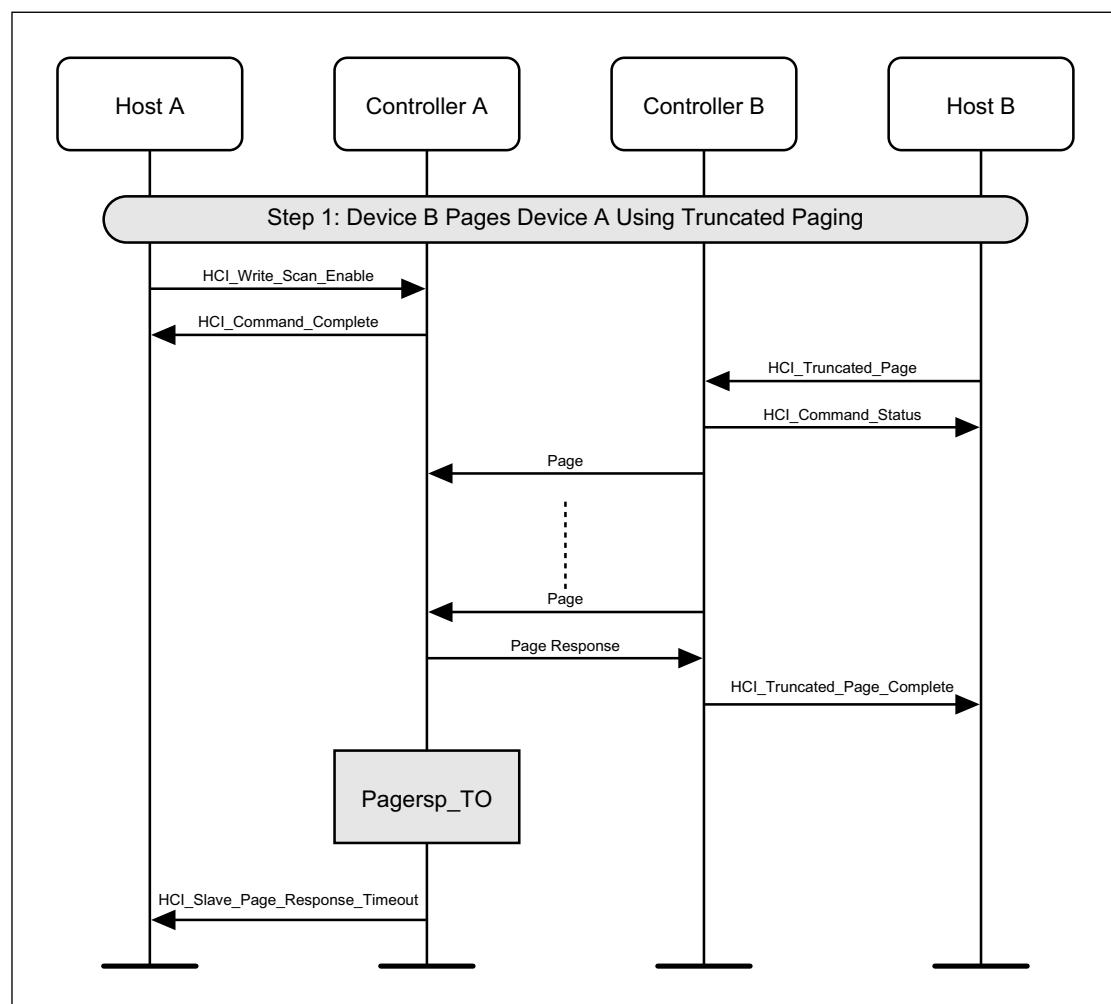
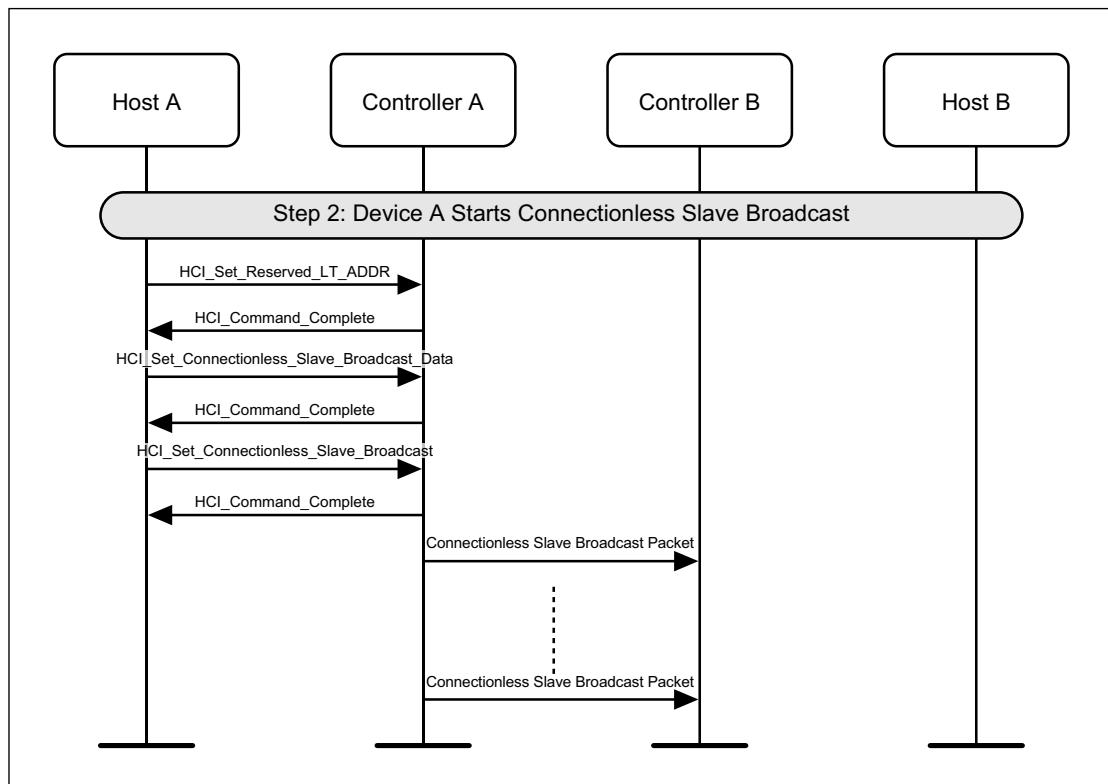


Figure 9.1: Truncated Paging MSC.

*Connectionless Slave Broadcast*

**Figure 9.2** illustrates how Device A starts transmitting Connectionless Slave Broadcast packets to Device B.



*Figure 9.2: Connectionless Slave Broadcast transmitter start MSC.*

*Connectionless Slave Broadcast*

**Figure 9.3** shows the Synchronization Train feature. Device A is the Connectionless Slave Broadcast Transmitter. Device B is the Connectionless Slave Broadcast Receiver.

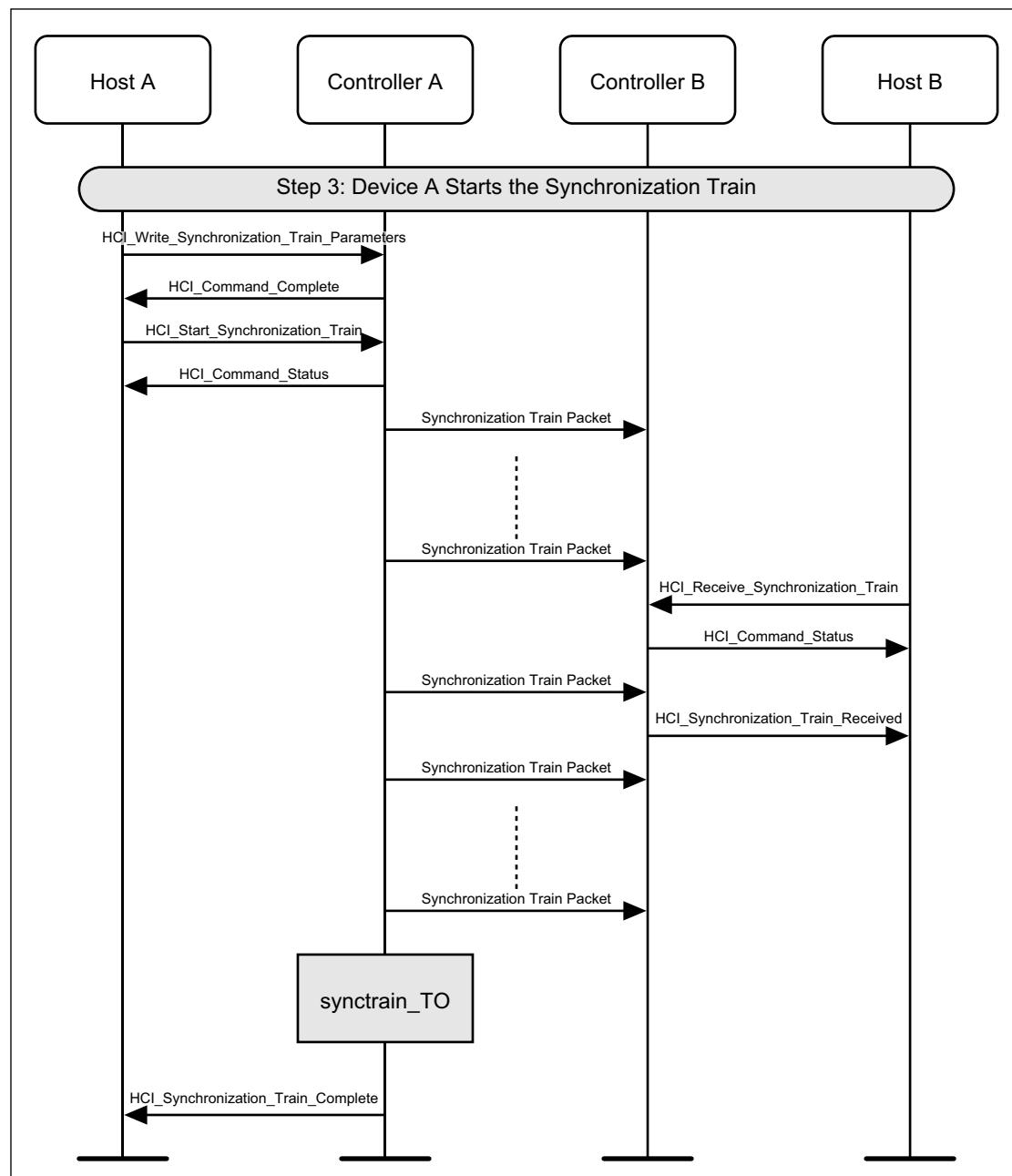
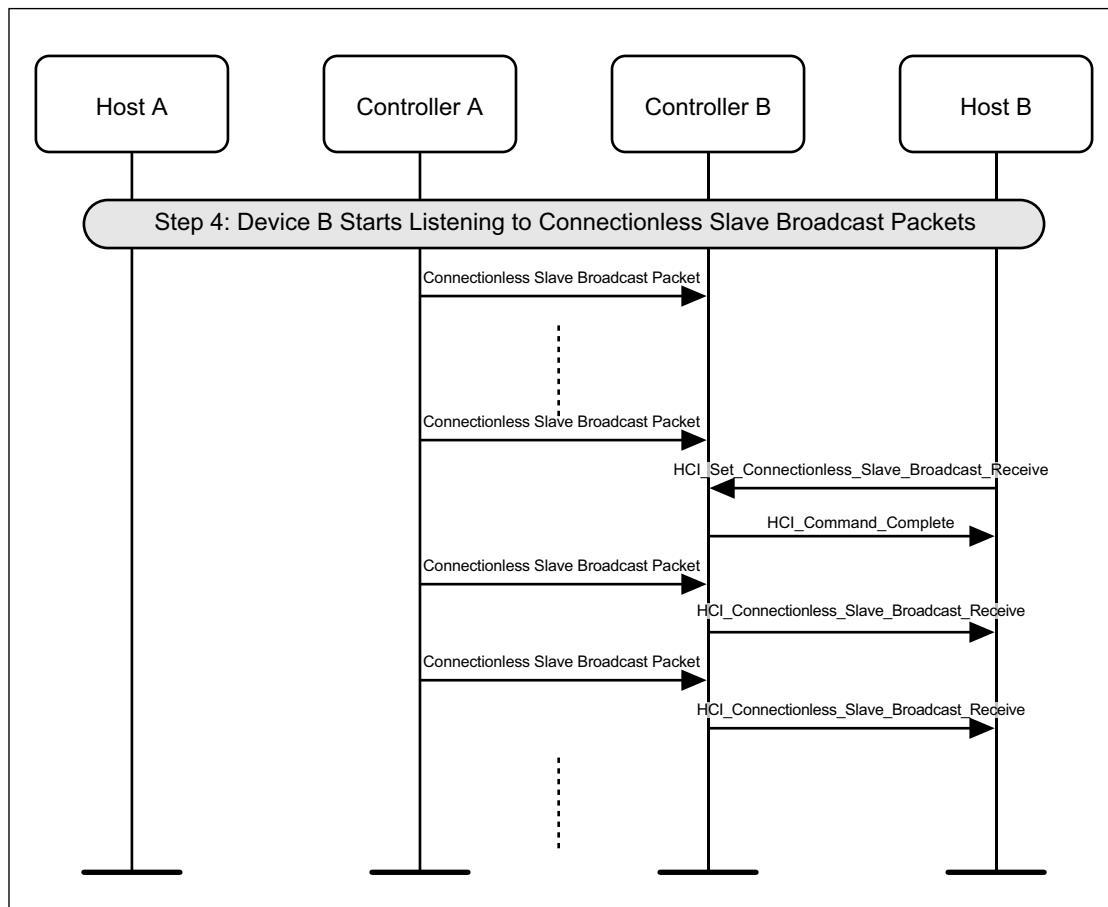


Figure 9.3: Synchronization Train MSC.

*Connectionless Slave Broadcast*

**Figure 9.4** illustrates how Device B starts receiving Connectionless Slave Broadcast packets from Device A.



*Figure 9.4: Connectionless Slave Broadcast Receiver start MSC.*

## 7 VOL 2, PART G (SAMPLE DATA) CHANGES:

---

### 11 CONNECTIONLESS SLAVE BROADCAST SAMPLE DATA [New Section]

---

This section contains an example of the DM3 packet used for the synchronization train (see [Vol 2] Part B, Section 8.11.2).

Packet header: (MSB...LSB)

-----  
LT\_ADDR = 0  
TYPE = 1010 (DM3)  
FLOW = 0  
ARQN = 0  
SEQN = 0

Payload:

-----  
logical channel = 10 (binary) (L2CAP start or no fragmentation)  
payload length = 28 bytes  
flow = 0  
current CLK = 0x2345678  
next connectionless slave broadcast instant = 0x23457a0  
AFH channel map = all channels used except 16 and 42 to 47  
master BD\_ADDR = NAP 0xACDE, UAP 0x48, LAP 0x610316  
connectionless slave broadcast interval = 564 slots  
connectionless slave broadcast LT\_ADDR = 1  
service data = 0x69

AIR DATA

Packet header (including HEC, in transmitted bit order):

-----  
0000000000  
000111000111  
000  
000  
000  
11111111000000000000111

Payload

-----  
The data forming the payload consists of the following 32 octets  
(given in hexadecimal) in the order transmitted:

e2 00  
78 56 34 02  
d0 2b 1a 01  
ff ff fe ff ff 03 ff ff ff 7f

---

*Connectionless Slave Broadcast*

16 03 61 48 de ac  
34 02  
01  
69  
f2 85

The bit sequence transmitted (including FEC) will therefore begin:

0100011100 01001  
0000000001 10101  
1110011010 11011

and end:

0100111110 10001  
1000010000 00011

## 8 VOL 3, PART C (GENERIC ACCESS PROFILE) CHANGES:

### 4 MODES [Updated]

[Add rows to Table 4.1 as shown]

Procedure	Ref.	Support
Discoverability modes:	4.1	
Non-discoverable mode		C1
Discoverable mode		O
Limited discoverable mode		C3
General discoverable mode		C4
Connectability modes:	4.2	
Non-connectable mode		O
Connectable mode		M
Bondable modes:	4.3	
Non-bondable mode		O
Bondable mode		C2
<u>Synchronizability modes:</u>	4.4	
<u>Non-synchronizable mode</u>		M
<u>Synchronizable mode</u>		O
C1: If limited discoverable mode is supported, non-discoverable mode is mandatory, otherwise optional.		
C2: If the bonding procedure is supported, support for bondable mode is mandatory, otherwise optional.		
C3: If Discoverable mode is supported, support for Limited Discoverable mode is mandatory, otherwise optional.		
C4: If Discoverable mode is supported, support for General Discoverable mode is mandatory, otherwise optional		

Table 4.1; Conformance requirements related to modes defined in this section.

## 4.4 SYNCHRONIZABILITY MODES [New Section]

A Bluetooth device shall be either in non-synchronizable mode or synchronizable mode. The synchronization train procedure is defined in [Vol. 2] Part B, Section 2.7.2.

When a Bluetooth device is in synchronizable mode, it transmits timing and frequency information for its active Connectionless Slave Broadcast packets. When a Bluetooth device is non-synchronizable, timing and frequency information is not transmitted.

The host is able to configure the Synchronization Train interval based on trade-offs between bandwidth, potential interference to other devices, power consumption, and the desired time for a slave to receive a synchronization train packet.

### 4.4.1 Non-synchronizable Mode [New Section]

#### 4.4.1.1 Definition [New Section]

When a Bluetooth device is in non-synchronizable mode it shall never enter the **synchronization train** substate.

#### 4.4.1.2 Term on UI-level [New Section]

Bluetooth device is ‘non-synchronizable’ or in ‘non-synchronizable mode’.

### 4.4.2 Synchronizable Mode [New Section]

#### 4.4.2.1 Definition [New Section]

When a Bluetooth device is in synchronizable mode, it shall enter the **synchronization train** substate using a synchronization train interval of  $T_{GAP}(\text{Sync\_Train\_Interval})$ .

After being made synchronizable, the Bluetooth device shall be synchronizable for at least  $T_{GAP}(\text{Sync\_Train\_Duration})$ .

## 7 ESTABLISHMENT PROCEDURES [Updated]

[Update section 7 as follows]

	Procedure	Ref.	Support in A	Support in B
1	Link establishment	<a href="#">7.1</a>	M	M
2	Channel establishment	<a href="#">7.2</a>	O	M
3	Connection establishment	<a href="#">7.3</a>	O	O
<a href="#">4</a>	<a href="#">Synchronization establishment</a>	<a href="#">7.5</a>	<a href="#">O</a>	<a href="#">O</a>

Table 7.1: Establishment procedures.

The establishment procedures defined here do not include any discovery part. Before establishment procedures are initiated, the information provided during device discovery (in the FHS packet or the extended inquiry response packet of the inquiry response or in the response to a name request or in the synchronization train packet) ~~has to~~must be available in the initiating device. This information is:

- The Bluetooth Device Address (BD\_ADDR) from which the Device Access Code is generated;
- The system clock of the remote device;
- The page scan mode used by the remote device for Link establishment.

Additional information provided during device discovery that ~~is~~may be useful for making the decision to initiate an establishment procedure is:

- The Class of device;
- The Device name;
- The supported Service Classes.

## 7.5 SYNCHRONIZATION ESTABLISHMENT [New Section]

### 7.5.1 Purpose [New Section]

The purpose of the Synchronization Establishment procedure is for a device to receive synchronization train packets using the procedures in Vol. 2, Part B Section 2.7.3.

### 7.5.2 Term on UI Level [New Section]

'Bluetooth synchronization establishment'

### 7.5.3 Description [New Section]

In Figure 7.7 the receiving device (B) is attempting to receive synchronization train packets from device (A).

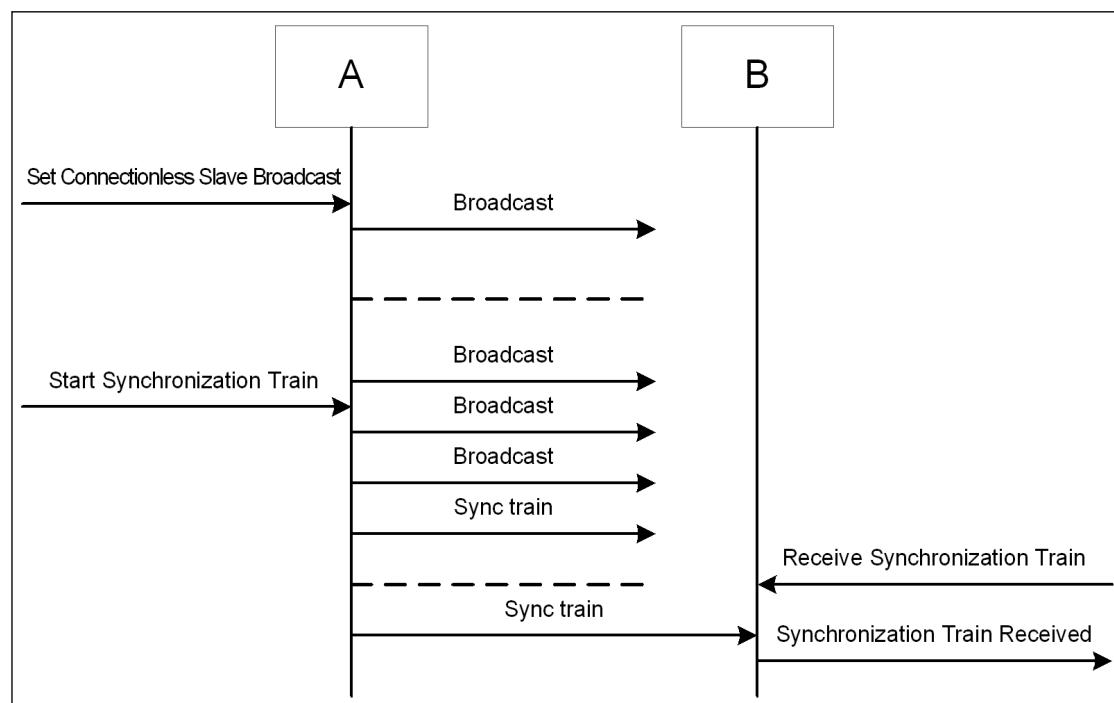


Figure 7.7: Synchronization establishment procedure.

#### 7.5.4 Conditions [New Section]

After receiving a synchronization train packet, the receiving device can listen to and receive profile data sent via Connectionless Slave Broadcast by device A.

Note that devices A and B may go through a separate Link Establishment procedure any time they desire to establish an ACL logical transport between each other. They may also use Connectionless Slave Broadcast procedures with an ACL logical transport already established.

The receiving device shall enter the **synchronization scan** substate using a scan interval of  $T_{GAP}(\text{Sync\_Scan\_Interval})$  and a scan window of  $T_{GAP}(\text{Sync\_Scan\_Window})$ .

## 17 APPENDIX A (NORMATIVE): TIMERS AND CONSTANTS [Updated]

[Add the following rows to the end of Table 17.1]

Timer name	Value	Description	Requirement or Recommendation
$T_{GAP}(\text{conn\_param\_timeout})$	30 s	Timer used in connection parameter update procedure	Recommended value
$T_{GAP}(\text{Sync\_Train\_Interval})$	80 ms	<u>Interval between Synchronization Train events</u>	<u>Recommended value</u>
$T_{GAP}(\text{Sync\_Train\_Duration})$	30.72 s	<u>Duration of synchronizability mode.</u>	<u>Required value</u>
$T_{GAP}(\text{Sync\_Scan\_Interval})$	320 ms	<u>Interval between the start of adjacent synchronization scan windows.</u>	<u>Recommended value</u>
$T_{GAP}(\text{Sync\_Scan\_Window})$	91.25 ms	<u>Duration of Synchronization scan window</u>	<u>Recommended value</u>

Table 17.1: Defined GAP timers.

Part C

# **UNENCRYPTED UNICAST CONNECTIONLESS DATA SUPPORT**

## CONTENTS

---

<b>1</b>	<b>Change Instance .....</b>	<b>127</b>
1.1	Change #1 - Vol 3, Part C (GAP).....	127

# 1 CHANGE INSTANCE

---

## 1.1 CHANGE #1 - VOL 3, PART C (GAP)

[Change section 5.2.2.8 Security Database as follows]

### 5.2.2.8 Security Database

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

- Level 3, for services with the following attributes:
  - MITM protection required
  - Encryption required
  - User interaction acceptable
- Level 2, for services with the following attributes:
  - MITM protection not required
  - Encryption required
- Level 1, for services with the following attributes:
  - MITM protection not required
  - Minimal user interaction desired
- Level 0: Service requires the following:
  - MITM protection not required
  - No encryption required
  - No user interaction required

Security Mode 4 Level 0 shall only be used for:

- a) L2CAP fixed signaling channels with CIDs 0x0001, 0x0003, and 0x003F
- b) SDP
- c) broadcast data sent on the connectionless L2CAP channel (CID 0x0002)
- d) services with the combinations of Service Class UUIDs and L2CAP traffic types listed in Core Specification Supplement, Part C.

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

- ~~Level 0: Service requires the following:~~

---

*Unencrypted Unicast Connectionless Data Support*

~~MITM protection not required~~

~~No encryption required~~

~~No user interaction required~~

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

~~0x1000 + BASE\_ID (Service Discovery Server)~~

Part D

## **FAST ADVERTISING INTERVAL**

## CONTENTS

---

<b>1</b>	<b>Change Instances .....</b>	<b>131</b>
1.1	Change #1 – Vol 3, Part C (GAP), Section 9.2.2, Non-Discoverable Mode .....	131
1.2	Change #2 – Vol 3, Part C (GAP), Section 9.3.2, Non-Connectable Mode.....	131
1.3	Change #3 – Vol 3, Part C (GAP), Section 9.3.11, Connection Establishment Timing Parameters.....	132
1.4	Change #4 – Vol 3, Part C (GAP), Appendix A (Normative): Timers and Constants .....	133

## 1 CHANGE INSTANCES

---

The changes below are related to the *GAP Connection Parameters Change* that was adopted as part of Core Specification Addendum (CSA) 3.

### 1.1 CHANGE #1 – VOL 3, PART C (GAP), SECTION 9.2.2, NON-DISCOVERABLE MODE

#### 9.2.2.2 Conditions

A device in the non-discoverable mode that sends advertising events shall not set the ‘LE General Discoverable Mode’ flag or ‘LE Limited Discoverable Mode’ flag in the Flags AD type. A Peripheral device in the non-discoverable mode may send non-connectable undirected advertising events or scannable undirected advertising events or may not send advertising packets.

If the Peripheral device in the non-discoverable mode sends non-connectable advertising events or scannable undirected advertising events then it is recommended that the Host configures the Controller as follows:

- The Host should set the advertising filter policy to either ‘process scan and connection requests only from devices in the White List’ or ‘process scan and connection requests from all devices’.
- The Host should set the advertising intervals as defined in Section 9.3.11.

### 1.2 CHANGE #2 – VOL 3, PART C (GAP), SECTION 9.3.2, NON-CONNECTABLE MODE

#### 9.3.2.2 Conditions

While a device is in the Peripheral role the device shall support the non-connectable mode. A device in the Central role cannot send advertising events, therefore the device is considered to support the non-connectable mode. A device in the Broadcaster or Observer role cannot establish a connection, therefore the device is considered to support the non-connectable mode.

A Peripheral device in the non-connectable mode may send non-connectable undirected advertising events or scannable undirected advertising events. In this case it is recommended that the Host configures the Controller as follows:

- The Host should set the advertising filter policy to either ‘process scan and connection requests only from devices in the White List’ or ‘process scan and connection requests from all devices’.
- The Host should set the advertising intervals as defined in Section 9.3.11.

## **1.3 CHANGE #3 – VOL 3, PART C (GAP), SECTION 9.3.11, CONNECTION ESTABLISHMENT TIMING PARAMETERS**

### 9.3.11.2 Conditions

A Central device starting a GAP connection establishment procedure should use the recommended scan interval  $T_{GAP}(\text{scan\_fast\_interval})$  and scan window  $T_{GAP}(\text{scan\_fast\_window})$  for  $T_{GAP}(\text{scan\_fast\_period})$ .

A Central device that is background scanning should use the recommended scan interval  $T_{GAP}(\text{scan\_slow\_interval1})$  and scan window  $T_{GAP}(\text{scan\_slow\_window1})$ . Alternatively the central may use  $T_{GAP}(\text{scan\_slow\_interval2})$  and scan window  $T_{GAP}(\text{scan\_slow\_window2})$ .

A Peripheral device ~~starting a GAP connection establishment mode~~ entering any of the following GAP Modes should use the recommended ~~advertisement advertising~~ interval  $T_{GAP}(\text{adv\_fast\_interval1})$  for  $T_{GAP}(\text{adv\_fast\_period})$ :

- Undirected Connectable Mode
- Limited Discoverable Mode and sending connectable undirected advertising events
- General Discoverable Mode and sending connectable undirected advertising events

A Peripheral device when entering any of the following GAP Modes and sending either non-connectable advertising events or scannable undirected advertising events should use the recommended advertising interval  $T_{GAP}(\text{adv\_fast\_interval2})$  for  $T_{GAP}(\text{adv\_fast\_period})$ :

- Non-Discoverable Mode
- Non-Connectable Mode
- Limited Discoverable Mode
- General Discoverable Mode

A Peripheral device that is background advertising in any GAP Mode other than the GAP Directed Connectable Mode should use the recommended ~~advertisement advertising~~ interval  $T_{GAP}(\text{adv\_slow\_interval})$ .

## 1.4 CHANGE #4 – VOL 3, PART C (GAP), APPENDIX A (NORMATIVE): TIMERS AND CONSTANTS

Timer name	Value	Description	Requirement or Recommendation
T <sub>GAP</sub> (adv_fast_period)	30 s	Minimum time to perform advertising when user initiated	Recommended value
T <sub>GAP</sub> (adv_fast_interval)	<del>30 ms to 60 ms</del>	<del>Minimum to maximum advertisement interval in any discoverable or connectable mode when user initiated</del>	<del>Recommended value</del>
T <sub>GAP</sub> (adv_fast_interval1)	<u>30 ms to 60 ms</u>	<u>Minimum to maximum advertising interval in the following GAP Modes when user initiated:</u> 1. <u>Undirected Connectable Mode</u> 2. <u>Limited Discoverable Mode and sending connectable undirected advertising events</u> 3. <u>General Discoverable Mode and sending connectable undirected advertising events</u>	<u>Recommended value</u>
T <sub>GAP</sub> (adv_fast_interval2)	<u>100 ms to 150 ms</u>	<u>Minimum to maximum advertising interval in the following GAP Modes when user initiated and sending either non-connectable advertising events or scannable undirected advertising events:</u> 1. <u>Non-Discoverable Mode</u> 2. <u>Non-Connectable Mode</u> 3. <u>Limited Discoverable Mode</u> 4. <u>General Discoverable Mode</u>	<u>Recommended value</u>
T <sub>GAP</sub> (adv_slow_interval)	1 s to 1.2 s	Minimum to maximum advertisement interval in any discoverable or connectable mode when background advertising	Recommended value

Table 1.1: Excerpt from Table 17.1: Defined GAP timers

Part E

## **eSCO RESERVED SLOTS CLARIFICATION**

## CONTENTS

---

<b>1</b>	<b>Change Instance .....</b>	<b>136</b>
1.1	Change #1 – Vol 2, Part B, Section 8.6.3 eSCO .....	136

# 1 CHANGE INSTANCE

---

## 1.1 CHANGE #1 – VOL 2, PART B, SECTION 8.6.3 eSCO

### 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 initiating 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 responding device shall apply the initialization method as indicated by the initialization flag. 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:

$$CLK_{27-1} \bmod T_{ESCO} = D_{ESCO} \quad \text{for initialization 1}$$

$$(\overline{CLK}_{27}, CLK_{26-1}) \bmod T_{ESCO} = D_{ESCO} \quad \text{for initialization 2}$$

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

---

*eSCO Reserved Slots Clarification*

| an eSCO retransmission window on a given eSCO LT\_ADDR only if it addressed that eSCO LT\_ADDR or did not transmit any packet in the immediately preceding eSCO reserved slots. A slave may transmit on an eSCO LT\_ADDR in the eSCO reserved slots only if the slave did not receive 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 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.

