IEEE Std 802.15.1™-2005
(Revision of
IEEE Std 802.15.1-2002)

IEEE Standards

802.15.1™

IEEE Standard for
Information technology—
Telecommunications and information exchange between systems—
Local and metropolitan area networks—
Specific requirements

Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE

3 Park Avenue, New York, NY 10016-5997, USA

14 June 2005

Print: SH95323
PDF: SS95323

Recognized as an
American National Standard (ANSI)

IEEE Std 802.15.1™-2005
(Revision of
IEEE Std 802.15.1-2002)

IEEE Standard for Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements

Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 31 May 2005

American National Standards Institute

Approved 14 February 2005

IEEE-SA Standards Board

Abstract: Methods for communicating devices in a personal area network (PAN) are covered in this standard.
Keywords: Bluetooth™, communications protocol, ISM, personal area network, WPAN

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 0-7381-4707-9 SH95323
PDF: ISBN 0-7381-4708-7 SS95323

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

NOTE—Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction

This introduction is not part of IEEE Std 802.15.1-2005, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements: Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs).

This standard defines services and protocol elements that permit the exchange of management information between stations associated in a personal area network (PAN).

Notice to users

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. A patent holder or patent applicant has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates, terms, and conditions of the license agreements offered by patent holders or patent applicants. Further information may be obtained from the IEEE Standards Department.

Participants

The following is a list of participants in the Wireless Personal Area Networks Working Group at the time this standard was approved.

Robert F. Heile, *Chair, IEEE 802.15*
Thomas M. Siep, *Chair, Task Group 1a*

Jon Adams
Roberto Aiello
Hiroshi Akagi
Masaaki Akahane
Richard Alfvin
James Allen
Anand Anandakumar
Jong Hoon Ann
Mikio Aoki
Michimasa Aramaki
Larry Arnett
Arun Arunachalam
Naiel Askar
Venkatl Bahl
Yasaman Bahreini
Daniel Bailey
Jay Bain
James Baker
Jaiganesh Balakrishnan
Paul Ballentine
John Barr
Anuj Batra
Dagnachew Birru
Kenneth Boehlke
Monique Bourgeois
Mark Bowles
Charles Brabenac

Vern Brethour
Ronald Brown
Peter Cain
Ed Callaway
Pat Carson
Kisoo Chang
Soo-Young Chang
Jonathon Cheah
Chee Wei Chew
Francois Po Shin Chin
Aik Chindapol
Sangsung Choi
Yun Choi
Craig Conkling
Robert Charles Cragie
David Cypher
Anand Dabak
Kai Dombrowski
Michael Dydyk
Jason Ellis
Shahriar Emami
Dwayne Escola
Mark W. Fidler
Chris Fisher
Reed Fisher
Jeff Foerster
Etsumi Fujita
Taketo Fukui
David Furuno
Pierre Gandolfo
Atul Garg
Michael Genossar
Vafa Ghazi
Ian Gifford
James Gilb
Tim Godfrey
Sorin Goldenberg
Paul Gorday
Martin Gravenstein
Evan Green
Bernd Grohmann
Assaf Gurevitz
Jose Gutierrez
Dongwoon Hahn
Thomas Hamilton
Yasuo Harada
Drew Harrington
Allen Heberling
Robert Heile
Barry Herold
Karl Heubaum
Reed Hinkel
Michael Hoghooghi
Srinath Hosur
Robert Huang
Eran Igler
Katsumi Ishii
Phil Jamieson
Ho-In Jeon
Jeyhan Karaoguz
Masami Katagiri
Joy Kelly
Michael Kelly
Stuart J. Kerry
Ryoji Kido
In Hwan Kim
Kyoung-A Kim
Myoung Soo Kim
Yongsuk Kim
Young Hwan Kim
Kursat Kimyacioglu
Patrick Kinney
Guenter Kleindl
Toshiya Kobashi
Ryuji Kohno
Bruce P. Kraemer
Rajeev Krishnamoorthy
Do-Hoon Kwon
John V. Lampe
Jim Lansford
Torbjorn Larsson
Hyung Soo Lee
Myung Lee
Nag Lee
Simon Lee
Woo-Kyung Lee
David Leeper
Liang Li
Susan Lin
Hui-Ling Lou
Akira Maeki
Steven March
Frederick Martin
Noriaki Matsuno
John McCorkle
Michael McInnis
Michael McLaughlin
James McLean
Daniel Meacham
Jim Meyer
Leonard Miller
Akira Miura
Shaomin Mo
Andreas Molisch
Antonio Mondragon
Mark Moore
Anthony Morelli
Said Moridi
Steven Morton
Marco Naeve
Ken Naganuma
Yves-Paul Nakache
Chiu Ngo
Erwin R. Noble
Christopher O'Connor
John C. O'Conor
Knut Odman
Hiroyo Ogawa
Eric Ojard
John B. Pardee
Jonghun Park
Joon Goo Park
Dave Patton
Marcus Pendergrass
Xiaoming Peng
Robert D. Poor
Paul Popescu
Pekka Ranta
Yaron Rashi
Gregg Rasor
Charles Razzell
Ivan Reede
Glyn Roberts
Richard Roberts
Martin Rofheart
Chris Rogers
Gerald Rogerson
Philippe Rouzet
Chandos Rypinski
Shin Saito
Tomoki Saito
John Santhoff
John Sarallo
Yasufumi Sasaki
Mark Schrader
Tom Schuster
Erik Schylander
Michael Seals
Kevin Shelby
Stephen J. Shellhammer
Shusaku Shimada
Cheol-Ho Shin
Yuichi Shiraki
Etan Shirron
Gadi Shor
William Shvodian
Thomas Siep
Kazimierz Siwiak
V. Somayazulu
Carl Stevenson
Rene Struik
Hiroto Sugahara
Robert A. Sutton
MItsuhiro Suzuki
Katsumi Takaoka
Kenichi Takizawa
Teik-Kheong Tan
Mike Tanahashi
James Taylor
Jerome Tjia
Kiyohito Tokuda
Jean Tsao
Stephen Turner
Hans van Leeuwen
Bhupender Virk
Thierry Walrant
Jerry Wang
Jing Wang
Fujio Watanabe
Katsumi Watanabe
Matthew Welborn
Richard Wilson
Gerald Wineinger
Stephen Wood
Edward G. Woodrow
David Yaish
Hirohisa Yamaguchi
Wonyong Yoon
Amos Young
Song-Lin Young
Serdar Yurdakul
Honggang Zhang
James Zyren

The following members of the balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Jon Adams
Jaemin Ahn
Roberto Aiello
Hiroshi Akagi
Richard Alfvin
James Allen
Richard Allen
Anand Anandakumar
Jong Hoon Ann
Mikio Aoki
Larry Arnett
Naiel Askar
Venkatb Bahl
Yasaman Bahreini
Jay Bain
James Baker
Jaiganesh Balakrishnan
Kannan Balakrishnan
Paul Ballentine
John Barr
Anuj Batra
Phil Beecher
Dagnachew Birru
Kenneth Boehlke
Herve Bonneville
John Boot
Bruce Bosco
Monique Bourgeois
Mark Bowles
Charles Brabenac
Jennifer Bray
David Brenner
Vern Brethour
Ronald Brown
Peter Cain
Ed Callaway
Pat Carson
Kisoo Chang
Soo-Young Chang
Jonathon Cheah
CheeWei Chew
Francois Chin
Yu-Chang Chiu
Sarm Cho
Sangsung Choi
Yun Choi
Chia-Chin Chong
Manoj Choudhary
Craig Conkling
Celestino Corral
Robert Cragie
David Cypher
Anand Dabak
Scott Davis
Joe Decuir
Javier Del Prado Pavon
Kai Dombrowski
Stefan Drude
Eryk Dutkiewicz
Michael Dydyk
Jason Ellis
Shahriar Emami
Yossi Erlich
Dwayne Escola
Mark Fidler
Chris Fisher
Reed Fisher
Kristoffer Fleming
Etsumi Fujita
Taketo Fukui
David Furuno
Ricardo Gandia Sanchez
Pierre Gandolfo
Michael Genossar
Vafa Ghazi
Ian Gifford
James Gilb
Tim Godfrey
Sung-Wook Goh
Sorin Goldenberg
Paul Gorday
Martin Gravenstein
Evan Green
Bernd Grohmann
Assaf Gurevitz
Dongwoon Hahn
Julian Hall
Thomas Hamilton
Yasuo Harada
Drew Harrington
Jeff Harris
Amer Hassan
Vann Hasty
Allen Heberling
Robert Heile
Barry Herold
Karl Heubaum
Jin-Meng Ho
Michael Hoghooghi
Srinath Hosur
Patrick Houghton
Chi-Hao Huang
Robert Huang
Xiaojing Huang
Eran Igler
Tetsushi Ikegami
Yeong Min Jang
Bruno Jechoux
Ho-In Jeon
Tzyy Hong Jiang (Chiang)
Peter Johansson
Jeyhan Karaoguz
Masami Katagiri
Joy Kelly
Michael Kelly
Stuart Kerry
Ryoji Kido
Haksun Kim
Inhwan Kim
Jae Young Kim
Myoung Kim
Yongsuk Kim
Young Hwan Kim
Youngsoo Kim
Kursat Kimyacioglu
Matthias Kindler
Patrick Kinney
Guenter Kleindl
Toshiya Kobashi
Noam Kogos
Ryuji Kohno
Rajeev Krishnamoorthy
Haim Kupershmidt
Yuzo Kuramochi
Do-Hoon Kwon
Taekyoung Kwon
Kang KyuMin
John Lampe
Jim Lansford
Torbjorn Larsson
David Leach
Dongjun Lee
Hyung Soo Lee
Kyung-Kuk Lee
Myung Lee
Nag Lee
Simon Lee
Woo-Kyung Lee
David Leeper
Fabrice Legrand
Israel Leibovich
Henry Li
Huan-Bang Li
Liang Li
Jie Liang
Susan Lin
Yong Liu
Hui-Ling Lou
Darryn Lowe
Steve Ma
Tadahiko Maeda
Akira Maeki
Frederick Martin
Abbie Mathew
Masafumi Matsumura
John McCorkle
Michael McInnis
Michael McLaughlin
James McLean
Daniel Meacham
Jim Meyer
Leonard Miller
Akira Miura
Hitoshi Miyasaka
Shaomin Mo
Andreas Molisch
Antonio Mondragon
Mark Moore
Anthony Morelli
Steven Morton
Marco Naeve
Ken Naganuma
Hiroyuki Nagasaka

Yves-Paul Nakache
Chiu Ngo
Erwin Noble
Mizukoshi Nobuyuki
Masaki Noda
Richard Noens
Christopher O'Connor
John (Jay) O'Conor
Knut Odman
Hiroyo Ogawa
Eric Ojard
Philip Orlik
Eiichiro Otobe
John Pardee
Bonghyuk Park
Jonghun Park
Joon Goo Park
Young Jin Park
Vijay Patel
Dave Patton
Miguel Pellon
Xiaoming Peng
Robert Poor
Paul Popescu
Clinton Powell
Raad Raad
Ajay Rajkumar
Pekka Ranta
Yaron Rashi
Gregg Rasor
Charles Razzell
Ivan Reede
Mark Rich
Yuko Rikuta
Benno Ritter
Terry Robar
Glyn Roberts
Richard Roberts
Martin Rofheart
Christopher Rogers
Gerald Rogerson
Jaeho Roh
Philippe Rouzet
Chandos Rypinski
Zafer Sahinoglu
Tomoki Saito
John Santhoff
John Sarallo
Yasufumi Sasaki
Sidney Schrum
Tom Schuster
Erik Schylander
Michael Seals
Huai-Rong Shao
Kevin Shelby
Stephen Shellhammer
Chih-Chung Shi
Shusaku Shimada
Cheol-Ho Shin
Yuichi Shiraki
Etan Shirron
Matthew Shoemake
Gadi Shor
William Shvodian
Thomas Siep
Michael Sim
Kazimierz Siwiak
Yoram Solomon
V. Somayazulu
Amjad Soomro
Carl Stevenson
Marinus Struik
Hiroto Sugahara
Robert Sutton
MItsuhiro Suzuki
Kazuaki Takahashi
Kenichi Takizawa
Teik-Kheong Tan
Mike Tanahashi
James Taylor
John Terry
Jerome Tjia
Kiyohito Tokuda
Jean Tsao
Stephen Turner
Oltac Unsal
Hans Van Leeuwen
Bhupender Virk
Timothy Wakeley
Thierry Walrant
Vivek Wandile
Jerry Wang
Jing Wang
Yunbiao Wang
Fujio Watanabe
Chris Weber
Matthew Welborn
Richard Wilson
Gerald Wineinger
Andreas Wolf
Timothy Wong
Stephen Wood
Patrick Worfolk
Xiaodong Wu
Yu-Ming Wu
David Yaish
Hirohisa Yamaguchi
Kamya Yekeh Yazdandoost
Wonyong Yoon
Yutaka Yoshida
Song-Lin Young
Hon Yung
Serdar Yurdakul
Honggang Zhang
Frank Xiaojun Zheng
Chunhui Zhu
James Zyren

When the IEEE-SA Standards Board approved this standard on 14 February 2005, it had the following membership:

Don Wright, *Chair*
Steve M. Mills, *Vice Chair*
Judith Gorman, *Secretary*

Chuck Adams
Stephen Berger
Mark D. Bowman
Joseph A. Bruder
Bob Davis
Roberto de Marca Boisson
Julian Forster*
Arnold M. Greenspan
Mark S. Halpin
Raymond Hapeman
Richard J. Holleman
Richard H. Hulett
Lowell G. Johnson
Joseph L. Koepfinger*
Hermann Koch
Thomas J. McGean
Daleep C. Mohla
Paul Nikolich
T. W. Olsen
Ronald C. Petersen
Gary S. Robinson
Frank Stone
Malcolm V. Thaden
Doug Topping
Joe D. Watson

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Alan Cookson, *NIST Representative*

Don Messina
IEEE Standards Project Editor

Contents

IEEE Standard for
Information technology—

Telecommunications and information exchange between systems—

Local and metropolitan area networks—

Specific requirements

Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)

1. Overview

Wireless personal area networks (WPANs) are used to convey information over short distances among a private, intimate group of participant devices. Unlike a wireless local area network (WLAN), a connection made through a WPAN involves little or no infrastructure or direct connectivity to the world outside the link. This allows small, power-efficient, inexpensive solutions to be implemented for a wide range of devices.

1.1 Scope

This standard defines physical layer (PHY) and medium access control (MAC) specifications for wireless connectivity with fixed, portable, and moving devices within or entering a personal operating space (POS). A POS is the space about a person or object that typically extends up to 10 m in all directions and envelops the person whether stationary or in motion.

The original goal of the IEEE 802.15.1 Task Group was to achieve a level of interoperability that could allow the transfer of data between a WPAN device and an IEEE 802.11™ device. Although this proved infeasible, IEEE Std 802.15.1-2005 does have mechanisms defined to allow better coexistence with IEEE 802.11b™ class of devices.

Both this standard and the previous version are based upon technology originally developed by the Bluetooth™ Special Interest Group (SIG).

1.2 WPAN definition

The term *WPAN* in this standard refers specifically to a wireless personal area network as used in this standard.

Specifically, this standard describes the following:

— The functions and services required by an IEEE 802.15.1-2005 device to operate within ad hoc networks.

— The following MAC procedures to support the asynchronous connectionless or connection-oriented (ACL) and synchronous connection-oriented (SCO) link delivery services:

  — The baseband (BB) layer, specifying the lower level operations at the bit and packet levels, e.g., forward error correction (FEC) operations, encryption, cyclic redundancy check (CRC) calculations, Automatic Repeat Request (ARQ) Protocol.

  — The link manager (LM) layer, specifying connection establishment and release, authentication, connection and release of SCO and ACL channels, traffic scheduling, link supervision, and power management tasks.

  — The Logical Link Control and Adaptation Protocol (L2CAP) layer, forming an interface to standard data transport protocols. It handles the multiplexing of higher layer protocols and the segmentation and reassembly (SAR) of large packets. The data stream crosses the LM layer, where packet scheduling on the ACL channel takes place. The audio stream is directly mapped on an SCO channel and bypasses the LM layer. The LM layer, though, is involved in the establishment of the SCO link. Between the LM layer and the application, control messages are exchanged in order to configure the IEEE 802.15.1-2005 transceiver for the considered application.

— The 2.4 GHz industrial, scientific, and medical (ISM) band PHY signaling techniques and interface functions that are controlled by the IEEE 802.15.1-2005 MAC. Requirements are defined for two reasons:

  — To provide compatibility between the radios used in the system.

  — To define the quality of the system.

Above the L2CAP layer may reside the Serial Cable Emulation Protocol based on ETSI TS 07.10 (RFCOMM), Service Discovery Protocol (SDP), Telephone Control Protocol specification (TCS), voice-quality channels for audio and telephony, and other network protocols. These protocols are necessary for interoperably for end-user products, but are outside the scope of this standard.

2. Normative references

The following referenced documents are indispensable for the application of this standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

2.1 IEEE documents

IEEE Std 802®, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.[1, 2]

IEEE Std 802.15.2™, IEEE Recommended Practice for Telecommunications and Information exchange between systems—Local and metropolitan area networks—Specific Requirements—Part 15.2: Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Band.

2.2 ISO documents

ISO/IEC 3309, Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures — Frame structure.[3]

ISO/IEC 7498-1, Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model.

ISO/IEC 8802-2, Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical link control.

ISO/IEC 10039, Information technology — Open Systems Interconnection — Local Area Networks — Medium Access Control (MAC) service definition.

ISO/IEC 15802-1, Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Common specifications — Part 1: Medium Access Control (MAC) service definition.

2.3 ITU documents

ITU-T Recommendation G.711, Pulse code modulation (PCM) of voice frequencies.[4]

ITU-T Recommendation O.150, Digital test patterns for performance measurements on digital transmission equipment.

ITU-T Recommendation O.153, Basic parameters for the measurement of error performance at bit rates below the primary rate.

---

[1]IEEE and 802 are registered trademarks in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

[2]IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://standards.ieee.org/).

[3]ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[4]ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

ITU-T Recommendation X.200, Information technology—Open systems interconnection—Basic reference model: The basic model.

2.4 Other documents

IETF RFC 1363, A Proposed Flow Specification.[5]

IETF RFC 1661, The Point-to-Point Protocol (PPP).

IrDA Object Exchange Protocol (IrOBEX), Version 1.2.[6]

[5] IETF documents are available from Internet Engineering Task Force (http://www.ietf.org/).
[6] IrDA documents are available from the Infrared Data Association (http://www.irda.org/).

3. Definitions

For the purposes of this standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition [B7][7], should be referenced for terms not defined in this clause.

3.1 active slave broadcast (ASB): The logical transport that is used to transport Logical Link Control and Adaptation Protocol (L2CAP) user traffic to all active devices in the piconet.

3.2 ad hoc network: A network typically created in a spontaneous manner. An ad hoc network requires no formal infrastructure and is limited in temporal and spatial extent.

3.3 authenticated device: A device whose identity has been verified during the lifetime of the current link, based on the authentication procedure.

3.4 authentication: A generic procedure based on link management profile authentication that determines whether a link key exists or, on Link Manager Protocol (LMP) pairing, whether no link key exists.

3.5 authorization: A procedure where a user of a device grants a specific (remote) device access to a specific service. Authorization implies that the identity of the remote device can be verified through authentication.

3.6 authorize: The act of granting a specific device access to a specific service. It may be based upon user confirmation or given the existence of a trusted relationship.

3.7 baseband (BB): The part of the system that specifies or implements the medium access control (MAC) layer and physical layer (PHY) procedures to support the exchange of real-time voice, data information streams, and ad hoc networking between devices.

3.8 beacon train: A pattern of reserved slots within a basic or adapted piconet physical channel. Transmissions starting in these slots are used to resynchronize parked devices.

3.9 Bluetooth device address (BD_ADDR): The address used to identify a device conforming to this standard.

3.10 Bluetooth wireless technology: The general term used to describe the technolgy orginally developed by the Bluetooth Special Interest Group (SIG). It defines a wireless communication link, operating in the unlicensed industrial, scientific, and medical (ISM) band at 2.4 GHz using a frequency hopping transceiver. The link protocol is based on time slots.

3.11 bond: A relation between two devices defined by creating, exchanging, and storing a common link key. The bond is created through the bonding or Link Manager Protocol (LMP) pairing procedures.

3.12 channel: Either a physical channel or an Logical Link Control and Adaptation Protocol (L2CAP) channel, depending on the context.

3.13 connect (to service): The establishment of a connection to a service. If not already done, this also includes establishment of a physical link, logical transport, logical link, and Logical Link Control and Adaptation Protocol (L2CAP) channel.

3.14 connectable device: A device in range that periodically listens on its page scan physical channel and will respond to a page on that channel.

[7]The numbers in brackets correspond to the numbers of the bibliography in Annex A.

3.15 connected devices: Two devices in the same piconet and with a physical link between them.

3.16 connecting: A phase in the communication between devices when a connection between them is being established. (Connecting phase follows after the link establishment phase is completed.)

3.17 connection: A connection between two peer applications or higher layer protocols mapped onto a Logical Link Control and Adaptation Protocol (L2CAP) channel.

3.18 connection establishment: A procedure for creating a connection mapped onto a channel.

3.19 controller: A subsystem containing the physical layer (PHY), baseband (BB), resource controller, link manager (LM), device manager, and a host controller interface (HCI) conforming to this standard.

3.20 coverage area: The area where two devices can exchange messages with acceptable quality and performance.

3.21 creation of a secure connection: A procedure of establishing a connection, including authentication and encryption.

3.22 creation of a trusted relationship: A procedure where the remote device is marked as a trusted device. This includes storing a common link key for future authentication and pairing (if the link key is not available).

3.23 device: A device that is capable of short-range wireless communications using this standard.

3.24 device address: A 48-bit address used to identify each device.

3.25 device discovery: A procedure for retrieving the device address, clock, class-of-device field, and used page scan mode from discoverable devices.

3.26 discoverable device: A device in range that periodically listens on an inquiry scan physical channel and will respond to an inquiry on that channel. Discoverable devices are normally also connectable.

3.27 estimated clock (CLKE): Estimate of another device's clock. CLKE may be a slave's estimate of a master's clock, a paging devices's estimate of the paged device's clock, or other such use.

3.28 host: A computing device, peripheral, cellular telephone, access point to public switched telephone network (PSTN) or local area network (LAN), etc. A host attached to a controller may communicate with other hosts attached to their controllers as well.

3.29 host controller interface (HCI): A command interface to the baseband (BB) controller and link manager (LM) that provides access to hardware status and control registers and provides a uniform method of accessing the BB capabilities.

3.30 idle: Description of a device, as seen from a remote device, when no link is established between the devices.

3.31 inquiring device: A device that is carrying out the inquiry procedure.

3.32 inquiry: A procedure where a device transmits inquiry messages and listens for responses in order to discover the other devices that are within the coverage area.

3.33 inquiry scan: A procedure where a device listens for inquiry messages received on its inquiry scan physical channel.

3.34 isochronous data: Information in a stream where each information entity in the stream is bound by a time relationship to previous and successive entities.

3.35 known device: A device for which at least the Bluetooth device address (BD_ADDR) is stored.

3.36 link: Shorthand for a logical link.

3.37 link establishment: A procedure for establishing the default ACL link and hierarchy of links and channels between devices.

3.38 link key: A secret key that is known by two devices and is used in order to authenticate each device to the other.

3.39 LMP authentication: A procedure on the Link Manager Protocol (LMP) level for verifying the identity of a remote device. The procedure is based on a challenge-response mechanism using a random number, a secret key, and the Bluetooth device address (BD_ADDR) of the noninitiating device. The secret key used can be a previously exchanged link key.

3.40 LMP pairing: A procedure that authenticates two devices, based on a personal identification number (PIN), and subsequently creates a common link key that can be used as a basis for a trusted relationship or a (single) secure connection. The procedure consists of the following steps: creation of an initialization key (based on a random number and a PIN), creation and exchange of a common link key, and Link Manager Protocol (LMP) authentication based on the common link key.

3.41 logical channel: Identical to a Logical Link Control and Adaptation Protocol (L2CAP) channel, but deprecated due to inconsistent usage in IEEE Std 802.15.1-2002.

3.42 logical link: The lowest architectural level used to offer independent data transport services to clients of the system.

3.43 logical transport: Used to represent commonality between different logical links due to shared acknowledgment protocol and link identifiers.

3.44 L2CAP channel: A logical connection on the Logical Link Control and Adaptation Protocol (L2CAP) level between two devices serving a single application or higher layer protocol.

3.45 L2CAP channel establishment: A procedure for establishing a logical connection on the Logical Link Control and Adaptation Protocol (L2CAP) level.

3.46 master clock (CLK): Native clock of the piconet's master.

3.47 mode: A set of directives that defines how a device will respond to certain events.

3.48 name discovery: A procedure for retrieving the user-friendly name (the device name) of a connectable device.

3.49 native clock (CLKN): A 28-bit clock internal to a controller subsystem that ticks every 312.5 μs. The value of this clock defines the slot numbering and timing in the various physical channels.

3.50 packet: Format of aggregated bits that are transmitted on a physical channel.

3.51 page: The initial phase of the connection procedure where a device transmits a train of page messages until a response is received from the target device or a timeout occurs.

3.52 page scan: A procedure where a device listens for page messages received on its page scan physical channel.

3.53 paging device: A device that is carrying out the page procedure.

3.54 paired device: A device with which a link key has been exchanged (either before connection establishment was requested or during connecting phase).

3.55 parked device: A device operating in a basic mode piconet that is synchronized to the master, but has given up its default ACL logical transport.

3.56 parked slave broadcast (PSB): The logical transport that is used for communications from the master to parked slave devices. These communications may also be received by active devices.

3.57 participant in multiple piconets: A device that is concurrently a member of more than one piconet. It achieves this status using time division multiplexing (TDM) to interleave its activity on each piconet physical channel.

3.58 personal identification number (PIN): A user-friendly number that can be used to authenticate connections to a device before pairing has taken place.

3.59 physical channel: A channel characterized by synchronized occupancy of a sequence of radio frequency (RF) carriers by one or more devices. A number of physical channel types exist with characteristics defined for their different purposes.

3.60 physical link: A connection on the baseband (BB) level between two devices established using paging.

3.61 piconet: A collection of devices occupying a shared physical channel where one of the devices is the piconet master and the remaining devices are connected to it.

3.62 piconet physical channel: A channel that is divided into time slots in which each slot is related to a radio frequency (RF) hop frequency. Consecutive hops normally correspond to different RF hop frequencies and occur at a standard hop rate of 1600 hop/s. These consecutive hops follow a pseudo-random hopping sequence, hopping through a 79-RF channel set, or optionally fewer channels when adaptive frequency hopping (AFH) is in used.

3.63 piconet master: The device in a piconet whose clock and device address are used to define the piconet physical channel characteristics.

3.64 piconet slave: Any device in a piconet that is not the piconet master, but is connected to the piconet master, and that controls piconet timing and access by its transmissions to slaves.

3.65 prepaired device: A device with which a link key was exchanged and stored before link establishment.

3.66 scatternet: Two or more piconets that include one or more devices participating in more than one piconet.

3.67 service discovery (SD): Procedures for querying and browsing for services offered by or through another device.

3.68 service layer protocol: A protocol that uses a Logical Link Control and Adaptation Protocol (L2CAP) channel for transporting protocol data units (PDUs).

3.69 silent device: A device appears as silent to a remote device if it does not respond to inquiries made by the remote device.

3.70 trusted device: A paired device that is explicitly marked as trusted.

3.71 unknown device: A device for which no information (e.g., device address, link key) is stored.

3.72 unpaired device: A device for which there was no exchanged link key available before connection establishment was requested.

4. Acronyms and abbreviations

This clause contains two classes of acronyms and abbreviations. The first class is based on this and other standards and is the type usually found in IEEE standards. The second class refers to acronyms and abbreviations from the Bluetooth specification that are used by this standard. This second class is included in this standard as an aid to the reader.

4.1 Standard-based acronyms and abbreviations

ACK	acknowledge
ACL	asynchronous connection-oriented [logical transport]
ACL-C	ACL control [logical link] (LMP)
ACL-U	ACL user [logical link] (L2CAP)
ACO	authenticated ciphering offset
AFH	adaptive frequency hopping
AHS	adapted hop sequence
AR_ADDR	access request address
ARQ	automatic repeat request
ARQN	acknowledgment indication
ASB	active slave broadcast [logical transport]
ASB-U	active slave broadcast user [logical link] (L2CAP)
BB	baseband
BCH	Bose, Chaudhuri, and Hocquenghem
BD_ADDR	device address
BER	bit error rate
BT	bandwidth time
CAC	channel access code
CID	channel identifier
CL	connectionless
CLK	master clock
CLKE	estimated clock
CLKN	native clock
CODEC	coder decoder
COF	ciphering offset
CQDDR	channel quality-driven data rate
CRC	cyclic redundancy check
CVSD	continuous variable slope delta [modulation]
DA	destination address [field]

DAC	device access code
DCI	default check initialization
DH	data–high rate [packet]
DIAC	dedicated inquiry access code
DLL	data link layer
DM	data–medium rate [packet]
DUT	device under test
DV	data-voice [packet]
EN_RAND	encryption random number
ERP	ear reference point
eSCO	extended synchronous connection-oriented [logical transport]
eSCO-S	stream extended synchronous connection-oriented
EV	extended voice [packet]
ERTX	extended response timeout expired [timer]
FCS	frame check sequence
FEC	forward error correction
FHS	frequency hop synchronization
FHSS	frequency hopping spread spectrum
FIFO	first in first out
GAP	generic access profile
GFSK	Gaussian frequency shift keying
GIAC	general inquiry access code
HCI	host controller interface
HEC	header error check
HID	human interface device
HV	high-quality voice [packet]
IAC	inquiry access code
IN_RAND	initialization random number
IP	Internet Protocol
ISDN	integrated services digital network
ISM	industrial, scientific, and medical
L2CAP	Logical Link Control and Adaptation Protocol
LAP	lower address part
LC	link control [logical link]
LCID	local channel identifier
LCP	Link Control Protocol

LFSR	linear feedback shift register
LIAC	limited inquiry access code
LLC	logical link control
LLID	logical link identifier
LM	link manager
LMP	Link Manager Protocol
LPO	low-power oscillator
LR	loudness rating
LSB	least significant bit
LT_ADDR	logical transport address
M	master or mandatory
MAC	medium access control
MPS	maximum PDU payload size
MSB	most significant bit
MSC	message sequence chart
MSDU	MAC service data unit
MTU	maximum transmission unit
NAK	negative acknowledge
NAP	nonsignificant address part
NOP	no operation
O	optional
OBEX	Object Exchange Protocol
OCF	opcode command field
OGF	opcode group field
PCM	pulse code modulation
PDU	protocol data unit
PGA	programmable gain amplifier
PHT	pseudo-Hadamard transform
PIN	personal identification number
PM_ADDR	parked member address
PN	pseudo-random noise
POS	personal operating space
ppm	parts per million
PRBS	pseudo-random bit sequence
PSB	parked slave broadcast [logical transport]
PSB-C	parked slave broadcast control [logical link] (LMP)

PSB-U	parked slave broadcast user [logical link] (L2CAP)
PSM	protocol/service multiplexer
PSTN	public switched telephone network
QoS	quality of service
RAND	random number
RF	radio frequency
RFCMode	retransmission and flow control mode
RFCOMM	Serial Cable Emulation Protocol based on ETSI TS 07.10
RLR	receive loudness rating
RSSI	received signal strength indication
RX	receive
RTX	response timeout expired [timer]
S	slave
SA	source address [field]
SAP	service access point
SAR	segmentation and reassembly
SCO	synchronous connection-oriented [logical transport]
SCO-S	stream synchronous connection-oriented (unframed)
SD	service discovery
SDAP	service discovery applicaiton profile
SDP	Service Discovery Protocol
SDU	service data unit
SEQN	sequence number
SLR	send loudness rating
SRES	signed response
TCS	Telephony Control Protocol specification
TDD	time-division duplex
TDM	time-division multiplexing
TX	transmit
UAP	upper address part
u_int	unsigned integer
USB	universal serial bus

4.2 Bluetooth specification names

References to upper layer Bluetooth protocols use the abbreviations in Table 1.

Table 1—Abbreviations of the Bluetooth specification names

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

Table 1—Abbreviations of the Bluetooth specification names *(continued)*

Name	Reference	Placement in Bluetooth specification
RFCOMM	Serial Cable Emulation Protocol based on ETSI TS 07.10	vol 7 part A
SAP	SIM Access Profile Specification	vol 12 part C
SDAP	Service Discovery Application Profile Specification	vol 5 part B
SDP (1)	Service Discovery Protocol Specification (server)	vol 3 part B
SDP (2)	Service Discovery Protocol Specification (client)	vol 5 part A
SPP	Serial Port Profile Specification	vol 7 part B
Synch	Synchronization Profile Specification	vol 8 part D
TCI	Test Control Interface	vol 3 part D, section 2
TCP	Telephony Control Protocol Specification	vol 9 part A
UDI	Unrestricted Digital Information Profile Specification	vol 12 part B

5. General description

5.1 New features

Several new features are introduced in IEEE Std 802.15.1-2005. The major areas of improvement are as follows:

— Architectual overview
— Faster connection
— Adaptive frequency hopping (AFH)
— Extended SCO links
— Enhanced error detection and flow control
— Enhanced synchronization capability
— Enhanced flow specification

These feature descriptions are incorporated into the text within this standard.

5.2 Changes in wording

Two general classes of changes to the wording of IEEE Std 802.15.1-2002 have been done in IEEE Std 802.15.1-2005. They are a conformance to the formalization of the language by using conventions established by the IEEE and a regularization of Bluetooth wireless technology-specific terms.

5.2.1 IEEE language update

Many portions of IEEE Std 802.15.1-2002 used imprecise or inaccurate terms to describe attributes of the protocol. This standard now conforms to the correct usage of the key verbs that describe requirements. Table 2 is a summary of the verbs whose usage was regularized based on the “IEEE Style Guide” [B8].

Table 2—IEEE nomenclature

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

5.2.2 Nomenclature changes

The nomenclature used to describe the protocol has also been changed in IEEE Std 802.15.1-2005. Several terms were used more than once, for different concepts in IEEE Std 802.15.1-2002. The text has been updated to regularize this standard-specific usage. The nomenclature is introduced together with the new features in the new architecture subclause (see 6.2).

5.3 Structure changes

This standard has been significantly restructured for better consistency and readability. The most important structure changes have been performed in BB, Link Manager Protocol (LMP), host controller interface (HCI), and L2CAP. The text in these clauses have been rearranged to provide the following:

— Presentation of the information in a more logical progression
— Removal of redundant text and requirements
— Consolidation of BB-related requirements (e.g., moving the BB timers and audio subclauses into Clause 8 about the BB)

5.4 Deprecated features

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

The functions no longer recommended are being deprecated. The term *deprecation* does not mean that these functions are no longer allowed, but that they are no longer recommended as the best way of performing a given function.

Features deprecated in IEEE Std 802.15.1-2005 are as follows:

— The use of unit keys for security
— Optional paging schemes
— The 23-channel hopping sequence

6. Architecture

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

The term *core system* is used in this clause to denote the combination of a radio frequency (RF) transceiver, BB, and protocol stack. The system offers services that enable the connection of devices and the exchange of a variety of classes of data between these devices.

This clause of this standard provides an overview of the system architecture, communication topologies, and data transport features. This clause is informative.

6.1 General description

The RF (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 frequency hopping spread spectrum (FHSS) carriers. RF operation uses a shaped, binary frequency modulation to minimize transceiver complexity. The symbol rate is 1 Msymbol/s supporting the bit rate of 1 Mb/s.

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

Devices in a piconet use a specific frequency hopping pattern, which is algorithmically determined by fields in the device address and the clock of the master. The basic hopping pattern is a pseudo-random ordering of the 79 frequencies in the ISM band. The hopping pattern may be adapted to exclude a portion of the frequencies that are used by interfering devices. The adaptive hopping technique improves coexistence with static (nonhopping) ISM systems when these are collocated and implements some of the recommendations of IEEE Std 802.15.2-2003.

The physical channel is subdivided into time units known as *slots*. Data are transmitted between devices in packets, which 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 the reception of packets. This standard 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 6.4.4 through 6.5, but are introduced here to aid the understanding of the remainder of this clause.

Within a physical channel, a physical link is formed between any two devices that transmit packets in either direction between them. In a piconet physical channel, there are restrictions on which devices may form a physical link. There is a physical link between each slave and the master. Physical links are not formed directly between the slaves in a piconet.

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

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

The LM function uses LMP to control the operation of devices in the piconet and provide services to manage the lower architectural levels (i.e., PHY and BB). The LMP is carried only on the default ACL logical transport and the default broadcast logical transport.

Above the BB, L2CAP provides a channel-based abstraction to applications and services. It carries out segmentation and reassembly (SAR) of application data and multiplexing and demultiplexing 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 may be carried on any logical link that supports the L2CAP.

6.2 Core system architecture

The core system covers the four lowest segments and associated protocols defined by this standard, and the overall profile requirements are specified in the generic access profile (GAP) (see Annex B). A complete application generally requires a number of additional service and higher layer protocols that are defined in the Bluetooth specification and are not described in this standard. The core system architecture is shown in Figure 1.

Core system architecture shows the four lowest layers, each with its associated communication protocol. The lowest three layers are sometimes grouped into a subsystem (known as the *controller*). This is a common implementation involving a standard physical communications interface (i.e., the host controller interface or HCI) and remainder of the system. This includes the L2CAP, service, and higher layers (known as the *host*). Although this interface is optional, the architecture is designed to allow for its existence and characteristics. This standard enables interoperability between independent systems by defining the protocol messages exchanged between equivalent layers and also interoperability between independent subsystems by defining a common interface between controllers and hosts.

A number of functional blocks are shown in Figure 1 and the path of services and data between these. The functional blocks shown in the diagram are informative; in general, this standard does not define the details of implementations except where this is required for interoperability. Thus the functional blocks in Figure 1 are shown in order to aid description of the system behavior. An implementation may be different from the system shown in Figure 1.

Standard interactions are defined for all interdevice operation, where devices exchange protocol signalling according to this standard. The core system protocols are the Radio Frequency (RF) Protocol, Link Control Protocol (LCP), LMP, and L2CAP, all of which are fully defined in subsequent parts of this standard.

The core system offers services through a number of service access points (SAPs) that are shown in Figure 1 as ellipses. These services consist of the basic primitives that control the core system. The services can be split into three types:

— Device control services that modify the behavior and modes of a device
— Transport control services that create, modify, and release traffic bearers (channels and links)
— Data services that are used to submit data for transmission over traffic bearers

It is common to consider the first two as belonging to the C-plane and the last as belonging to the U-plane.

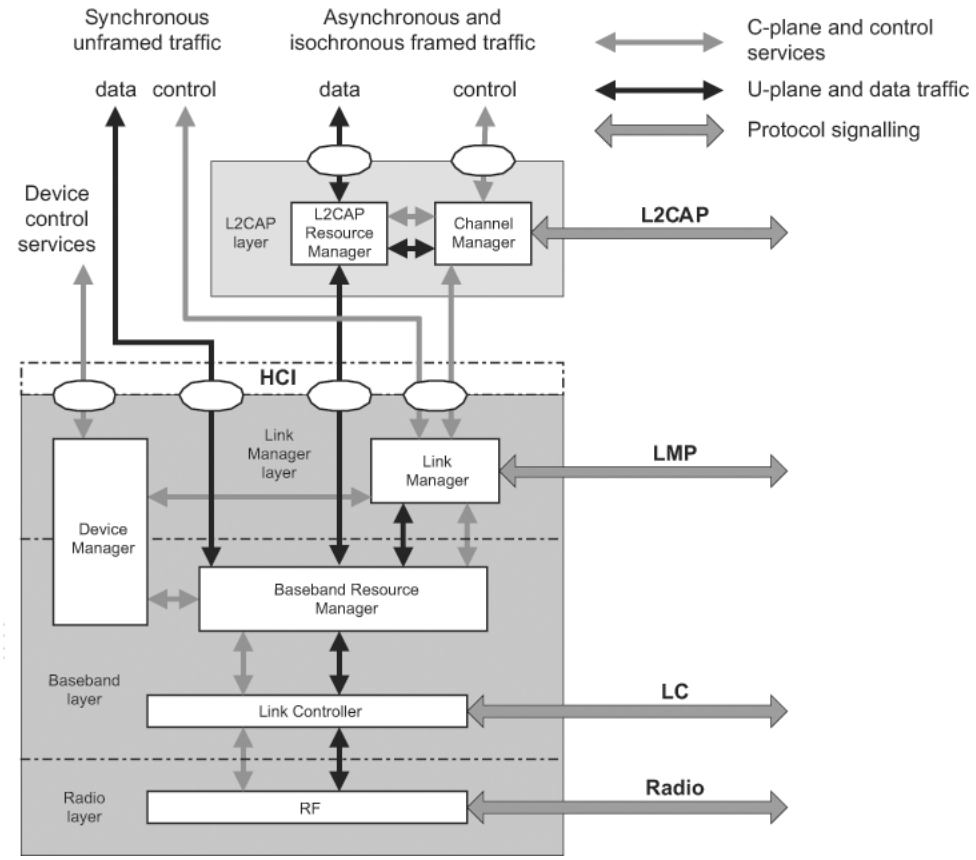


Figure 1—Core system architecture

A service interface to the controller subsystem is defined so that the controller may be considered a standard part. In this configuration, the controller operates the lowest three layers, and L2CAP is contained with the rest of the application in a host system. This standard interface is called the host controller interface (HCI), and its SAPs are represented by the ellipses on the upper edge of the controller subsystem in Figure 1. Implementation of this standard service interface is optional.

As the architecture is defined with the possibility of separate host and controller communicating through an HCI, a number of general assumptions are made. The controller is assumed to have limited data buffering capabilities in comparison with the host. Therefore, L2CAP is expected to carry out some simple resource management when submitting L2CAP protocol data units (PDUs) to the controller for transport to a peer device. This includes segmentation of L2CAP service data units (SDUs) into more manageable PDUs and then the fragmentation of PDUs into start and continuation packets of a size suitable for the controller buffers, and management of the use of controller buffers to ensure availability for channels with quality of service (QoS) commitments.

The BB protocol provides the basic ARQ Protocol. The L2CAP can optionally provide a further error detection and retransmission to the L2CAP PDUs. This feature is recommended for applications with requirements for a low probability of undetected errors in the user data. A further optional feature of L2CAP is a window-based flow control that can be used to manage buffer allocation in the receiving device. Both of these optional features augment the QoS performance in certain scenarios.

6.3 Core architectural blocks

This subclause describes the function and responsibility of each of the blocks shown in Figure 1, which describes a possible implementation architecture. An implementation is not required to follow the architecture described in this clause.

6.3.1 Channel manager

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

6.3.2 L2CAP resource manager

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

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

6.3.3 Device manager

The device manager is the functional block in the BB that controls the general behavior of the device. It is responsible for all operation of the system that is not directly related to data transport. This includes functions such as inquiring for the presence of other nearby devices, connecting to other devices, or making the device discoverable or connectable by other devices.

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

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

6.3.4 Link manager (LM)

The LM is responsible for the creation, modification, and release of logical links (and, if required, their associated logical transports) as well as the update of parameters related to physical links between devices. The LM achieves this by communicating with the LM in remote devices using the LMP.

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

6.3.5 BB resource manager

The BB resource manager is responsible for all access to the PHY. It has two main functions. At its heart is a scheduler that grants time on the physical channels to all of the entities that have negotiated an access

contract. The other main function is to negotiate access contracts with these entities. An access contract is effectively a commitment to deliver a certain QoS that is required in order to provide a user application with an expected performance.

The access contract and scheduling function must take account of any behavior that requires use of the radio, e.g., the normal exchange of data between connected devices over logical links, the logical transports, the carrying out of inquiries, the making of connections, the state of being discoverable or connectable, or the taking of readings from unused carriers during the use of AFH mode.

In some cases, the scheduling of a logical link results in changing to a different physical channel from the one that was previously used. This may be, for example, due to involvement in scatternet, a periodic inquiry function, or page scanning. When the physical channels are not time-slot-aligned, then the resource manager also accounts for the realignment time between slots on the original physical channel and slots on the new physical channel. In some cases, the slots will be naturally aligned due to the same device clock being used as a reference for both physical channels.

6.3.6 Link controller

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

The link controller carries out the LCP signalling (in close conjunction with the scheduling function of the resource manager), which is used to communicate flow control and acknowledgment and retransmission request signals. The interpretation of these signals is a characteristic of the logical transport associated with the BB packet. Interpretation and control of the link control signalling is normally associated with the resource manager's scheduler.

6.3.7 Radio frequency (RF)

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

6.4 Data transport architecture

The data transport system follows a layered architecture. This explanation of the system describes the core transport layers up to and including L2CAP channels. All operational modes follow the same generic transport architecture, which is shown in Figure 2.

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

IEEE Std 802.15.1-2002 described the ACL and SCO links as physical links. With the addition of extended SCO (eSCO) and for future expansion, it is better to consider these as logical transport types, which more accurately encapsulates their purpose. However, they are not as independent as might be desired, due to the shared use of the logical transport address (LT_ADDR) between SCO and ACL. Hence, the architecture is incapable of representing these logical transports with a single transport layer. The additional logical transport layer goes some way toward describing this behavior.

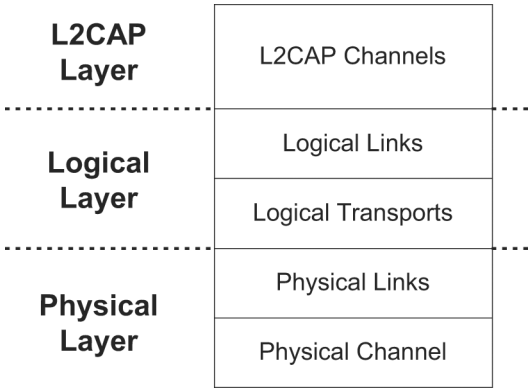


Figure 2—Generic data transport architecture

6.4.1 Core traffic bearers

The core system provides a number of standard traffic bearers for the transport of service protocol and application data. These are shown in Figure 3. For ease of representation, this is shown with higher layers to the left and lower layers to the right.

The core traffic bearers that are available to applications are shown in Figure 3 as the shaded rounded rectangles. The architectural layers that are defined to provide these services are described in 6.2. A number of data traffic types are shown on the left of the figure as linked to the traffic bearers that are typically suitable for transporting that type of data traffic.

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

The mapping of application traffic types to core traffic bearers in Figure 3 is based on matching the traffic characteristics with the bearer characteristics. It is recommended to use these mappings as they provide the most natural and efficient method of transporting the data with its given characteristics.

However, an application—or an implementation of the core system—may choose to use a different traffic bearer or a different mapping to achieve a similar result. For example, in a piconet with only one slave, the master may choose to transport L2CAP broadcasts over the ACL-U logical link rather than over the active slave broadcast user (ASB-U) or parked slave broadcast user (PSB-U) logical links. This will probably be more efficient in terms of bandwidth if the physical channel quality is not degraded. Use of alternative transport paths to those in Figure 3 is acceptable only if the characteristics of the application traffic type are preserved.

Figure 3 shows a number of application traffic types. These are used to classify the types of data that may be submitted to the core system. The original data traffic type may not be the same as the type that is submitted to the core system if an intervening process modifies it. For example, video data are generated at a constant rate, but an intermediate coding process may alter this to a variable rate, e.g., by MPEG4 encoding. For the purposes of the core system, only the characteristic of the submitted data is of interest.

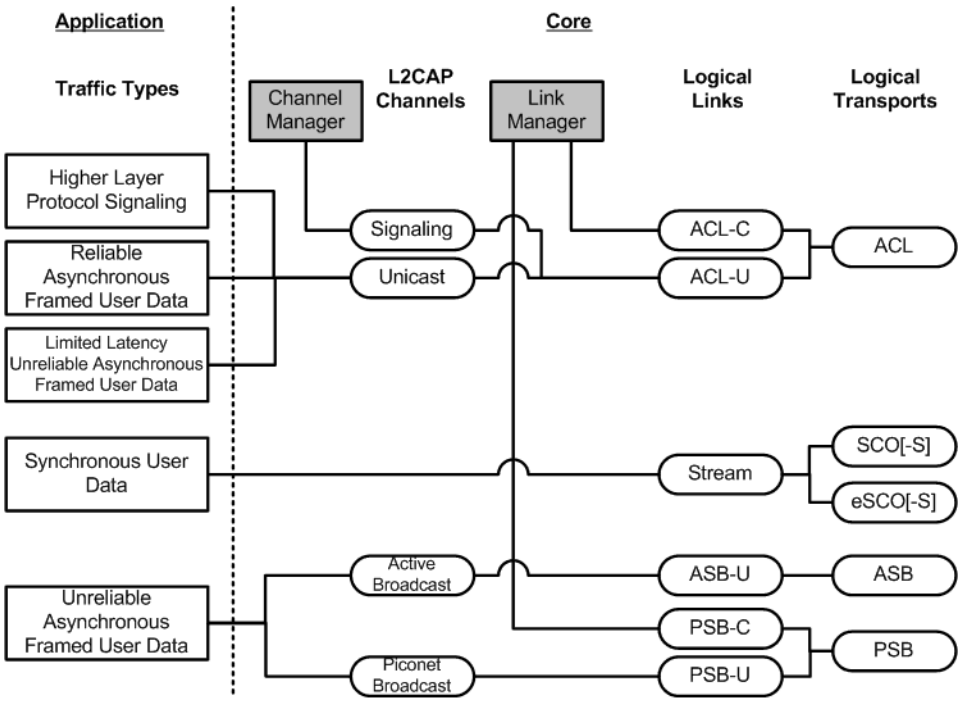


Figure 3—Traffic bearers

6.4.1.1 Framed data traffic

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

Connection-oriented L2CAP channels may be created for transport of unicast (point-to-point) data between two devices. A connectionless L2CAP channel exists for broadcasting data. In the case of piconet topologies, the master device is always the source of broadcast data, and the slave device(s) are the recipients. Traffic on the broadcast L2CAP channel is unidirectional. Unicast L2CAP channels may be unidirectional or bidirectional.

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

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

6.4.1.2 Unframed data traffic

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

The core system supports the direct transport of application data that are isochronous and of a constant bit rate, using a stream SCO (SCO-S) or stream extended SCO (eSCO-S) logical link. These logical links reserve physical channel bandwidth and provide a constant rate transport locked to the piconet clock. Data are transported in fixed size packets at fixed intervals with both of these parameters negotiated during channel establishment. eSCO links provide a greater choice of bit rates and also provide greater reliability by using limited retransmission in case of error. SCO and eSCO logical transports do not support multiplexed logical links or any further layering within the core. An application may choose to layer a number of streams within the submitted SCO/eSCO stream, provided that the submitted stream is, or has the appearance of being, a constant rate stream.

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

If the application data are of a variable rate (asynchronous), then they may be carried only by an L2CAP channel and hence will be treated as framed data.

6.4.1.3 Reliability of traffic bearers

This standard defines a wireless communications system. In high RF-noise environments, RF systems are inherently unreliable. To counteract this, the system provides levels of protection at each layer. The BB packet header uses forward error correction (FEC) coding to allow error correction by the receiver and a header error check (HEC) to detect errors remaining after correction. Certain BB packet types include FEC for the payload. Furthermore, some BB packet types include a cyclic redundancy check (CRC).

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

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

The L2CAP provides an additional level of error control that is designed to detect the occasional undetected errors in the BB protocol and request retransmission of the affected data. This provides the level of reliability required by typical IEEE 802.15.1-2005 applications.

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

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

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

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

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

6.4.2 Transport architecture entities

The transport architecture entities are shown in Figure 4 and are described from the lowest layer upward in the subsequent subclauses.

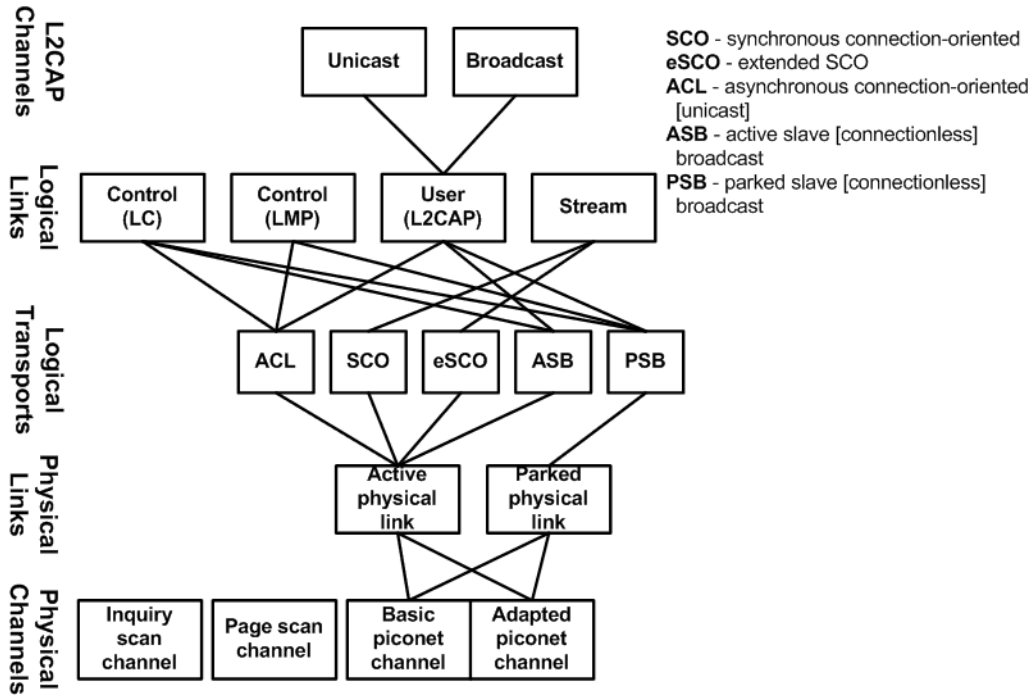


Figure 4—Overview of transport architecture entities and hierarchy

6.4.3 Generic packet structure

The general packet structure nearly reflects the architectural layers found in this standard. The packet structure is designed for optimal use in normal operation. It is shown in Figure 5.

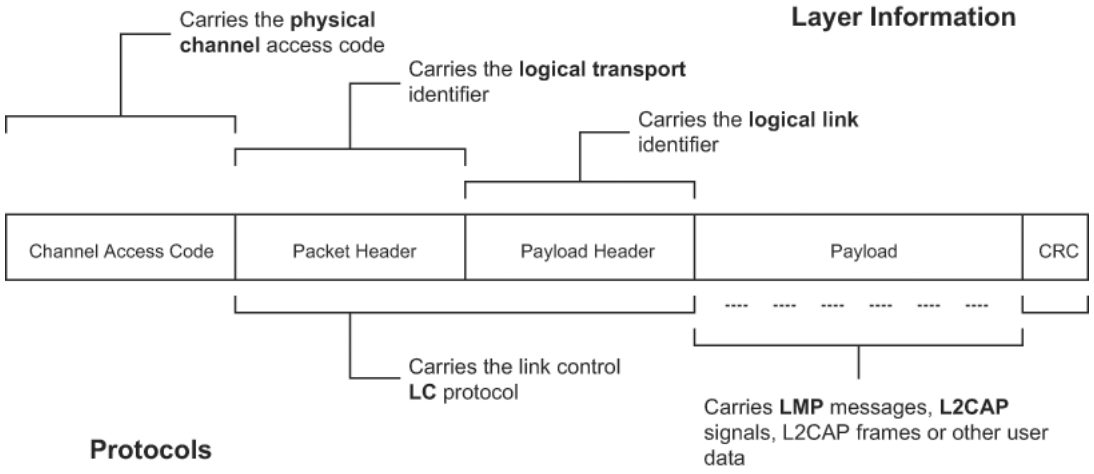


Figure 5—Packet structure

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

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

There is no direct field within the packet structure that represents or contains information relating to physical links. This information is implied in the LT_ADDR carried in the packet header.

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

The packet header also carries part of the LCP that is operated per logical transport (except for SCO logical transports, which are not affected by the LCP bits in the packet header).

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

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

6.4.4 Physical channels

The lowest architectural layer in the system is the physical channel. A number of types of physical channel are defined. All physical channels are characterized by an RF combined with temporal parameters and restricted by spatial considerations. All physical channel frequency hopping changes frequency periodically (frequency hop) to reduce the effects of interference and for regulatory reasons.

Two devices use a shared physical channel for communication. To achieve this, their transceivers need to be tuned to the same RF at the same time, and they need to be within a nominal range of each other.

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

Four physical channels are defined. Each is optimized and used for a different purpose. Two of these physical channels (i.e., the basic and adapted piconet physical channels) are used for communication between connected devices and are associated with a specific piconet. The remaining physical channels are used for discovering devices (i.e., the inquiry scan physical channel) and for connecting devices (i.e., the page scan physical channel).

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

Whenever a device is synchronized to the timing, frequency, and access code of a physical channel, it is said to be *connected* to this channel (whether or not it is actively involved in communications over the channel). This standard assumes that a device is capable of connecting to only one physical channel at any time. Advanced devices may be capable of connecting simultaneously to more than one physical channel, but this standard does not assume that this is possible.

6.4.4.1 Basic piconet physical channel

The basic piconet physical channel is used for communication between connected devices.

The basic piconet physical channel is characterized by a pseudo-random sequence hopping through the RF channels. The hopping sequence is unique for the piconet and is determined by the device address of the master. The phase in the hopping sequence is determined by the native clock (CLKN) of the master. All devices participating in the piconet are time- and hop-synchronized to the channel.

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

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

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

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

A special characteristic of the basic piconet physical channel is the use of some reserved slots to transmit a beacon train. The beacon train is used only if the piconet physical channel has parked slaves connected to it. In this situation, the master transmits a packet in the reserved beacon train slots. (These packets are used by the slave to resynchronize to the piconet physical channel.) The master may transmit packets from any logical transport in these slots, providing there is a transmission starting in each of the slots. In the case where there is information from the parked slave broadcast (PSB) logical transport to be transmitted, then this is transmitted in the beacon train slots and may take priority over any other logical transport.

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

There is, however, a limitation on the number of logical transports that can be supported within a piconet. This means that there is a limit to the number of these devices that can be actively involved in exchanging data with the master.

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

6.4.4.2 Adapted piconet physical channel

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

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

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

6.4.4.3 Inquiry scan physical channel

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

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

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

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

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

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

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

During the exchange of packets between an inquiring and discoverable device, it may be considered that a temporary physical link exists between these devices.

6.4.4.4 Page scan physical channel

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

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

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

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

The paging device may have some knowledge of the target device's CLKN (indicated during a previous inquiry transaction between the two devices or as a result of a previous involvement in a piconet with the device), in which case it is able to predict the phase of the target device's page scan physical channel. It may use this information to optimize the synchronization of the paging and page scanning process and speed up the formation of the connection.

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

During the exchange of packets between a paging and connectable device, a temporary physical link exists on the page scan physical channel between these devices.

A physical link represents a BB connection between devices. A physical link is always associated with exactly one physical channel (although a physical channel may support more than one physical link).

Within the system, a physical link is a virtual concept that has no direct representation within the structure of a transmitted packet. The Access Code Packet field, together with the clock and address of the master device are used to identify a physical channel. The physical link may be identified by association with the logical transport, as each logical transport is received only on one physical link.

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

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

6.4.5 Physical links

6.4.5.1 Links supported by basic and adapted piconet physical channels

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

6.4.5.1.1 Active physical link

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

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

Active physical links may be placed into HOLD or SNIFF mode. The effect of these modes is to modify the periods when the physical link is active and may carry traffic. Synchronous logical transports are not affected by these modes and continue according to their predefined scheduling behavior. The default ACL

logical transport and other links with undefined scheduling characteristics are subject to the mode of the active physical link.

6.4.5.1.2 Parked physical link

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

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

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

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

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

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

6.4.5.2 Links supported by scanning physical channels

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

6.4.6 Logical links and logical transports

A variety of logical links are available to support different application data transport requirements. Each logical link is associated with a logical transport, which has a number of characteristics. These characteristics include flow control, acknowledgment and repeat mechanisms, sequence numbering, and scheduling behavior. Logical transports are able to carry different types of logical links (depending on the type of the logical transport). In the case of some of the IEEE 802.15.1 logical links, these are multiplexed onto the same logical transport. Logical transports may be carried by active physical links on either the basic or the adapted piconet physical channel.

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

Table 3 lists all of the logical transport types, the supported logical link types, the types of physical links and physical channels that can support them, and a brief description of the purpose of the logical transport.

Table 3—Logical transport types

Logical transport	Links supported	Supported by	Overview
Asynchronous connection-oriented (ACL[a])	Control (LMP) ACL-C User (L2CAP) ACL-U	Active physical link, basic or adapted physical channel	Reliable or time-bounded, bidirectional, point-to-point.
Synchronous connection-oriented (SCO)	Stream (unframed) SCO-S	Active physical link, basic or adapted physical channel	Bidirectional, symmetric, point-to-point, audio-visual channels. Used for 64 kb/s constant rate data.
Extended synchronous connection-oriented (eSCO)	Stream (unframed) eSCO-S	Active physical link, basic or adapted physical channel	Bidirectional, symmetric or asymmetric, point-to-point, general regular data, limited retransmission. Used for constant rate data synchronized to the master's CLKN.
Active slave broadcast (ASB)	User (L2CAP) ASB-U	Active physical link, basic or adapted physical channel	Unreliable, unidirectional 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	Parked physical link, basic or adapted physical channel	Unreliable, unidirectional broadcast to all piconet devices. Used for LMP and L2CAP traffic to parked devices and for access requests from parked devices.

[a]It is clear that the most obvious abbreviation for asynchronous connection-oriented logical transport is ACO. However, this acronym has an alternative meaning in this standard. To avoid confusion between two possible meanings for ACO, the decision was made to retain the ACL abbreviation for the asynchronous connection-oriented logical transport.

The names given to the logical links and logical transports reflect some of the names used in IEEE Std 802.15.1-2002 in order to provide some degree of familiarity and continuation. However, these names do not reflect a consistent scheme, which is outlined later in this clause.

The classification of each link type follows from a selection procedure within three categories: casting (see 6.4.6.1), scheduling and acknowledgment scheme (see 6.4.6.2), and class of data (see 6.4.6.3).

6.4.6.1 Casting

The first category is that of casting. This may be either unicast or broadcast. There are no multicast links defined in this standard.

— *Unicast links.* Unicast links exist between exactly two endpoints. Traffic may be sent in either direction on unicast links. All unicast links are connection-oriented; a connection procedure takes place before the link may be used. In the case of the default ACL links, the connection procedure is an implicit step within the general paging procedure used to form ad hoc piconets.
— *Broadcast links.* Broadcast links exist between one source device and one or more receiver devices. Traffic is unidirectional; it is sent only from the source devices to the receiver devices. Broadcast links are connectionless; there is no procedure to create these links, and data may be sent over them at any time. Broadcast links are unreliable, and there is no guarantee that the data will be received.

6.4.6.2 Scheduling and acknowledgment scheme

The second category relates to the scheduling and acknowledgment scheme of the link and implies the type of traffic that is supported by the link. These are synchronous, isochronous, or asynchronous. There are no specific isochronous links defined in IEEE 802.15.1, although the default ACL link can be configured to operate in this fashion.

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

6.4.6.3 Class of data

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

— *Control links.* Control links are used only for transporting LMP messages between two LMs. These links are invisible above the BB layer and cannot be directly instantiated, configured, or released by applications, other than by the use of the connection and disconnection services that have this effect implicitly. Control links are always multiplexed with an equivalent L2CAP link onto an ACL logical transport. Subject to the rules defining the ARQ scheme, the control link traffic always takes priority over the L2CAP link traffic.
— *L2CAP links.* L2CAP links are used to transport L2CAP PDUs, which may carry the L2CAP signalling channel (on the default ACL-U logical link only) or framed user data submitted to user-instantiated L2CAP channels. L2CAP frames submitted to the BB may be larger than the available BB packets. A LCP embedded within the LLID field preserves the frame-start and frame-continuation semantics when the frame is transmitted in a number of fragments to the receiver.
— *Stream links.* Stream links are used to transport user data with no inherent framing that should be preserved when delivering the data. Lost data may be replaced by padding at the receiver.

6.4.6.4 Asynchronous connection-oriented (ACL)

The ACL logical transport is used to carry LMP and L2CAP control signalling and best effort asynchronous user data. The ACL logical transport uses a simple 1-bit acknowledgment scheme to provide simple channel

reliability. Every active slave device within a piconet has one ACL logical transport to the piconet master, known as the default ACL.

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

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

The default ACL may be used for isochronous data transport by configuring it to automatically flush packets after the packets have expired.

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

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

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

6.4.6.5 Synchronous connection-oriented (SCO)

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

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

The reuse of the default ACL's LT_ADDR for SCO logical transports is due to legacy behavior from IEEE Std 802.15.1-2002. In this earlier version of Bluetooth, the LT_ADDR (then known as the *active member address*) was used to identify the piconet member associated with each transmission. This was not easily extensible for enabling more logical links; therefore, the purpose of this field was redefined for the new features. Some of IEEE Std 802.15.1-2002 features, however, do not cleanly fit into the more formally described architecture.

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

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

6.4.6.6 Extended synchronous connection-oriented (eSCO)

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

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

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

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

6.4.6.7 Active slave broadcast (ASB)

The ASB logical transport is used to transport L2CAP user traffic to all devices in the piconet that are currently connected to the physical channel that is used by the ASB. There is no acknowledgment protocol, and the traffic is unidirectional from the piconet master to the slaves. The ASB channel may be used for L2CAP group traffic (a legacy of IEEE Std 802.15.1-2002 and the Bluetooth 1.1 specification) and is never used for L2CAP connection-oriented channels, L2CAP control signalling, or LMP control signalling.

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

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

An ASB is implicitly created whenever a piconet exists, and there is always one ASB associated with each of the basic and adapted piconet physical channels that exist within the piconet. Because the basic and adapted piconet physical channels are mostly coincident, a slave device cannot distinguish which of the ASB channels is being used to transmit the packets. This adds to the general unreliability of the ASB channel (although it is, perhaps, no more unreliable than general missed packets).

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

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

6.4.6.8 Parked slave broadcast (PSB)

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

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

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

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

The user information phase is used for the master to send L2CAP packets that are destined for all piconet slaves. User information may occupy one or more BB packets. If the user information occupies a single packet, then the user information packet is repeated in each of the piconet physical channel beacon train slots.

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

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

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

6.4.6.9 Logical links

Some logical transports are capable of supporting different logical links, either concurrently multiplexed, or one of the choice. Within such logical transports, the logical link is identified by the LLID bits in the payload header of BB packets that carry a data payload. The logical links distinguish between a limited set of core protocols that are able to transmit and receive data on the logical transports. Not all of the logical transports are able to carry all of the logical links (the supported mapping is shown in Figure 3). In particular, the SCO and eSCO logical transports are able to carry only constant data rate streams, and these are uniquely identified by the LT_ADDR. Such logical transports use only packets that do not contain a payload header, as their length is known in advance, and no LLID is necessary.

6.4.6.10 ACL control logical link (ACL-C)

The ACL-C logical link is used to carry LMP signalling between devices in the piconet. The control link is carried only on the default ACL logical transport and on the PSB logical transport (in the control information phase). The ACL-C link is always given priority over the ACL-U link when carried on the same logical transport.

6.4.6.11 Asynchronous/isochronous user logical link (ACL-U)

The ACL-U logical link is used to carry all asynchronous and isochronous framed user data. The ACL-U link is carried on all but the synchronous logical transports. Packets on the ACL-U link are identified by one of two reserved LLID values. One value is used to indicate whether the BB packet contains the start of an L2CAP frame and the other indicates a continuation of a previous frame. This ensures correct synchronization of the L2CAP reassembler following flushed packets. The use of this technique removes the need for a more complex L2CAP header in every BB packet (the header is required only in the L2CAP start packets), but adds the requirement that a complete L2CAP frame shall be transmitted before a new one is transmitted. (An exception to this rule being the ability to flush a partially transmitted L2CAP frame in favor of another L2CAP frame.)

6.4.6.12 SCO/eSCO streaming logical links (SCO-S/eSCO-S)

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

Variable rate isochronous data cannot be carried by the SCO-S or eSCO-S logical links. In this case, the data must be carried on ACL-U logical links, which use packets with a payload header. All versions to date of this standard have some limitations when supporting variable-rate isochronous data concurrently with reliable user data.

6.5 L2CAP channels

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

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

L2CAP channels may be configured to provide an appropriate QoS to the application. The L2CAP maps the channel onto the ACL-U logical link.

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

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

L2CAP can provide a per-channel flow control with the peer L2CAPs. This option is selected by the application when the channel is established. L2CAP can also provide enhanced error detection and retransmission to

— Reduce the probability of undetected errors being passed to the application
— Recover from loss of portions of the user data when the BB protocol performs a flush on the ACL-U logical link

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

6.6 Communication topology

Any time an IEEE 802.15.1-2005 link is formed, it is within the context of a piconet. A piconet consists of two or more devices that occupy the same physical channel (in other words, they are synchronized to a common clock and hopping sequence). The common (piconet) clock is identical to the CLKN of one of the devices in the piconet, known as the master of the piconet; and the hopping sequence is derived from the master's CLKN and the master's device address. All other synchronized devices are referred to as slaves in the piconet. The terms *master* and *slave* are used only when describing these roles in a piconet.

6.6.1 Piconet topology

Within a common location, a number of independent piconets may exist. Each piconet has a different physical channel (i.e., a different master device and an independent piconet clock and hopping sequence).

A device may participate concurrently in two or more piconets. It does this on a TDM basis. A device can never be a master of more than one piconet. (Since the piconet is defined by synchronization to the master's CLKN, it is impossible to be the master of two or more piconets.) A device may be a slave in many independent piconets.

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

In Figure 6, an example of topology is shown that demonstrates a number of the architectural features described below. Device A is a master in a piconet (represented by the shaded area and known as piconet A) with devices B, C, D and E as slaves. Two other piconets are shown:

— With device F as master (known as piconet F) and devices E, G, and H as slaves
— With device D as master (known as piconet D) and device J as slave

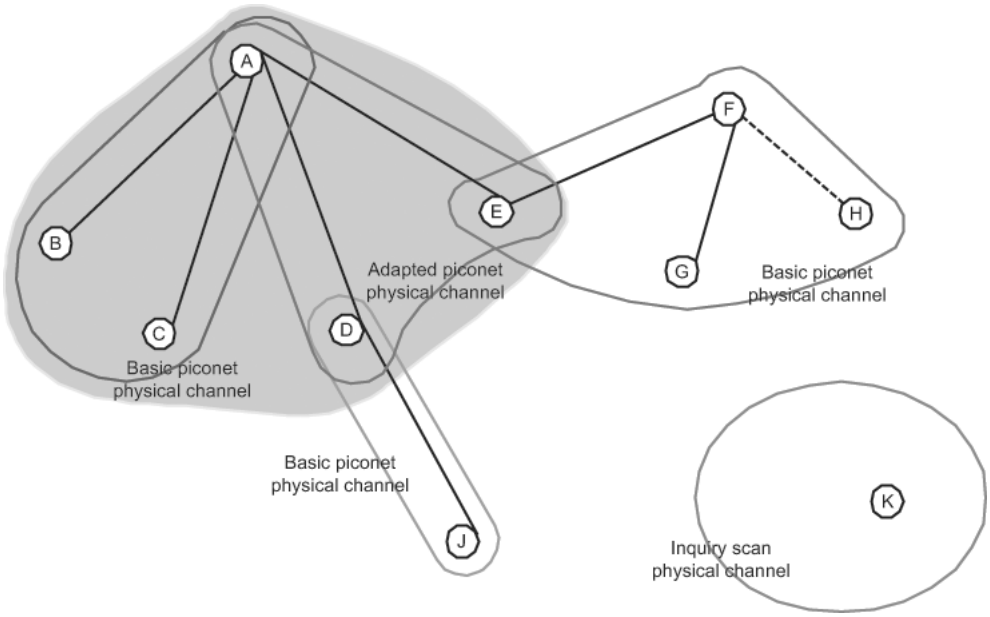


Figure 6—Example of IEEE 802.15.1-2005 topology

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

Piconets D and F are both using only a basic piconet physical channel. In the case of piconet D, this is because device J does not support the adaptive hopping mode. Although device D supports adaptive hopping, it cannot use it in this piconet. In piconet F, device F does not support adaptive hopping; therefore, it cannot be used in this piconet.

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

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

Logical transports, logical links, and L2CAP channels are used to provide capabilities for the transport of data, but are not shown on this figure. They are described in more detail in 6.4.6 and 6.5.

6.6.2 Operational procedures and modes

The typical operational mode of a device is to be connected to other devices (in a piconet), which exchange data with that device. Since IEEE 802.15.1 is an ad hoc wireless communications technology, there are also a number of operational procedures that enable piconets to be formed so that the subsequent communications can take place. Procedures and modes are applied at different layers in the architecture; therefore, a device may be engaged in a number of these procedures and modes concurrently.

6.6.2.1 Inquiry (discovering) procedure

All IEEE 802.15.1 devices use the inquiry procedure to discover nearby devices or to be discovered by devices in their locality.

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

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

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

6.6.2.2 Paging (connecting) procedure

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

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

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

6.6.2.3 Connected mode

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

Additional logical links are created using the LM that exchanges LMP messages with the remote device to negotiate the creation and settings for these links. Default ACL-C and ACL-U logical links are always created during the connection process, and these are used for LMP messages and the L2CAP signalling channel, respectively.

It is noted that two default logical links are created when two units are initially connected. One of these links (ACL-C) transports the LMP control protocol and is invisible to the layers above the LM. The other link (ACL-U) transports the L2CAP signalling protocol and any multiplexed L2CAP best-effort channels. It is common to refer to a default ACL logical transport, which can be resolved by context, but typically refers to the default ACL-U logical link. Also note that these two logical links share a logical transport.

During the time that a slave device is actively connected to a piconet, there is always a default ACL logical transport between the slave and the master device. There are two methods of deleting the default ACL logical transport.

The first method is to detach the device from the piconet physical channel, at which time the entire hierarchy of L2CAP channels, logical links, and logical transports between the devices is deleted.

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

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

6.6.2.4 HOLD mode

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

6.6.2.5 SNIFF mode

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

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

6.6.2.6 PARK state

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

When the physical link to a slave device is parked, this means that there are restrictions on when the master and slave may communicate, defined by the PSB logical transport parameters. During times when the PSB

logical transport is inactive (or absent), then the devices may engage in activity on other physical channels or enter reduced power mode.

6.6.2.7 Role switch procedure

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

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

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

7. Physical layer (PHY)

7.1 Scope

IEEE 802.15.1-2005 devices operate in the unlicensed 2.4 GHz ISM band. A frequency hop transceiver is applied to combat interference and fading. A shaped, binary FM is applied to minimize transceiver complexity. The symbol rate is 1 Msymbol/s. For full duplex transmission, a TDD scheme is used. This clause defines the requirements for an IEEE 802.15.1-2005 radio.

Requirements are defined for two reasons:

— To provide compatibility between radios used in the system
— To define the quality of the system

The radio shall fulfill the stated requirements under the operating conditions specified in 7.5, 7.6, and 7.7.

7.1.1 Regional authorities

This standard is based on the established regulations for Europe, Japan, and North America. The standard documents listed in 7.1.1.1, 7.1.1.2, and 7.1.1.3 are only for information and are subject to change or revision at any time.

7.1.1.1 Europe

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

7.1.1.2 Japan

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

7.1.1.3 North America

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

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

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

7.1.2 Frequency bands and channel arrangement

IEEE 802.15.1 operates in the 2.4 GHz ISM band. This frequency band is from 2400 MHz to 2483.5 MHz.

RF channels are spaced 1 MHz and are ordered in channel number k as shown in Table 4. In order to comply with out-of-band regulations in each country, a guard band is used at the lower and upper band edge. See Table 5. Because the RF channels are 1 MHz wide, centered on the RF channels defined in Table 4, the operating range of IEEE 802.15.1 radios is from 2401.5 MHz to 2480.5 MHz.

Table 4—Operating frequency bands

Regulatory range	RF channels
2.400–2.4835 GHz	f = 2402 + k MHz, k = 0,…,78

Table 5—Guard bands

Lower guard band	Upper guard band
1.5 MHz	3 MHz

7.2 Transmitter characteristics

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

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

If transmitting antennas of directional gain greater than 0 dBi are used, the applicable paragraphs in EN 300 328, EN 301 489-17 and FCC Part 15 shall be compensated for. The device is classified into three power classes. In Table 6, P_{min} is the minimum output power at maximum power setting.

Table 6—Power classes

Power class	Maximum output power (P_{max})	Nominal output power	Minimum output power (P_{min})	Power control
1	100 mW (20 dBm)	N/A	1 mW (0 dBm)	P_{min}<+4 dBm to P_{max} Optional: P_{min}[a] to P_{max}
2	2.5 mW (4 dBm)	1 mW (0 dBm)	0.25 mW (–6 dBm)	Optional: P_{min}[a] to P_{max}
3	1 mW (0 dBm)	N/A	N/A	Optional: P_{min}[a] to P_{max}

[a]The lower power limit P_{min}<–30dBm is suggested, but is not mandatory, and may be chosen according to application needs.

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

Devices with power control capability optimize the output power in a physical link with LMP commands (see Clause 9). This is done by each device checking its received signal strength indication (RSSI) and reporting to the peer if the power is above or below a desired golden range. On receiving such a report, a device that implements power control shall adjust its transmit power unless such an adjustment would take

the power above the device's maximum power or the maximum power for the device's power class or take the power below the device's minimum power.

In a connection, the output power shall not exceed the maximum output power of power class 2 for transmitting packets if the receiving device does not support the necessary messaging for sending the power control messages (see 9.3.1.3). In this case, the transmitting device shall comply with the rules of a class 2 or class 3 device.

If a class 1 device is paging or inquiring very close to another device, the input power can be larger than the requirement in 7.4.5. This can cause the receiving device to fail to respond. It may, therefore, be useful to page at power class 2 or 3 in addition to paging at power class 1.

7.3 Modulation characteristics

The modulation is Gaussian frequency shift keying (GFSK) (see Figure 7) with a bandwidth-bit period product, known as bandwidth time (BT), of 0.5. The modulation index shall be between 0.28 and 0.35. A binary one shall be represented by a positive frequency deviation, and a binary zero shall be represented by a negative frequency deviation. The symbol timing shall be less than ± 20 ppm.

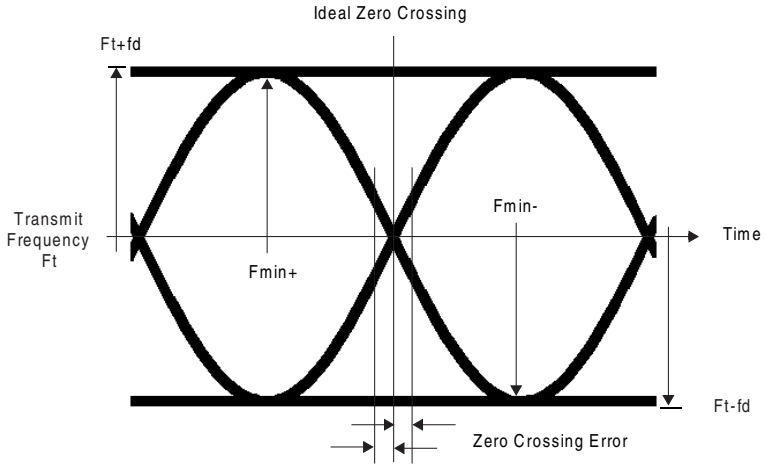


Figure 7—GFSK parameters definition

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

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

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

7.3.1 Spurious emissions

In-band spurious emissions shall be measured with a frequency hopping radio transmitting on one RF channel and receiving on a second RF channel; this means that the synthesizer may change RF channels between reception and transmission, but always returns to the same transmit RF channel. There will be no reference

in this standard to out-of-ISM-band spurious emissions; the equipment manufacturer is responsible for compliance in the intended country of use.

7.3.1.1 In-band spurious emission

Within the ISM band, the transmitter shall pass a spectrum mask, given in Table 7.[8] The spectrum shall comply with the 20 dB bandwidth definition in FCC Part 15.247 and shall be measured accordingly. In addition to the FCC requirement, an adjacent channel power on adjacent channels with a difference in RF channel number of two or greater is defined. This adjacent channel power is defined as the sum of the measured power in a 1 MHz RF channel. The transmitted power shall be measured in a 100 kHz bandwidth using maximum hold. The device shall transmit on RF channel M, and the adjacent channel power shall be measured on RF channel number N. The transmitter shall transmit a pseudo-random data pattern in the payload throughout the test.

Table 7—Transmit spectrum mask

Frequency offset	Transmit power
± 500 kHz	–20 dBc
2MHz (|M – N| = 2)	–20 dBm
3MHz or greater (|M – N| ≥ 3)	–40 dBm
NOTE—If the output power is less than 0 dBm, then, wherever appropriate, the FCC's 20 dB relative requirement overrules the absolute adjacent channel power requirement stated in this table.	

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

7.3.1.2 RF tolerance

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

The limits on the transmitter center frequency drift within a packet are specified in Table 8. The different packets are defined in Clause 8.

Table 8—Maximum allowable frequency drifts in a packet

Duration of packet	Frequency drift
Maximum length 1-slot packet	± 25 kHz
Maximum length 3-slot packet	± 40 kHz
Maximum length 5-slot packet	± 40 kHz
Maximum drift rate[a]	400 Hz/μs

[a]The maximum drift rate is allowed anywhere in a packet.

[8]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement this standard.

7.4 Receiver characteristics

The reference sensitivity level is –70 dBm.

7.4.1 Actual sensitivity level

The actual sensitivity level is defined as the input level for which a raw bit error rate (BER) of 0.1% is met. The receiver sensitivity shall be below or equal to –70 dBm with any transmitter compliant to the transmitter specified in 7.2.

7.4.2 Interference performance

The interference performance on co-channel and adjacent 1 MHz and 2 MHz shall be measured with the wanted signal 10 dB over the reference sensitivity level. For interference performance on all other RF channels, the wanted signal shall be 3 dB over the reference sensitivity level. If the frequency of an interfering signal is outside of the 2400–2483.5 MHz band, the out-of-band blocking specification (see 7.4.3) shall apply. The interfering signal shall be modulated as specified in 7.4.7. The BER shall be ≤0.1% for all the signal-to-interference ratios listed in Table 9.

Table 9—Interference performance

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

[a]In-band image frequency.

[b]If the image frequency ≠ n*1 MHz, then the image reference frequency is defined as the closest n*1 MHz frequency.

If two adjacent channel specifications from Table 9 are applicable to the same channel, the more relaxed specification applies.

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

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

7.4.3 Out-of-band blocking

The out-of-band suppression (or rejection) shall be measured with the wanted signal 3 dB over the reference sensitivity level. The interfering signal shall be a continuous wave signal. The BER shall be ≤0.1%. The out-of-band blocking shall fulfill the requirements in Table 10.

Table 10—Out-of-band suppression (or rejection) requirements

Interfering signal frequency	Interfering signal power level
30 – 2000 MHz	–10 dBm
2000 – 2399 MHz	–27 dBm
2484 – 3000 MHz	–27 dBm
3000 MHz – 12.75 GHz	–10 dBm

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

7.4.4 Intermodulation characteristics

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

— The wanted signal shall be at frequency f_0 with a power level 6 dB over the reference sensitivity level.
— A static sine wave signal shall be at frequency f_1 with a power level of –39 dBm.
— A modulated signal (see 7.4.7) shall be at frequency f_2 with a power level of –39 dBm.

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

7.4.5 Maximum usable level

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

7.4.6 Receiver signal strength indicator

If a device supports RSSI, the accuracy shall be ± 6 dBm.

7.4.7 Reference signal definition

A modulated interfering signal shall be defined as follows:

Modulation = GFSK
Modulation index = 0.32 ± 1%
BT = 0.5 ± 1%
Bit rate = 1 Mbps ± 1 ppm
Modulating data for wanted signal = pseudo-random bit sequence 9 (PRBS9)
Modulating data for interfering signal = pseudo-random bit sequence 15 (PRBS15)
Frequency accuracy better than ± 1 ppm

7.5 Nominal test conditions

7.5.1 Nominal temperature

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

7.5.2 Nominal power source

7.5.2.1 Mains voltage

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

7.5.2.2 Lead-acid battery power sources used in vehicles

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

7.5.2.3 Other power sources

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

7.6 Extreme test conditions

7.6.1 Extreme temperatures

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

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

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

7.6.2 Extreme power source voltages

Tests at extreme power source voltages specified in 7.6.2.1 through 7.6.2.4 are not required when the equipment under test is designed for operation as part of, and powered by, another system or piece of equipment. Where this is the case, the limit values of the host system or host equipment shall apply. The appropriate limit values shall be declared by the manufacturer and recorded in the test report.

7.6.2.1 Mains voltage

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

7.6.2.2 Lead-acid battery power source used on vehicles

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

7.6.2.3 Power sources using other types of batteries

The lower extreme test voltage for equipment with power sources using the indicated types of battery shall be as follows:

a) For Leclanché, alkaline, or lithium batteries: 0.85 times the nominal voltage of the battery
b) For mercury or nickel-cadmium batteries: 0.9 times the nominal voltage of the battery

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

7.6.2.4 Other power sources

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

7.7 Test condition parameters

The radio parameters shall be tested in the conditions shown in Table 11.

Table 11—Test conditions

Parameter	Temperature	Power source
Output power	ETC[a]	ETC
Power control	NTC[b]	NTC
Modulation index	ETC	ETC
Initial carrier frequency accuracy	ETC	ETC
Carrier frequency drift	ETC	ETC
Conducted in-band spurious emissions	ETC	ETC
Radiated in-band emissions	NTC	NTC
Sensitivity	ETC	ETC
Interference performance	NTC	NTC
Intermodulation characteristics	NTC	NTC
Out-of-band blocking	NTC	NTC
Maximum usable level	NTC	NTC
RSSI	NTC	NTC

[a]ETC = extreme test conditions.
[b]NTC = nominal test conditions.

8. Baseband (BB)

This clause describes the protocols and other lower level link routines that bridge between the PHY and upper level protocols.

8.1 General description

This standard provides a point-to-point connection or a point-to-multipoint connection (see a and b in Figure 8). In a point-to-point connection, the physical channel is shared between two IEEE 802.15.1-2005 devices. In a point-to-multipoint connection, the physical channel is shared among several devices. Two or more devices sharing the same physical channel form a *piconet*. One device acts as the master of the piconet, whereas the other device(s) act as slave(s). Up to seven slaves can be active in the piconet. Additionally, many more slaves can remain connected in PARK state. These parked slaves are not active on the channel, but remain synchronized to the master and can become active without using the connection establishment procedure. Both for active and parked slaves, the channel access is controlled by the master.

Piconets that have common devices are called a *scatternet* (see c in Figure 8). Each piconet has only a single master; however, slaves can participate in different piconets on a TDM basis. In addition, a master in one piconet can be a slave in other piconets. Piconets shall not be frequency synchronized, and each piconet has its own hopping sequence.

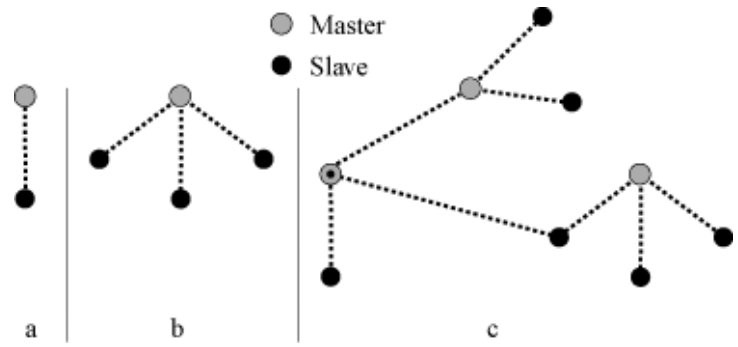


Figure 8—Piconets with a single slave operation (a), a multislave operation (b) and a scatternet operation (c)

Data are transmitted over the air in packets. The general packet format is shown in Figure 9. Each packet consists of 3 entities: the access code, the header, and the payload.

ACCESS CODE	HEADER	PAYLOAD

Figure 9—Standard packet format

8.1.1 Clock

Every IEEE 802.15.1-2005 device shall have a CLKN that shall be derived from a free-running system clock. For synchronization with other devices, offsets are used that, when added to the CLKN, provide temporary clocks that are mutually synchronized. It should be noted that the master clock (CLK) has no relation to the time of day; therefore, it may be initialized to any value. The clock has a cycle of about a day. If the

clock is implemented with a counter, a 28-bit counter is required that shall wrap around at $2^{28} - 1$. The least significant bit (LSB) shall tick in units of 312.5 μs (i.e., half a time slot), giving a clock rate of 3.2 kHz.

The clock determines critical periods and triggers the events in the device. Four periods are important in the IEEE 802.15.1-2005 system: 312.5 μs, 625 μs, 1.25 ms, and 1.28 s. These periods correspond to the timer bits CLK_0, CLK_1, CLK_2, and CLK_{12}, respectively (see Figure 10).

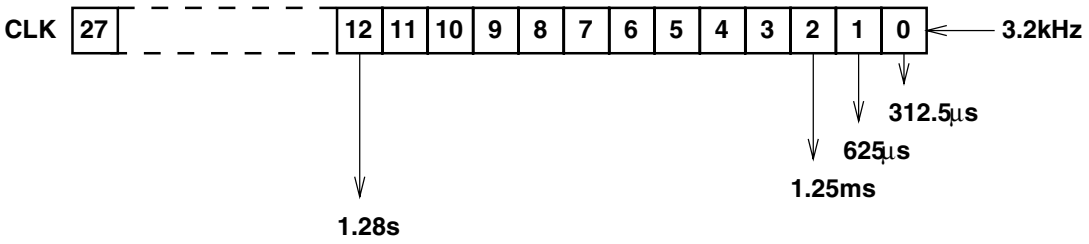


Figure 10—IEEE 802.15.1-2005 clock

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

— CLKN native clock
— CLKE estimated clock
— CLK master clock

CLKN is the native clock and shall be the reference to all other clock appearances. In STANDBY and PARK states and in HOLD and SNIFF modes, the CLKN may be driven by a low-power oscillator (LPO) with worst-case accuracy (± 250 ppm). Otherwise, the CLKN shall have a worst-case accuracy of ± 20 ppm.

See 8.2.2.4 for the definition of CLK and 8.2.4.1 for the definition of CLKE.

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

8.1.2 Device addressing

Each IEEE 802.15.1-2005 device shall be allocated a unique 48-bit device address (BD_ADDR). This address shall be obtained from the IEEE Registration Authority. The address is divided into the following three fields:

— LAP field: lower address part consisting of 24 bits
— UAP field: upper address part consisting of 8 bits
— NAP field: nonsignificant address part consisting of 16 bits

The LAP and UAP form the significant part of the BD_ADDR. The bit pattern in Figure 11 is an example of a BD_ADDR.

The BD_ADDR may take any values except the 64 reserved LAP values for general and dedicated inquiries (see 8.1.2.1).

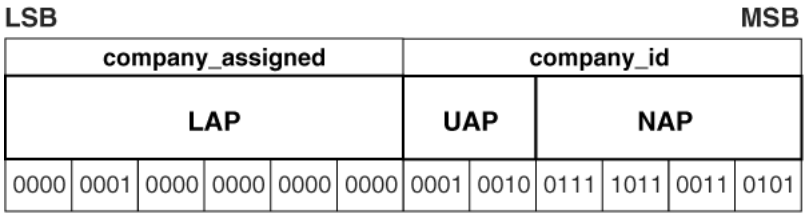


Figure 11—Format of BD_ADDR

8.1.2.1 Reserved addresses

A block of 64 contiguous LAPs is reserved for inquiry operations. One LAP common to all devices is reserved for general inquiry, and the remaining 63 LAPs are reserved for dedicated inquiry of specific classes of devices (see Bluetooth Assigned Numbers [B1]). The same LAP values are used regardless of the contents of UAP and NAP. Consequently, none of these LAPs can be part of a user BD_ADDR.

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

8.1.3 Access codes

In IEEE 802.15.1, all transmissions over the physical channel begin with an access code. Three different access codes are defined (see also 8.6.3.1):

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

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

8.2 Physical channels

The lowest architectural layer in the IEEE 802.15.1-2005 system is the physical channel. A number of types of physical channel are defined. All physical channels are characterized by the combination of a pseudo-random frequency hopping sequence, the specific slot timing of the transmissions, the access code, and packet header encoding. These aspects, together with the range of the transmitters, define the signature of the physical channel. For the basic and adapted piconet physical channels, frequency hopping is used to change frequency periodically to reduce the effects of interference and to satisfy local regulatory requirements.

Two devices that wish to communicate use a shared physical channel for this communication. To achieve this, their transceivers must be tuned to the same RF at the same time, and they must be within a nominal range of each other.

Given that the number of RF carriers is limited and that many IEEE 802.15.1-2005 devices may be operating independently within the same spatial and temporal area, there is a strong likelihood of two independent devices having their transceivers tuned to the same RF carrier, resulting in a physical channel collision. To mitigate the unwanted effects of this collision, each transmission on a physical channel starts with an access code that is used as a correlation code by devices tuned to the physical channel. This CAC is a property of the physical channel. The access code is always present at the start of every transmitted packet.

Four physical channels are defined. Each is optimized and used for a different purpose. Two of these physical channels (i.e., the basic and adapted piconet physical channels) are used for communication between connected devices and are associated with a specific piconet. The remaining physical channels are used for discovering (i.e., the inquiry scan physical channel) and connecting (i.e., the page scan physical channel) devices.

An IEEE 802.15.1-2005 device can use only one of these physical channels at any given time. In order to support multiple concurrent operations, the device uses TDM between the channels. In this way, a device can appear to operate simultaneously in several piconets as well as being discoverable and connectable.

Whenever an IEEE 802.15.1-2005 device is synchronized to the timing, frequency, and access code of a physical channel, it is said to be *connected* to this channel (whether or not it is actively involved in communications over the channel). At a minimum, a device need be capable of connection to only one physical channel at a time; however, advanced devices may be capable of connecting simultaneously to more than one physical channel, but this standard does not assume that this is possible.

8.2.1 Physical channel definition

Physical channels are defined by a pseudo-random RF channel hopping sequence, the packet (slot) timing, and an access code. The hopping sequence is determined by the UAP and LAP of a device address and the selected hopping sequence. The phase in the hopping sequence is determined by the CLK. 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 8.2.6). The maximum hop rate is 1600 hop/s in the CONNECTION state and the maximum is 3200 hop/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

8.2.2 Basic piconet physical channel

During the CONNECTION state, the basic piconet physical channel is used by default. The adapted piconet physical channel may also be used. The adapted piconet physical channel is identical to the basic piconet physical channel except for the differences listed in 8.2.3.

8.2.2.1 Master-slave definition

The basic piconet physical channel is defined by the master of the piconet. The master controls the traffic on the piconet physical channel by a polling scheme (see 8.8.5).

By definition, the device that initiates a connection by paging is the master. Once a piconet has been established, master-slave roles may be exchanged. This is described in 8.8.6.5.

8.2.2.2 Hopping characteristics

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

The basic piconet physical channel uses the basic channel hopping sequence, which is described in 8.2.6.

8.2.2.3 Time slots

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

A TDD scheme is used where master and slave alternatively transmit (see Figure 12). The packet start shall be aligned with the slot start. Packets may extend over up to five time slots.

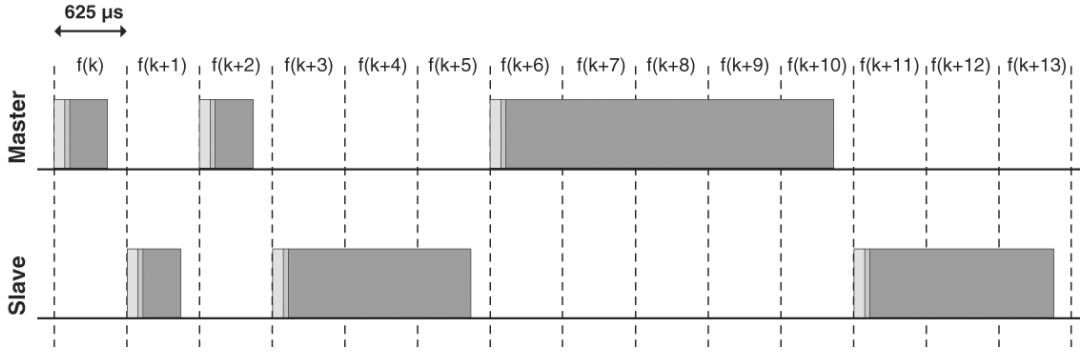


Figure 12—Multislot packets

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

8.2.2.4 Piconet clocks

CLK is the master clock of the piconet. It shall be used for all timing and scheduling activities in the piconet. All devices shall use the CLK to schedule their transmission and reception. The CLK shall be derived from the CLKN (see 8.1.1) by adding an offset (see Figure 13). The offset shall be zero for the master since CLK is identical to its own CLKN. Each slave shall add an appropriate offset to its CLKN so that the CLK

corresponds to the CLKN of the master. Although all CLKNs in the devices run at the same nominal rate, mutual drift causes inaccuracies in CLK. Therefore, the offsets in the slaves must be regularly updated so that CLK is approximately the CLKN of the master.

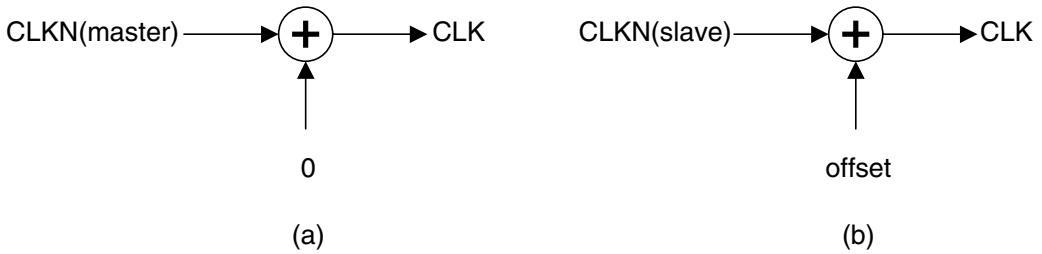


Figure 13—Derivation of CLK in master (a) and in slave (b)

8.2.2.5 Transmit/receive timing

The master transmission shall always start at even-numbered time slots ($CLK_1 = 0$), and the slave transmission shall always start at odd-numbered time slots ($CLK_1 = 1$). Due to packet types that cover more than a single slot, master transmission may continue in odd-numbered slots, and slave transmission may continue in even-numbered slots (see Figure 12).

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

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

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

where

T_N is the nominal slot length (625 μs);

j_k denotes jitter ($|j_k| \leq 1$ μs) at the start of slot k;

d_k denotes the drift ($|d_k| \leq 20$ ppm) within slot k.

The jitter and drift may vary arbitrarily within the given limits for every slot, while offset is an arbitrary, but fixed, constant. For HOLD mode, PARK state, and SNIFF mode, the drift and jitter parameters specified in 9.3.3.1 apply.

8.2.2.5.1 Piconet physical channel timing

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

The master TX/RX timing is shown in Figure 14. In Figure 14 and Figure 15, the channel hopping frequencies are indicated by $f(k)$ where k is the time slot number. After transmission, a return packet is expected

$N \times 625$ μs after the start of the TX packet where N is an odd integer larger than 0. N depends on the type of the transmitted packet.

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

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

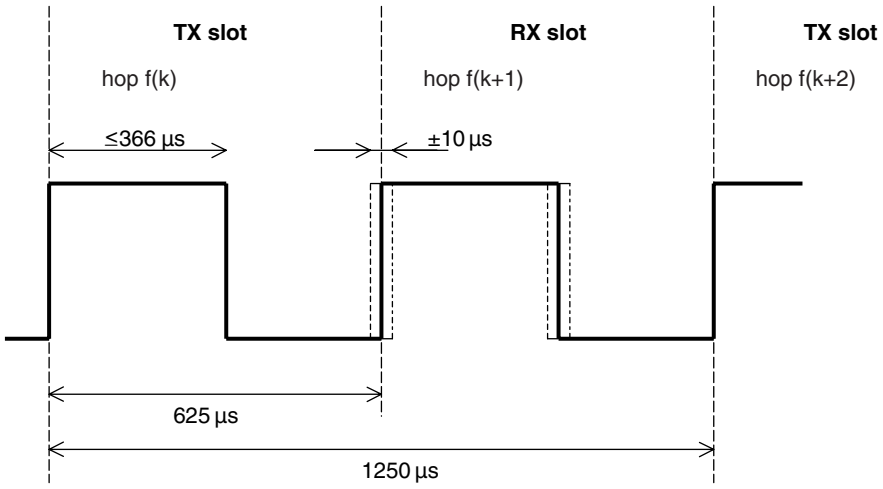


Figure 14—RX/TX cycle of master transceiver in normal mode for single-slot packets

Each master transmission shall be derived from bit 2 of the CLK; thus, the current transmission will be scheduled $M \times 1250$ μs after the start of the previous master TX burst where M depends on the transmitted and received packet type and is an even integer larger than 0. The master TX timing shall be derived from CLK, and thus it will not be affected by time drifts in the slave(s).

Slaves maintain an estimate of CLK by adding a timing offset to the slave's CLKN (see 8.2.2.4). This offset shall be updated each time a packet is received from the master. By comparing the exact RX timing of the received packet with the estimated RX timing, slaves shall correct the offset for any timing misalignments. Since only the CAC is required to synchronize the slave, slave timing can be corrected with any packet sent with the correct CAC.

The slave's TX/RX timing is shown in Figure 15. The slave's transmission shall be scheduled $N \times 625$ μs after the start of the slave's RX packet where N is an odd, positive integer larger than 0. If the slave's RX timing drifts, so will its TX timing. During periods when a slave is in the active mode (see 8.8.6) and is not able to receive any valid CACs from the master, the slave may increase its receive uncertainty window and/or use predicted timing drift to increase the probability of receiving the master's bursts when reception resumes.

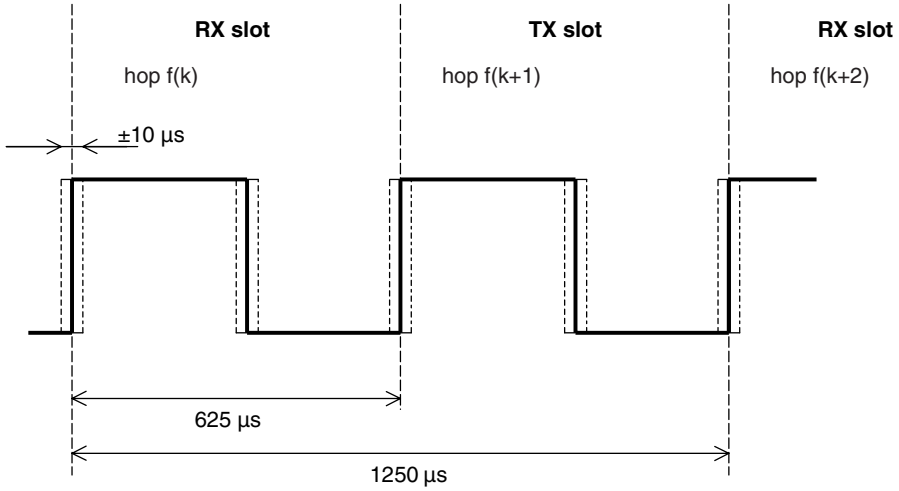


Figure 15—RX/TX cycle of slave transceiver for single-slot packets

8.2.2.5.2 Piconet physical channel resynchronization

In the piconet physical channel, a slave may lose synchronization if it does not receive a packet from the master at least every 250 ms (or less if the low-power clock is used). This may occur in SNIFF mode, HOLD mode , PARK state, or a scatternet or due to interference. When resynchronizing to the piconet physical channel, a slave device shall listen for the master before it may send information. In this case, the length of the search window in the slave device may be increased from 20 µs to a larger value *X* µs as illustrated in Figure 16. 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.

The slave is using AFH transmit on the master's transmit frequency and using the same CAC as the master. When a resynchronizing slave has a receive window of more than one-slot duration, it could falsely synchronize on a slave transmission instead of a master transmission. To avoid this, when receive windows are wider than a single slot, the slave may check that the whitening pattern in use corresponds to CLKE (i.e., the slave's estimate of the master's CLKN). As the whitening pattern changes with the master's clock value, this allows a slave to distinguish master transmissions even when a slave transmits on the master's transmit frequency.

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

It is recommended that single-slot packets be transmitted by the master during slave resynchronization.

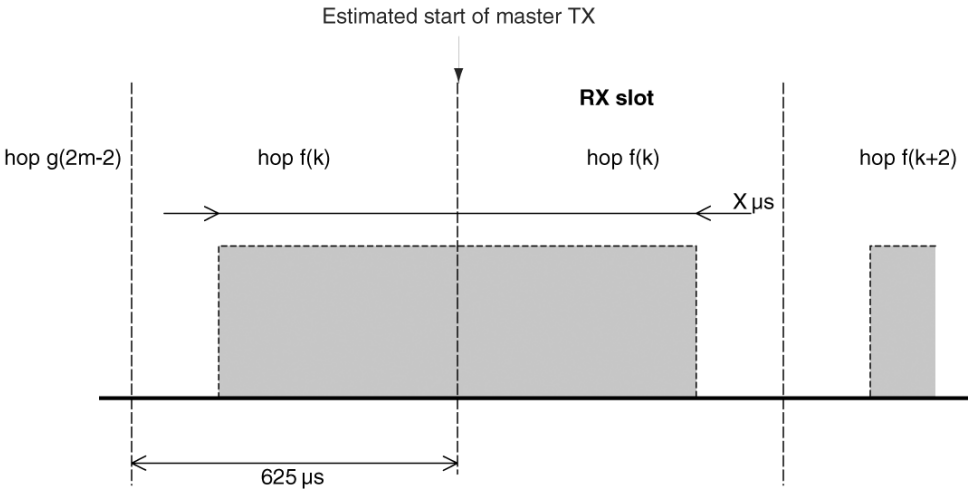


Figure 16—RX timing of slave returning from HOLD mode

8.2.3 Adapted piconet physical channel

8.2.3.1 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 8.2.6.

Adapted piconet physical channels can be used for connected devices that have AFH enabled. There are two differences between basic and adapted piconet physical channels:

— The slave uses the same frequency as the preceding master transmission when AFH is in effect.
— AFH uses less than the full 79 frequencies that the basic piconet uses.

8.2.4 Page scan physical channel

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

8.2.4.1 Clock estimate for paging

A paging device uses an estimate of the CLKN of the page scanning device, CLKE, i.e., an offset shall be added to the CLKN of the pager to approximate the CLKN of the recipient (see Figure 17). CLKE shall be derived from the reference CLKN by adding an offset. By using the CLKN of the recipient, the pager might be able to speed up the connection establishment.

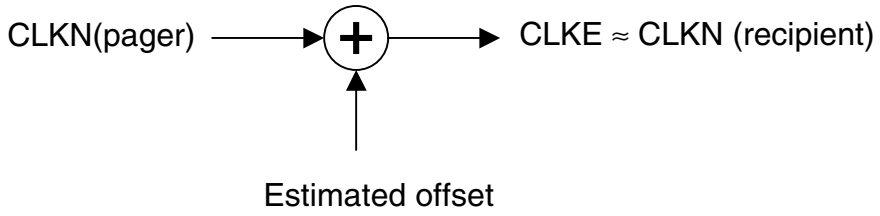


Figure 17—Derivation of CLKE

8.2.4.2 Hopping characteristics

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

The page scan physical channel uses the page, master page response, slave page response, and page scan hopping sequences specified in 8.2.6.

8.2.4.3 Paging procedure timing

During the paging procedure, the master shall transmit paging messages (see Table 26) corresponding to the slave to be connected. Since the paging message is a very short packet, the hop rate is 3200 hop/s. In a single TX slot interval, the paging device shall transmit on two different hop frequencies. In Figure 18 through Figure 22, $f(k)$ is used for the frequencies of the page hopping sequence and $f'(k)$ denotes the corresponding page response sequence frequencies. The first transmission starts where $CLK_0 = 0$, and the second transmission starts where $CLK_0 = 1$.

In a single RX slot interval, the paging device shall listen for the slave page response message on two different hop frequencies. Similar to transmission, the nominal reception starts where $CLK_0 = 0$, and the second reception nominally starts where $CLK_0 = 1$ (see Figure 18). During the TX slot, the paging device shall send the paging message at the TX hop frequencies $f(k)$ and $f(k + 1)$. In the RX slot, it shall listen for a response on the corresponding RX hop frequencies $f'(k)$ and $f'(k + 1)$. The listening periods shall be exactly timed 625 μs after the corresponding paging packets and shall include a ± 10 μs uncertainty window.

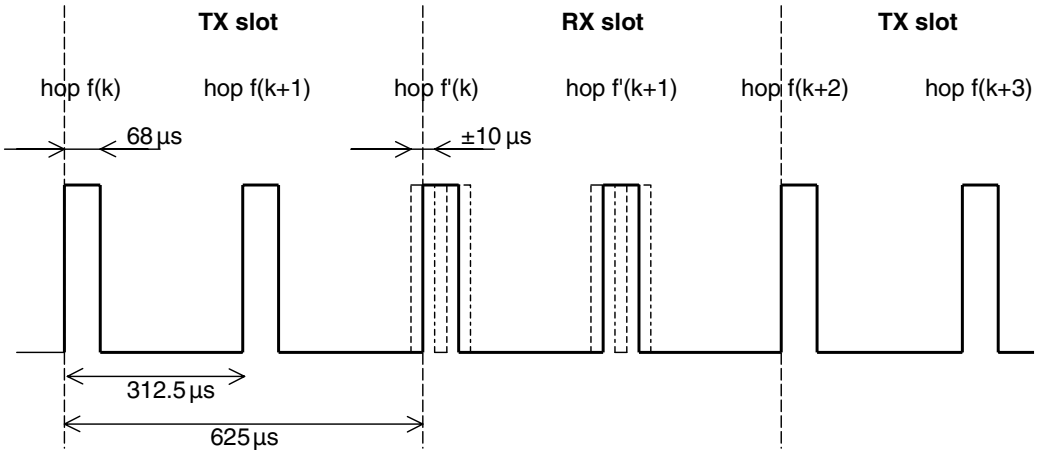


Figure 18—RX/TX cycle of transceiver in page mode

8.2.4.4 Page response timing

At connection setup, a master page response packet is transmitted from the master to the slave (see Table 26). This packet establishes the timing and frequency synchronization. After the slave device has received the page message, it shall return a response message that consists of the slave page response packet and shall follow 625 μs after the receipt of the page message. The master shall send the master page response packet in the TX slot following the RX slot in which it received the slave response, according to the RX/TX timing of the master. The time difference between the slave page response and master page response message will depend on the timing of the page message the slave received. In Figure 19, the slave receives the paging message sent **first** in the master-to-slave slot. It then responds with a first slave page response packet in the first half of the slave-to-master slot. The timing of the master page response packet is based on the timing of the page message sent first in the preceding master-to-slave slot: there is an exact 1250 μs delay between the first page message and the master page response packet. The packet is sent at the hop frequency $f(k+1)$, which is the hop frequency following the hop frequency $f(k)$ in which the page message was received.

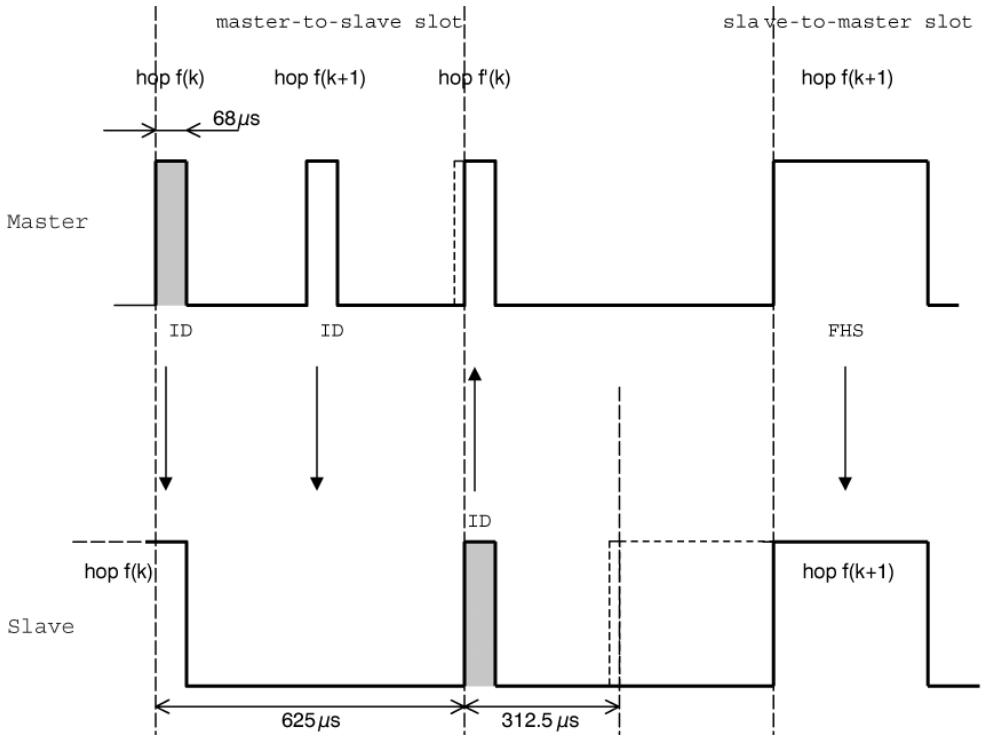


Figure 19—Timing of page response packets on successful page in first half slot

In Figure 20, the slave receives the paging message sent second in the master-to-slave slot. It then responds with a slave page response packet in the second half of the slave-to-master slot exactly 625 μs after the receipt of the page message. The timing of the master page response packet is still based on the timing of the page message sent **first** in the preceding master-to-slave slot: there is an exact 1250 μs delay between the **first** page message and the master page response packet. The packet is sent at the hop frequency $f(k+2)$, which is the hop frequency following the hop frequency $f(k+1)$ in which the page message was received.

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

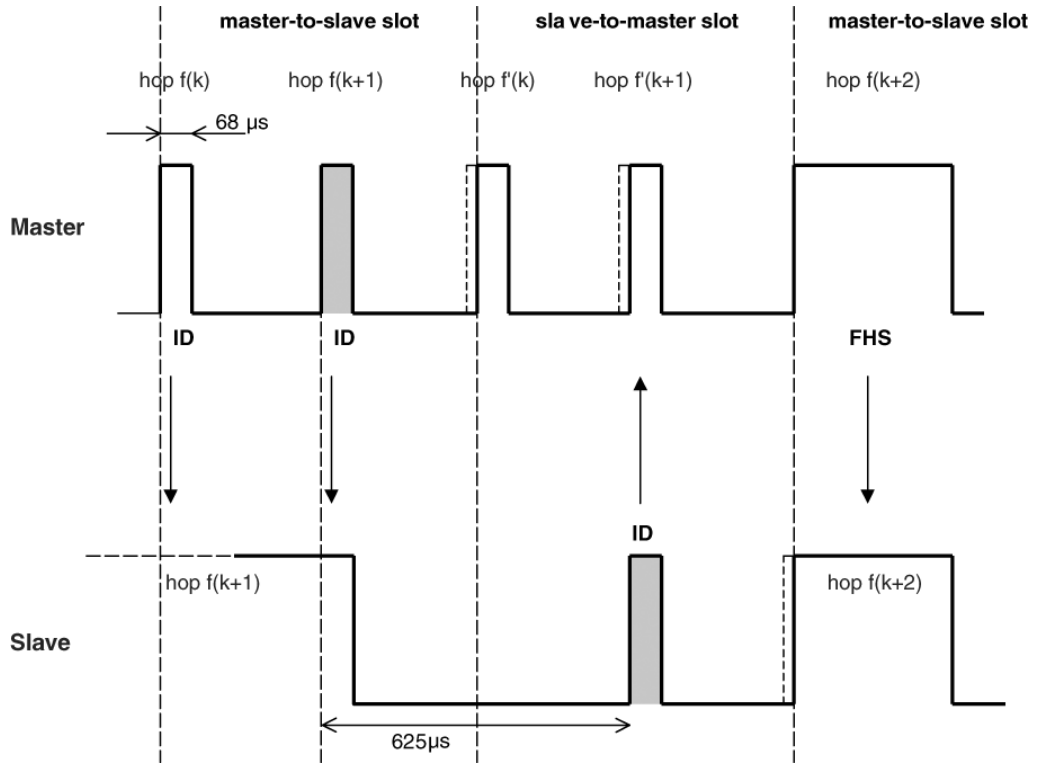


Figure 20—Timing of page response packets on successful page in second half slot

8.2.5 Inquiry scan physical channel

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

8.2.5.1 Clock for inquiry

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

8.2.5.2 Hopping characteristics

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

The inquiry scan physical channel uses the inquiry, inquiry response, and inquiry scan hopping sequences described in 8.2.6.

8.2.5.3 Inquiry procedure timing

During the inquiry procedure, the master shall transmit inquiry messages with the GIAC or DIAC. The timing for inquiry is the same as for paging (see 8.2.4.3).

8.2.5.4 Inquiry response timing

An inquiry response packet is transmitted from the slave to the master after the slave has received an inquiry message (see Table 28). This packet contains information necessary for the inquiring master to page the slave (see definition of the FHS packet in 8.6.5.1.4) and follows 625 µs after the receipt of the inquiry message. In Figure 21 and Figure 22, $f(k)$ is used for the frequencies of the inquiry hopping sequence and $f'(k)$ denotes the corresponding inquiry response sequence frequency. The packet is received by the master at the hop frequency $f'(k)$ when the inquiry message received by the slave was first in the master-to-slave slot.

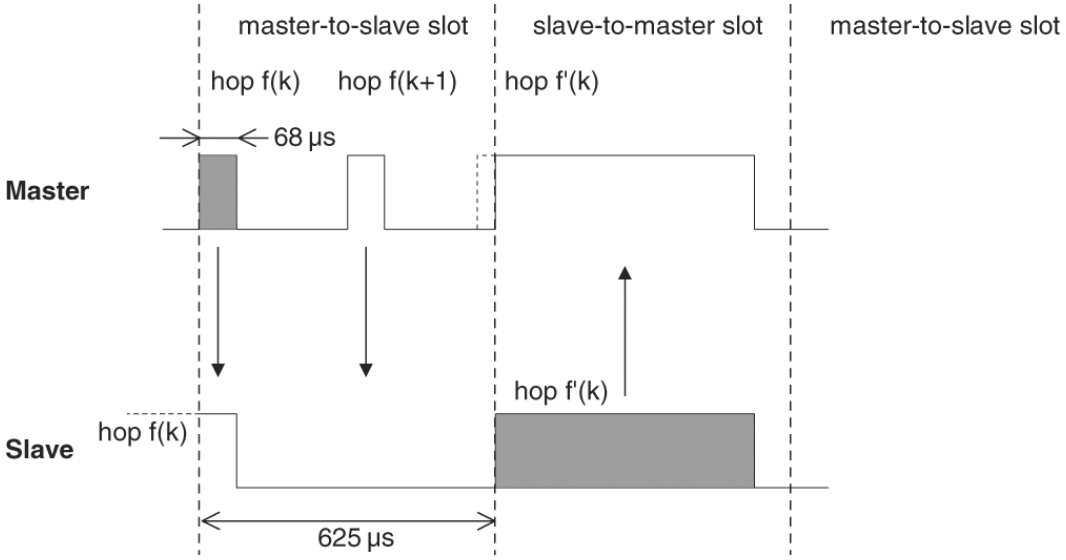


Figure 21—Timing of inquiry response packet on successful inquiry in first half slot

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

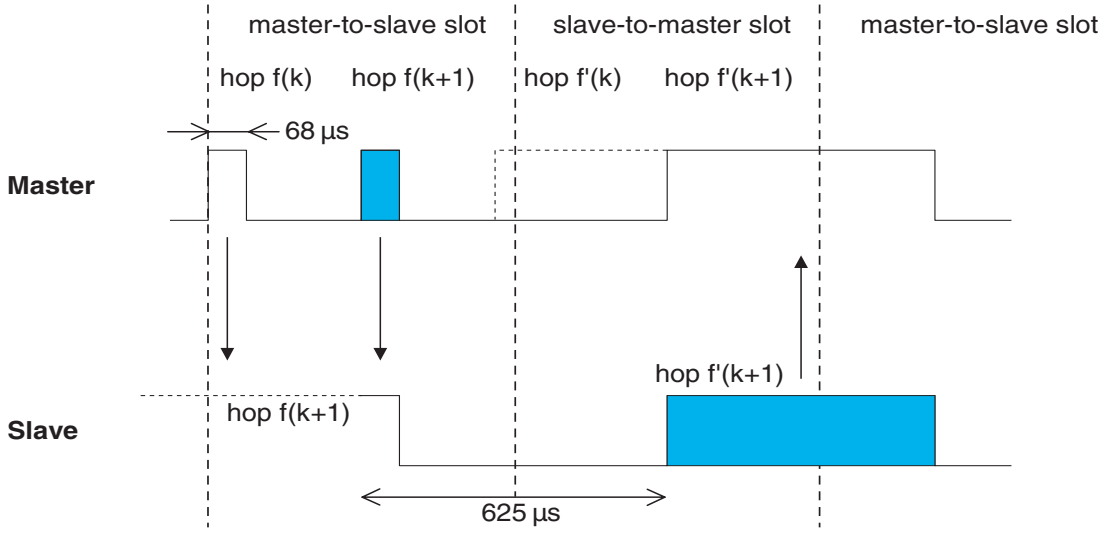


Figure 22—Timing of inquiry response packet on successful inquiry in second half slot

8.2.6 Hop selection

IEEE 802.15.1-2005 devices shall use the hopping kernel as defined in 8.2.6.2 and 8.2.6.3.

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 AFH. These sequences are as follows:

— 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 used only in place of the basic channel hopping sequence. All other hopping sequences are not affected by hop sequence adaptation.

8.2.6.1 General selection scheme

The selection scheme consists of two parts:

— Selecting a sequence
— Mapping this sequence onto the hop frequencies

The general block diagram of the hop selection scheme is shown in Figure 23. The mapping from the input to a particular RF channel index is performed in the selection box.

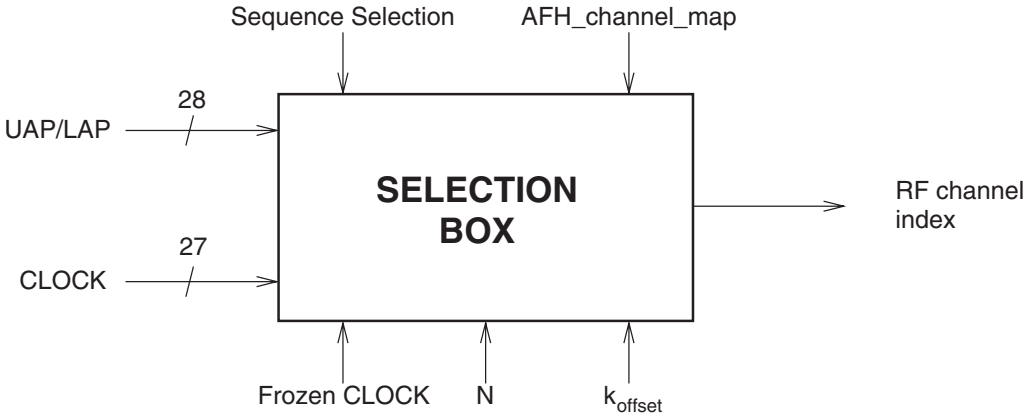


Figure 23—General block diagram of hop selection scheme

The inputs to the selection box are the selected clock, frozen clock, N, k_{offset}, address, sequence selection and AFH_channel_map. The source of the clock input depends on the hopping sequence selected. Additionally, each hopping sequence uses different bits of the clock (see Table 13). N and k_{offset} are defined in 8.2.6.4.

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

— Page scan
— Inquiry scan
— Page
— Inquiry
— Master page response
— Slave page response
— Inquiry response
— Basic channel
— Adapted channel

The address input consists of 28 bits including the entire LAP and the 4 LSBs of the UAP. This is designated as the UAP/LAP. When the basic or adapted channel hopping sequence is selected, the device address of the master (BD_ADDR) shall be used. When the page, master page response, slave page response, or page scan hopping sequences are selected, the BD_ADDR given by the host of the paged device shall be used (see 11.7.1.5). When the inquiry, inquiry response, or inquiry scan hopping sequences are selected, the UAP/LAP corresponding to the GIAC shall be used even if it concerns a DIAC. Whenever one of the reserved BD_ADDRs (see 8.1.2.1) is used for generating a frequency hop sequence, the UAP shall be replaced by the DCI (see 8.7.1). The hopping sequence is selected by the sequence selection input to the selection box.

When the adapted channel hopping sequence is selected, the AFH_channel_map is an additional input to the selection box. The AFH_channel_map indicates which channels shall be used and which shall be unused. These terms are defined in 8.2.6.3.

The output *RF channel index* constitutes a pseudo-random sequence. The RF channel index is mapped to RF channel frequencies using the equation in Table 4 (in Clause 7).

The selection scheme chooses a segment of 32 hop frequencies spanning about 64 MHz and visits these hops in a pseudo-random order. Next, a different 32-hop segment is chosen, etc. In the page, master page response, slave page response, page scan, inquiry, inquiry response, and inquiry scan hopping sequences, the same 32-hop segment is used all the time. (The segment is selected by the address; different devices will have different paging segments.) When the basic channel hopping sequence is selected, the output constitutes a pseudo-random sequence that slides through the 79 hops. The principle is depicted in Figure 24.

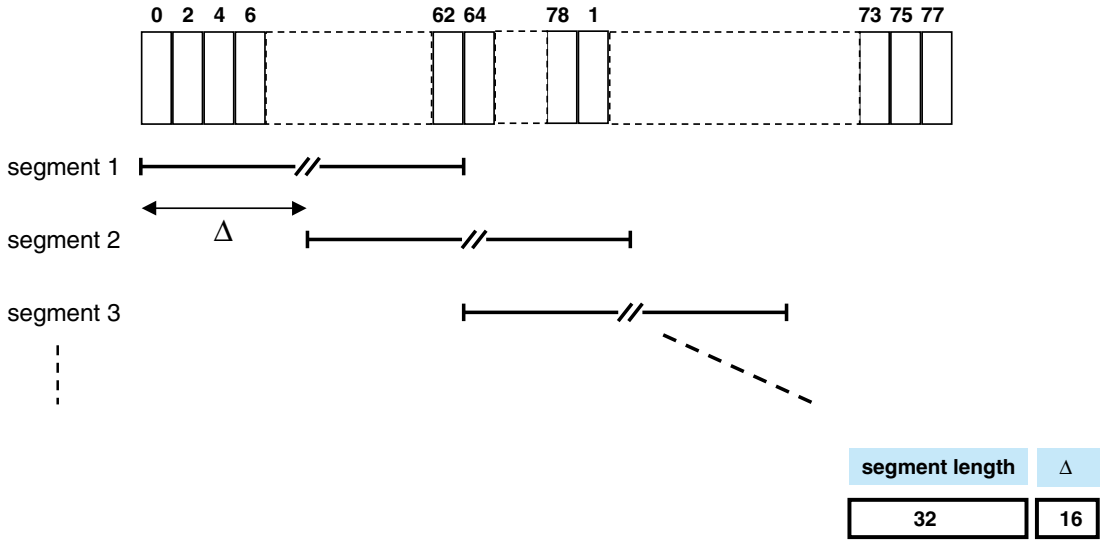


Figure 24—Hop selection scheme in CONNECTION state

The RF shall remain fixed for the duration of the packet. The RF for the packet shall be derived from the CLK value in the first slot of the packet. The RF in the first slot after a multislot packet shall use the frequency as determined by the CLK value for that slot. Figure 25 illustrates the hop definition on single-slot and multislot packets.

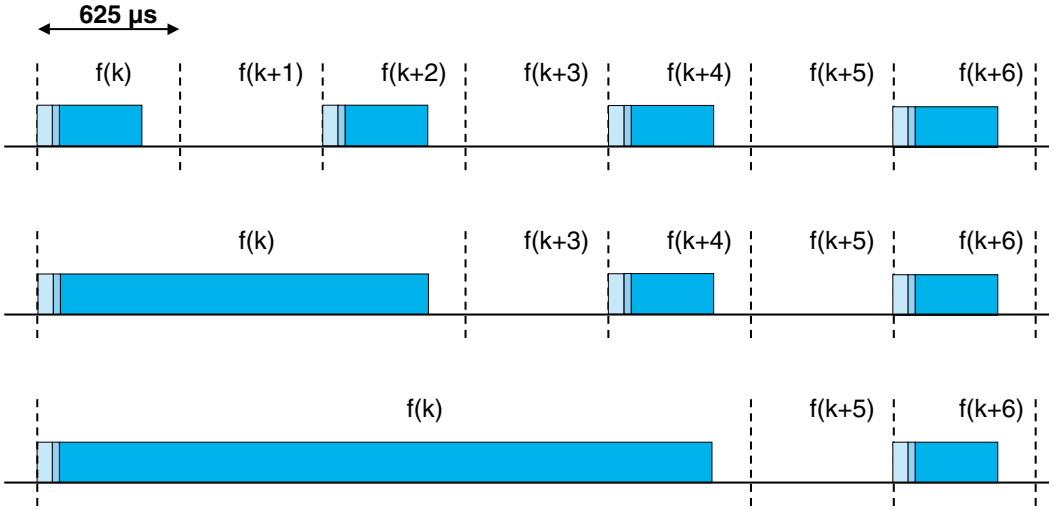


Figure 25—Single- and multislot packets

When the adapted channel hopping sequence is used, the pseudo-random sequence contains only frequencies that are in the RF channel set defined by the input AFH_channel_map. The adapted sequence has similar statistical properties to the nonadapted hop sequence (non-AHS). In addition, the slave responds with its packet on the same RF channel that was used by the master to address that slave (or would have been in the case of a synchronous reserved slot without a validly received master-to-slave transmission). This is called the *same channel mechanism* of AFH. Thus, the RF channel used for the master-to-slave packet is also used for the immediately following slave-to-master packet. An example of the same channel mechanism is illustrated in Figure 26. The same channel mechanism shall be used whenever the adapted channel hopping sequence is selected.

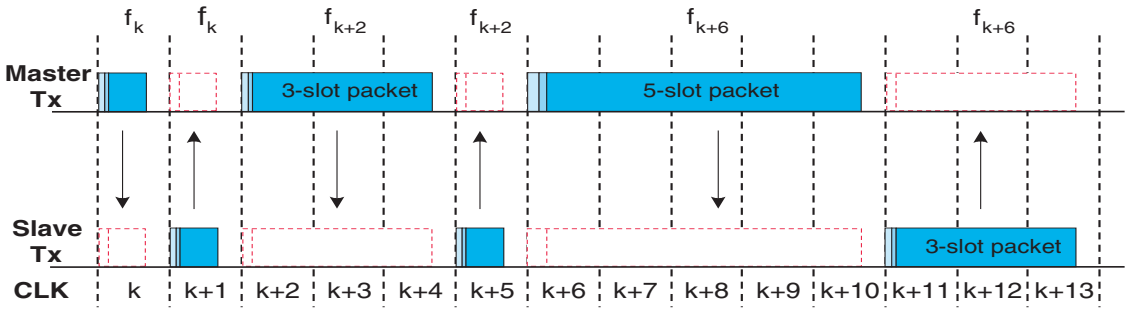


Figure 26—Example of the same channel mechanism

8.2.6.2 Basic hop selection kernel

The basic hop selection kernel shall be as shown in Figure 27 and is used for the page, page response, inquiry, inquiry response, and basic channel hopping selection kernels. In these substates, the input AFH_channel_map is unused. The adapted hop selection kernel is described in 8.2.6.3. The X input determines the phase in the 32-hop segment, whereas Y1 and Y2 select between master to slave and slave to master. Inputs A to D determine the ordering within the segment, inputs E and F determine the mapping onto

the hop frequencies. The kernel addresses a register containing the RF channel indices. This list is ordered so that first all even RF channel indices are listed and then all odd hop frequencies. In this way, a 32-hop segment spans about 64 MHz.

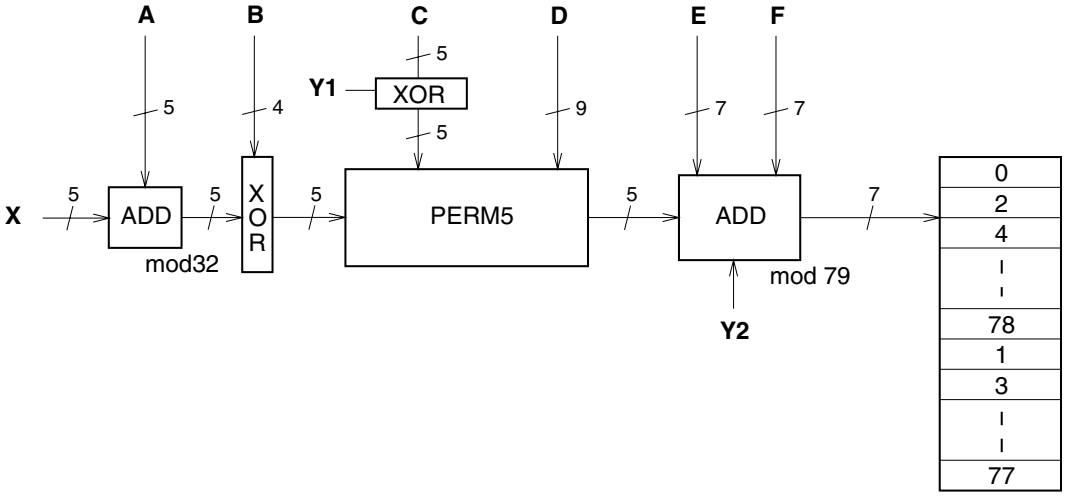


Figure 27—Block diagram of the basic hop selection kernel for the hop system

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

8.2.6.2.1 First addition operation

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

8.2.6.2.2 XOR operation

Let Z' denote the output of the first addition. In the XOR operation, the 4 LSBs of Z' are modulo-2 added to the address bits $A_{22–19}$. The operation is illustrated in Figure 28.

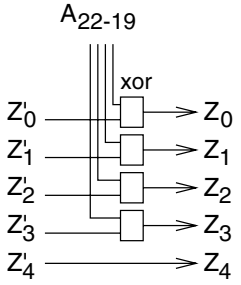


Figure 28—XOR operation for hop system

8.2.6.2.3 Permutation operation

The permutation operation involves the switching from five inputs to five outputs for the hop system, controlled by the control word. The permutation or switching box shall be as shown in Figure 29. It consists of seven stages of butterfly operations. The control of the butterflies by the control signals P is shown in Table 12. P_{0-8} corresponds to D_{0-8}, and, P_{i+9} corresponds to $C_i \oplus Y1$ for $i = 0...4$ in Figure 27.

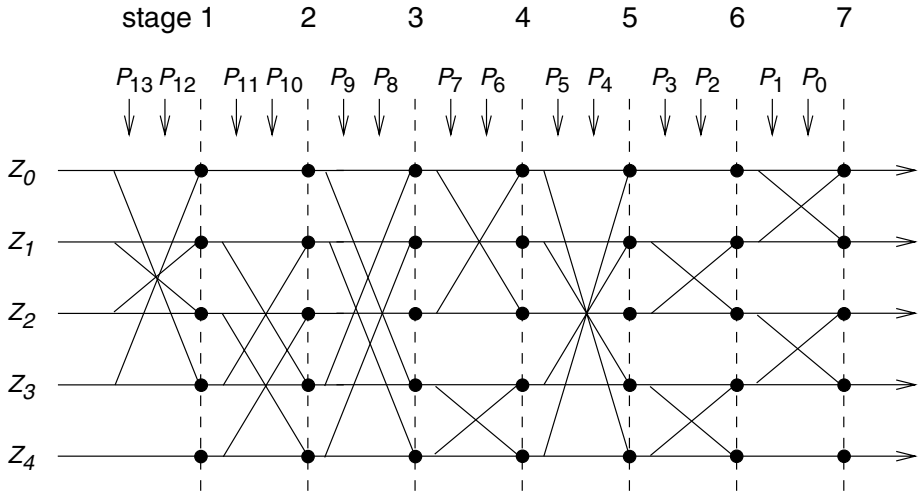


Figure 29—Permutation operation for hop system

Table 12—Control of butterflies for hop system

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

The Z input is the output of the XOR operation as described in 8.2.6.2.2. The butterfly operation can be implemented with multiplexers as depicted in Figure 30.

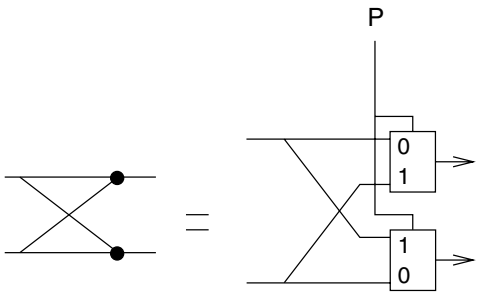


Figure 30—Butterfly implementation

8.2.6.2.4 Second addition operation

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

8.2.6.2.5 Register bank

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

8.2.6.3 Adapted hop selection kernel

The adapted hop selection kernel is based on the basic hop selection kernel defined in 8.2.6.2.

The inputs to the adapted hop selection kernel are the same as for the basic hop selection kernel except that the input AFH_channel_map (defined in 9.4.2) is used. The AFH_channel_map indicates which RF channels shall be used and which shall be unused. When hop sequence adaptation is enabled, the number of used RF channels may be reduced from 79 to some smaller value N. All devices shall be capable of operating on an adapted hop sequence (AHS) with $N_{min} \leq N \leq 79$, with any combination of used RF channels within the AFH_channel_map that meets this constraint. N_{min} is defined in 8.2.3.1.

Adaptation of the hopping sequence is achieved through two additions to the basic channel hopping sequence according to Figure 27:

— Unused RF channels are remapped uniformly onto used RF channels. In other words, if the hop selection kernel of the basic system generates an unused RF channel, an alternative RF channel out of the set of used RF channels is selected pseudo-randomly.
— The used RF channel generated for the master-to-slave packet is also used for the immediately following slave-to-master packet (see 8.2.6.1).

8.2.6.3.1 Channel remapping function

When the adapted hop selection kernel is selected, the basic hop selection kernel according to Figure 27 is initially used to determine an RF channel. If this RF channel is unused according to the AFH_channel_map, the unused RF channel is remapped by the remapping function to one of the used RF channels. If the RF channel determined by the basic hop selection kernel is already in the set of used RF channels, no adjustment is made. The hop sequence of the (nonadapted) basic hop equals the sequence of the adapted selection kernel on all locations where used RF channels are generated by the basic hop. This property facilitates non-AFH slaves remaining synchronized while other slaves in the piconet are using the adapted hopping sequence, although it is possible for a resynchronizing slave to mistake the transmissions of another slave that is using AFH for the transmissions of the master (see 8.2.2.5.2).

A block diagram of the remapping mechanism is shown in Figure 31. The remapping function is a post-processing step to the selection kernel from Figure 27, denoted as "Hop selection of basic hop system." The output f_k of the basic hop selection kernel is an RF channel number that ranges between 0 and 78. This RF channel will be either in the set of used RF channels or in the set of unused RF channels.

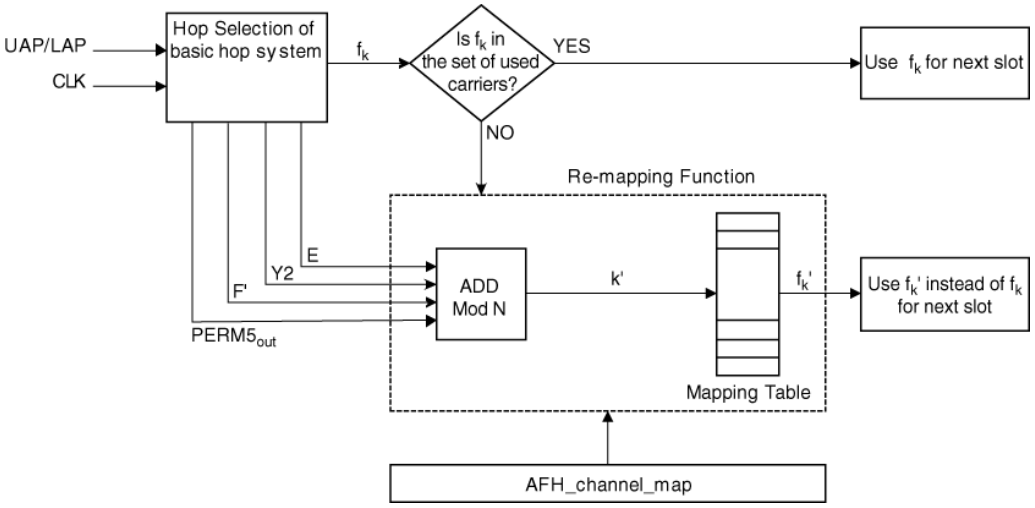


Figure 31—Block diagram of adaptive hop selection mechanism

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

$$k' = (PERM5_{out} + E + F' + Y2) \bmod N \quad (2)$$

where F' is defined in Table 13. The index k' is then used to select the remapped channel from a mapping table that contains all of the even used RF channels in ascending order followed by all the odd used RF channels in ascending order (i.e., the mapping table of Figure 27 with all the unused RF channels removed).

8.2.6.4 Control word

In 8.2.6.4 through 8.2.6.4.6, X_{j-i}, $i < j$, will denote bits i, $i + 1$,...,j of the bit vector X. By convention, X_0 is the LSB of the vector X.

The control word of the kernel is controlled by the overall control signals X, Y1, Y2, A to F, and F' as illustrated in Figure 27 and Figure 31. During paging and inquiry, the inputs A to E use the address values as given in the corresponding columns of Table 13. In addition, the inputs X, Y1, and Y2 are used. The inputs F and F' are unused. The clock bits CLK_{6-2} (i.e., input X) specifies the phase within the length 32 sequence. CLK_1 (i.e., inputs Y1 and Y2) is used to select between TX and RX. The address inputs determine the sequence order within segments. The final mapping onto the hop frequencies is determined by the register contents.

During the CONNECTION state (see 8.8.5), the inputs A, C, and D shall be derived from the address bits being bitwise XORed with the clock bits as shown in the "CONNECTION state" column of Table 13. (The 2 MSBs are XORed together; the two second MSBs are XORed together; etc.)

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

Table 13—Control for hop system

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

8.2.6.4.1 Page scan and inquiry scan hopping sequences

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

8.2.6.4.2 Page hopping sequence

When the sequence selection input is set to page, the paging device shall start using the **A**-train, i.e., $\{f(k-8), \ldots, f(k), \ldots, f(k+7)\}$, where $f(k)$ is the source's estimate of the current receiver frequency in the paged device. The index k is a function of all the inputs in Figure 27. There are 32 possible paging frequencies within each 1.28 s interval. Half of these frequencies belong to the **A**-train, the rest (i.e., $\{f(k+8), \ldots, f(k+15), f(k-16), \ldots, f(k-9)\}$) belong to the **B**-train. In order to achieve the –8 offset of the **A**-train, a constant of 24 shall be added to the clock bits (which is equivalent to –8 due to the modulo-32 operation). The **B**-train is obtained by setting the offset to 8. A cyclic shift of the order within the trains is also necessary in order to avoid a possible repetitive mismatch between the paging and scanning devices. Thus,

$$Xp = [\mathrm{CLKE}_{16-12} + k_{offset} + (\mathrm{CLKE}_{4-2,0} - \mathrm{CLKE}_{16-12}) \mod 16] \mod 32, \quad (3)$$

where

$$k_{offset} = \begin{cases} 24 & \text{A-train,} \\ 8 & \text{B-train.} \end{cases} \quad (4)$$

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

8.2.6.4.3 Slave page response hopping sequence

When the sequence selection input is set to slave page response, in order to eliminate the possibility of losing the link due to discrepancies of the CLKN and the estimate of the master's clock CLKE, the 4 bits CLKN_{16-12} shall be frozen at their current value. The value shall be frozen at the content it has in the slot where the recipient's access code is detected. The CLKN shall not be stopped; it is merely the values of the bits used for creating the input X that are kept fixed for a while. A frozen value is denoted by an asterisk (*) in the discussion in this subclause.

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

$$Xprs = [\mathrm{CLKN}^*_{16-12} + N] \bmod 32 \quad (5)$$

The counter N shall be set to zero in the slot where the slave acknowledges the page (see Figure 62 and Figure 63). Then, the value of N shall be increased by one each time CLKN_1 is set to zero, which corresponds to the start of a master TX slot. The input X shall be constructed this way until the first FHS packet is received and the immediately following response packet has been transmitted. After this, the slave shall enter the CONNECTION state using the parameters received in the FHS packet.

8.2.6.4.4 Master page response hopping sequence

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

$$\begin{aligned} Xprm = [&\mathrm{CLKE}^*_{16-12} + k_{offset}{}^* + \\ &(\mathrm{CLKE}^*_{4-2,0} - \mathrm{CLKE}^*_{16-12}) \bmod 16 + N] \bmod 32 \end{aligned} \quad (6)$$

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

8.2.6.4.5 Inquiry hopping sequence

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

$$Xi = [\mathrm{CLKN}_{16-12} + k_{offset} + (\mathrm{CLKN}_{4-2,0} - \mathrm{CLKN}_{16-12}) \mod 16] \mod 32, \quad (7)$$

where k_{offset} is defined by Equation (4). The initial choice of the offset is arbitrary.

The GIAC LAP and the 4 LSBs of the DCI (as A_{27-24}) shall be used as address input for the hopping sequence generator.

8.2.6.4.6 Inquiry response hopping sequence

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

$$Xir = [\mathrm{CLKN}_{16-12} + N] \mod 32 \quad (8)$$

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

The GIAC LAP and the 4 LSBs of the DCI (as A_{27-24}) shall be used as address input for the hopping sequence generator. The other input bits to the generator shall be the same as for page response.

8.2.6.4.7 Basic and adapted channel hopping sequence

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

8.3 Physical links

A physical link represents a BB 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 properties of the active physical link are as follows:

— Power control (see 9.3.1.3)
— Link supervision (see 8.3.1 and 9.3.1.6)
— Encryption (see 13.4 and 9.3.2.5)
— Channel quality-driven data rate (CQDDR) change (see 9.3.1.7)
— Multislot packet control (see 9.3.1.10)

All of these properties, except power control and CQDDR, may be applied to the parked physical link.

8.3.1 Link supervision

A connection can break down due to various reasons, such as when a device moves out of range or encounters severe interference or a power failure condition. Since this may happen without any prior warning, it is important to monitor the link on both the master and the slave side to avoid possible collisions when the LT_ADDR (see 8.4.2) or parked member address (PM_ADDR) (see 8.4.7.1) is reassigned to another slave.

To be able to detect link loss, both the master and the slave shall use a link supervision timer, $T_{supervision}$. Upon reception of a valid packet header with one of the slave's addresses (see 8.4.2) on the physical link, the

timer shall be reset. If, at any time in CONNECTION state, the timer reaches the *supervisionTO* value, the connection shall be considered disconnected. The same link supervision timer shall be used for SCO, eSCO, and ACL logical transports.

The timeout period *supervisionTO* is negotiated by the LM. Its value shall be chosen so that the supervision timeout will be longer than HOLD and SNIFF mode periods. Link supervision of a parked slave shall be done by unparking and reparking the slave.

8.4 Logical transports

8.4.1 General

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

— Synchronous connection-oriented (SCO) logical transport
— Extended synchronous connection-oriented (eSCO) logical transport
— Asynchronous connection-oriented (ACL) logical transport
— Active slave broadcast (ASB) logical transport
— Parked slave broadcast (PSB) logical transport

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

The ACL logical transport is also a point-to-point logical transport between the master and a slave. In the slots not reserved for synchronous logical transport(s), the master can establish an ACL logical transport on a per-slot basis to any slave, including the slave(s) already engaged in a synchronous logical transport. There are cases when the ACL logical transport may use slots reserved by synchronous logical transports, e.g., when all slots are reserved and the control logical link (LMP) has data to send.

The ASB logical transport is used by a master to communicate with active slaves. The PSB logical transport is used by a master for communications from the master to parked slave devices. Note that these communications may also be received by active devices.

8.4.2 Logical transport address (LT_ADDR)

Each slave active in a piconet is assigned a primary 3-bit LT_ADDR. The all-zero LT_ADDR is reserved for broadcast messages. The master does not have an LT_ADDR. A master's timing relative to the slaves' distinguishes it from the slaves. A secondary LT_ADDR is assigned to the slave for each eSCO logical transport in use in the piconet. Only eSCO traffic (i.e., NULL, POLL, and one of the **EV** packet types as negotiated at eSCO logical transport setup) may be sent on these LT_ADDRs. ACL traffic (including LMP) shall always be sent on the primary LT_ADDR. A slave shall accept only packets with matching primary or secondary LT_ADDR and broadcast packets. The LT_ADDR is carried in the packet header (see 8.6.4). The LT_ADDR shall be valid only 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, at role switch, or when the slave is unparked. At connection establishment and at role switch, the primary LT_ADDR is carried in the FHS payload. When unparking, the primary LT_ADDR is carried in the unpark message.

8.4.3 Synchronous logical transports

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

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

8.4.4 Asynchronous logical transport

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

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

8.4.5 Transmit/receive routines

This subclause describes the way to use the packets as defined in IEEE 802.15.1-2005 device in order to support the traffic on the ACL, SCO, and eSCO links. Both single-slave and multislave configurations are considered.

8.4.5.1 Flow control

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

8.4.5.1.1 Destination control

As long as data cannot be received, a stop indication shall be transmitted, and it is automatically inserted by the link controller into the header of the return packet. “Stop” shall be returned as long as the RX ACL buffer is not emptied by the BB resource manager. When new data can be accepted again, the go indication shall be returned. “Go” shall be the default value. All packet types not including data can still be received. Voice communication, for example, is not affected by the flow control. Although a device cannot receive new information, it may still continue to transmit information: the flow control shall be separate for each direction.

8.4.5.1.2 Source control

On the reception of a stop signal, the link controller shall automatically switch to the default packet type. The ACL packet transmitted just before the reception of the stop indication shall be kept until a go signal is received. It may be retransmitted as soon as a go indication is received. Only default packets shall be sent as

long as the stop indication is received. When no packet is received, “go” shall be assumed implicitly. Note that the default packets contain link control information (in the header) for the receive direction (which may still be open) and may contain synchronous data (**HV** or **EV** packets). When a go indication is received, the link controller may resume transmitting the data that are present in the TX ACL buffers.

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

8.4.6 Active slave broadcast (ASB) transport

The ASB logical transport is used to transport L2CAP user traffic to all devices in the piconet that are currently connected to the piconet physical channel that is used by the ASB. There is no acknowledgment protocol, and the traffic is unidirectional from the piconet master to the slaves. The ASB logical transport may be used only for L2CAP group traffic and shall never be used for L2CAP connection-oriented channels, L2CAP control signalling, or LMP control signalling.

The ASB logical transport is unreliable. To improve reliability somewhat each packet is transmitted a number of times. An identical SEQN is used to assist with filtering retransmissions at the slave device.

The ASB logical transport is identified by the reserved, all-zero LT_ADDR. Packets on the ASB logical transport may be sent by the master at any time.

8.4.7 Parked slave broadcast (PSB) transport

The PSB logical transport is used for communication from the master to the slaves that are parked. The PSB logical transport is more complex than the other logical transports as it consists of a number of phases, each having a different purpose. These phases are the control information phase (used to carry the LMP logical link), the user information phase (used to carry the L2CAP logical link), and the access phase (carrying BB signalling).

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

8.4.7.1 Parked member address (PM_ADDR)

A slave in the PARK state can be identified by its BD_ADDR or by a dedicated PM_ADDR. This latter address is an 8-bit member address that separates the parked slaves. The PM_ADDR shall be valid only for as long as the slave is parked. When the slave is activated, it shall be assigned an LT_ADDR, but shall lose the PM_ADDR. The PM_ADDR is assigned to the slave by the master during the parking procedure (see 9.3.5.2).

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

8.4.7.2 Access request address (AR_ADDR)

The AR_ADDR is used by the parked slave to determine the slave-to-master half slot in the access window where it is allowed to send access request messages (see also 8.8.9.6). The AR_ADDR shall be assigned to the slave when it enters the PARK state and shall be valid only for as long as the slave is parked. The AR_ADDR is not necessarily unique, i.e., different parked slaves may have the same AR_ADDR.

8.5 Logical links

Four logical links are defined as follows:

— Link control (LC)
— ACL control (ACL-C)
— Asynchronous/Isochronous user (ACL-U)
— Stream (SCO-S or eSCO-S)

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

8.5.1 Link control (LC) logical link

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

8.5.2 ACL control (ACL-C) logical link

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

8.5.3 Asynchronous/Isochronous user (ACL-U) logical link

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

8.5.3.1 Pausing the ACL-U logical link

When the ACL-U logical link is paused by the LM, the link controller transmits the current packet with ACL-U information, if any, until an ACK is received or, optionally, until an explicit negative acknowledge (NAK) is received. While the ACL-U logical link is paused, the link controller shall not transmit any packets with ACL-U logical link information.

If the ACL-U logical link was paused after an ACK, the next SEQN shall be used on the next packet. If the ACL-U logical link was paused after a NAK, the same SEQN shall be used on the next packet, and the unacknowledged packet shall be transmitted once the ACL-U logical link is unpaused.

When the ACL-U logical link is unpaused by the LM, the link controller may resume transmitting packets with ACL-U information.

8.5.4 Stream logical link

The stream logical link carries transparent synchronous data. This logical link may be carried over the synchronous logical transport (SCO) or over the extended synchronous logical transport (eSCO).

8.5.5 Logical link priorities

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

8.6 Packets

IEEE 802.15.1-2005 devices shall use the packets as defined in 8.6.1 through 8.6.7.

8.6.1 General format

The general packet format is shown in Figure 32. Each packet consists of three entities: the access code, the header, and the payload. In the figure, the number of bits per entity is indicated.

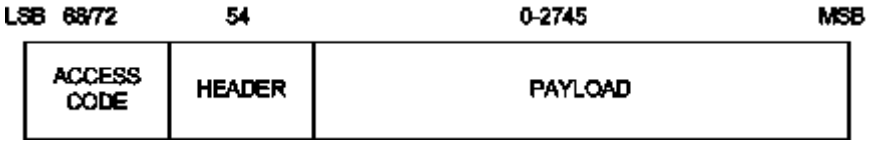


Figure 32—General packet format

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

— The shortened access code only
— The access code and the packet header
— The access code, the packet header, and the payload

8.6.2 Bit ordering

The bit ordering when defining packets and messages in BB follows the little Endian format. The following rules apply:

– The LSB corresponds to b_0.
– The LSB is the first bit sent over the air.
– In illustrations, the LSB is shown on the left side.

Furthermore, data fields generated internally at BB level, such as the packet header fields and payload header length, shall be transmitted with the LSB first. For instance, a 3-bit parameter X = 3 is sent as follows:

$$b_0 b_1 b_2 = 110$$

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

8.6.3 Access code

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

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

The access code consists of a preamble, a sync word, and possibly a trailer (see Figure 33). For details, see 8.6.3.1.

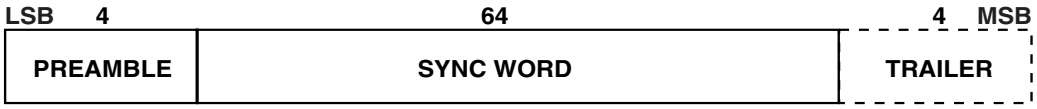


Figure 33—Access code format

8.6.3.1 Access code types

The different access code types use different LAPs to construct the sync word. The LAP field of the BD_ADDR is explained in 8.1.2. A summary of the different access code types is in Table 14.

Table 14—Summary of access code types

Code type	LAP	Code length	Comments
CAC	Master	72	See also 8.1.3
DAC	Paged device	68/72[a]	
GIAC	Reserved	68/72[a]	
DIAC	Dedicated	68/72[a]	

[a]Length 72 is used only in combination with FHS packets.

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

8.6.3.2 Preamble

The preamble is a fixed zero-one pattern of four symbols used to facilitate dc compensation. The sequence is either 1010 or 0101, depending on whether the LSB of the following sync word is 1 or 0, respectively. The preamble is shown in Figure 34.

Figure 34—Preamble

8.6.3.3 Sync word

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

8.6.3.3.1 Synchronization word definition

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

a) Generate information sequence.
b) XOR this with the "information covering" part of the PN overlay sequence.
c) Generate the code word.
d) XOR the code word with all 64 bits of the PN overlay sequence.

The information sequence is generated by appending 6 bits to the 24-bit LAP (step a). The appended bits are 001101 if the MSB of the LAP equals 0. If the MSB of the LAP is 1, the appended bits are 110010. The LAP MSB together with the appended bits constitute a length-seven Barker sequence. The purpose of including a Barker sequence is to further improve the auto-correlation properties. In step b, the information is prescrambled by XORing it with the bits $p_{34} \ldots p_{63}$ of the PN sequence (defined in 8.6.3.3.2). After generating the code word (step c), the complete PN sequence is XORed to the code word (step d). This step descrambles the information part of the code word. At the same time, the parity bits of the code word are scrambled. Consequently, the original LAP and Barker sequence are ensured a role as a part of the access code sync word, and the cyclic properties of the underlying code are removed. The principle is depicted in Figure 35

In the following discussion, binary sequences will be denoted by their corresponding D-transform (in which D^i represents a delay of i time units). Let $p'(D) = p'_0 + p'_1 D + \ldots + p'_{62} D^{62}$ be the 63-bit PN sequence, where p'_0 is the first bit (LSB) leaving the pseudo-random noise generation (see Figure 36), and p'_{62} is the last bit (MSB). To obtain 64 bits, an extra zero is appended at the end of this sequence (thus, $p'(D)$ is unchanged). For notational convenience, the reciprocal of this extended polynomial, $p(D) = D^{63} p'(1/D)$, will be used in the following discussion. This is the sequence $p'(D)$ in reverse order. The 24-bit LAP of the device address is denoted by $a(D) = a_0 + a_1 D + \ldots + a_{23} D^{23}$ (a_0 is the LSB of the device address).

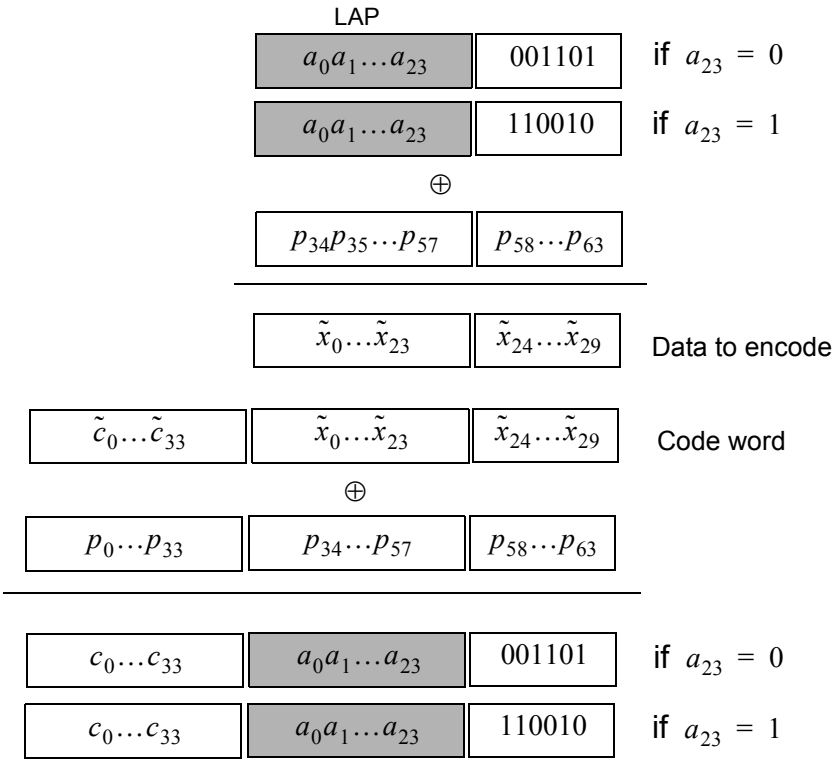


Figure 35—Construction of the sync word

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

$$g(D) = \texttt{260534236651,} \tag{9}$$

where the leftmost bit corresponds to the high-order (g_{34}) coefficient. The dc-free 4-bit sequences `0101` and `1010` can be written as

$$\begin{cases} F_0(D) = D + D^3, \\ F_1(D) = 1 + D^2, \end{cases} \tag{10}$$

respectively. Furthermore,

$$\begin{cases} B_0(D) = D^2 + D^3 + D^5, \\ B_1(D) = 1 + D + D^4, \end{cases} \tag{11}$$

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

a) Format the 30 information bits to encode:

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

b) Add the information covering part of the PN overlay sequence:

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

c) Generate parity bits of the (64,30) expurgated block code:[9]

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

d) Create the code word:

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

e) Add the PN sequence:

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

f) Append the (dc-free) preamble and trailer:

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

8.6.3.3.2 PN sequence generation

To generate the PN sequence, the primitive polynomial $h(D) = 1 + D + D^3 + D^4 + D^6$ shall be used. The linear feedback shift register (LFSR) and its starting state are shown in Figure 36. The PN sequence generated (including the extra terminating zero) becomes (hexadecimal notation) `83848D96BBCC54FC`. The LFSR output starts with the leftmost bit of this PN sequence. This corresponds to $p'(D)$ of 8.6.3.3.1. Thus, using the reciprocal $p(D)$ as overlay gives the 64-bit sequence:

$$p = \texttt{3F2A33DD69B121C1}, \tag{12}$$

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

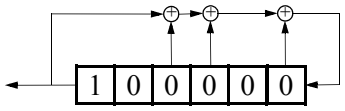


Figure 36—LFSR and the starting state to generate *p′*(*D*)

8.6.3.4 Trailer

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

The trailer is a fixed zero-one pattern of four symbols. The trailer together with the 3 MSBs of the sync word form a 7-bit pattern of alternating ones and zeroes, which may be used for extended dc compensation. The trailer sequence is either 1010 or 0101 depending on whether the MSB of the sync word is 0 or 1, respectively. The choice of trailer is illustrated in Figure 37.

[9] $x(D)$ mod $y(D)$ denotes the remainder when $x(D)$ is divided by $y(D)$.

Figure 37—Trailer in CAC when MSB of sync word is 0 (a) and when MSB of sync word is 1 (b)

8.6.4 Packet header

The header contains link control information and consists of six fields:

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

The total header, including the HEC, consists of 18 bits (see Figure 38) and is encoded with a rate 1/3 FEC (not shown, but described in 8.7.4) resulting in a 54-bit header. The LT_ADDR and TYPE fields shall be sent LSB first.

Figure 38—Header format

8.6.4.1 LT_ADDR field

The 3-bit LT_ADDR field contains the LT_ADDR for the packet (see 8.4.2). This field indicates the destination slave for a packet in a master-to-slave transmission slot and indicates the source slave for a slave-to-master transmission slot.

8.6.4.2 TYPE field

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 LT_ADDR in the packet. First, it shall be determined whether the packet is sent on an SCO logical transport, an eSCO logical transport, or an ACL logical transport. Then it can be determined which type of SCO packet, eSCO packet, or ACL packet has been received. The TYPE code determines how many slots the current packet will occupy (see the “Slot occupancy” column in Table 15 in 8.6.5). This allows the nonaddressed receivers to refrain from listening to the channel for the duration of the remaining slots. In 8.6.5, each packet type is described in more detail.

8.6.4.3 FLOW field

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 affects only 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 indication 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.

8.6.4.4 ARQN field

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 (ACK) or negative (NAK). See 8.7.6 for initialization and usage of this bit.

8.6.4.5 SEQN field

The SEQN bit provides a sequential numbering scheme to order the data packet stream. See 8.7.6.2 for initialization and usage of the SEQN bit. For broadcast packets, a modified sequencing method is used (see 8.7.6.5).

8.6.4.6 HEC field

Each header has an HEC to check the header integrity. The HEC is an 8-bit word (generation of the HEC is specified in 8.7.1.1). Before generating the HEC, the HEC generator is initialized with an 8-bit value. For FHS packets sent in **master response** substate, the slave upper address part (UAP) shall be used. For FHS packets sent in **inquiry response** substate, the DCI (see 8.1.2.1) shall be used. In all other cases, the UAP of the master device shall be used.

After the initialization, a HEC shall be calculated for the 10 header bits. Before checking the HEC, the receiver shall initialize the HEC check circuitry with the proper 8-bit UAP (or DCI). If the HEC does not check, the entire packet shall be discarded. More information can be found in 8.7.1.

8.6.5 Packet types

The packets used on the piconet are related to the logical transports in which they are used. Three logical transports with distinct packet types are defined (see 8.4): the SCO logical transport, the eSCO logical transport, and the ACL logical transport. (The ASB and PSB logical transports do not have distinct packet types, but use the packet types defined for the ACL logical transport.) For each of these logical transports, 15 different packet types can be defined.

To indicate the different packets on a logical transport, the 4-bit TYPE code is used. The packet types are divided into four segments. The first segment is reserved for control packets. All control packets occupy a single time slot. The second segment is reserved for packets occupying a single time slot. The third segment is reserved for packets occupying three time slots. The fourth segment is reserved for packets occupying five time slots. The slot occupancy is reflected in the segmentation and can directly be derived from the type code. Table 15 summarizes the packets defined for the SCO, eSCO, and ACL logical transport types.

8.6.5.1 Common packet types

There are five common kinds of packets. In addition to the types listed in segment 1 of Table 15, the ID packet is also a common packet type. It is not listed in segment 1, however, because it does not have a packet header.

Table 15—Packets defined for synchronous and asynchronous logical transport types

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

8.6.5.1.1 ID packet

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

8.6.5.1.2 NULL packet

The NULL packet has no payload and consists of the CAC and packet header only. Its total (fixed) length is 126 bits. The NULL packet may be used to return link information to the source regarding the success of the previous transmission (ARQN) or the status of the RX buffer (FLOW). The NULL packet does not require an acknowledgment.

8.6.5.1.3 POLL packet

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

8.6.5.1.4 FHS packet

The FHS packet is a special control packet containing, among other things, the 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 39 illustrates the format and contents of the FHS payload. The payload consists of 11 fields. The FHS packet is used in page master response, inquiry response and 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 from 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 physical channel has been established or when an existing piconet changes to a new piconet.

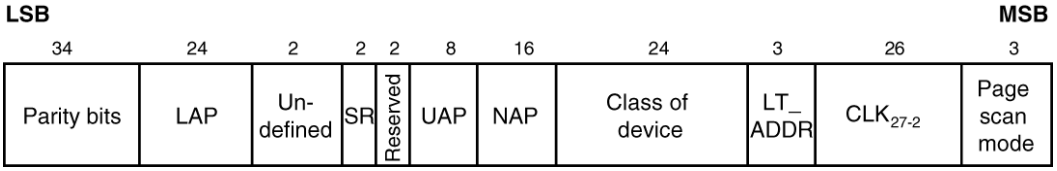


Figure 39—Format of the FHS payload

Each field is described in more detail below and in Table 16:

Table 16—Description of the FHS payload

Field name	Description
Parity Bits	This 34-bit field contains the parity bits that form the first part of the sync word of the access code of the device that sends the FHS packet. These bits are derived from the LAP as described in 8.1.2.
LAP	This 24-bit field shall contain the LAP of the device that sends the FHS packet.
Undefined	This 2-bit field is reserved for future use and shall be set to zero.
SR	This 2-bit field is the scan repetition field and indicates the interval between two consecutive page scan windows (see also Table 17 and Table 24).
Reserved	This 2-bit field shall be set to 10.
UAP	This 8-bit field shall contain the UAP of the device that sends the FHS packet.
NAP	This 16-bit field shall contain the NAP of the device that sends the FHS packet (see also 8.1.2 for LAP, UAP, and NAP).
Class of Device	This 24-bit field shall contain the class of the device that sends the FHS packet. The field is defined in Bluetooth Assigned Numbers [B1].
LT_ADDR	This 3-bit field shall contain the LT_ADDR the recipient shall use if the FHS packet is used at connection setup or role switch. A slave responding to a master or a device responding to an inquiry request message shall include an all-zero LT_ADDR field if it sends the FHS packet.

Table 16—Description of the FHS payload *(continued)*

Field name	Description
CLK_{27-2}	This 26-bit field shall contain the value of the CLKN of the device that sends the FHS packet, sampled at the beginning of the transmission of the access code of this FHS packet. This clock value has a resolution of 1.25 ms (two-slot interval). For every new transmission, this field is updated so that it accurately reflects the real-time clock value.
Page Scan Mode	This 3-bit field shall indicate which scan mode is used by the sender of the FHS packet. The interpretation of the page scan mode is illustrated in Table 18.

— The device sending the FHS shall set the SR bits according to Table 17.

Table 17—Contents of SR field

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

— The device sending the FHS shall set the Page Scan Mode bits according to Table 18.

Table 18—Contents of Page Scan Mode field

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

— The LAP, UAP, and NAP together form the 48-bit address of the device that sends the FHS packet. Using the parity bits and the LAP, the recipient can directly construct the CAC of the sender of the FHS packet.
— The DCI shall be used to initialize the HEC and CRC for the FHS packet of the inquiry response (see 8.1.2.1).

8.6.5.1.5 DM1 packet

DM1 is part of segment 1 in order to support control messages in any logical transport that allows the **DM1** packet (see Table 15). However, it may also carry regular user data. Since the **DM1** packet can be regarded as an ACL packet, it will be discussed in 8.6.5.4.

8.6.5.2 SCO packets

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

8.6.5.2.1 HV1 packet

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

8.6.5.2.2 HV2 packet

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

8.6.5.2.3 HV3 packet

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

8.6.5.2.4 DV packet

The **DV** packet is a combined data-voice packet. The **DV** packet shall be used only in place of an **HV1** packet. The payload is divided into a Voice field of 80 bits and a Data field containing up to 150 bits (see Figure 40). The Voice field is not protected by FEC. The Data field has between 1 and 10 information bytes (including the 1-byte payload header) and includes a 16-bit CRC. The data field is encoded with a rate 2/3 FEC. Since the **DV** packet has to be sent at regular intervals due to its synchronous contents, it is listed under the SCO packet types. The Voice and Data fields shall be treated separately. The Voice field shall be handled in the same way as normal SCO data and shall never be retransmitted, i.e., the Voice field is always new. The Data field is checked for errors and shall be retransmitted if necessary. When the asynchronous data field in the **DV** packet has not been acknowledged before the SCO logical transport is terminated, the asynchronous Data field shall be retransmitted in a **DM1** packet.

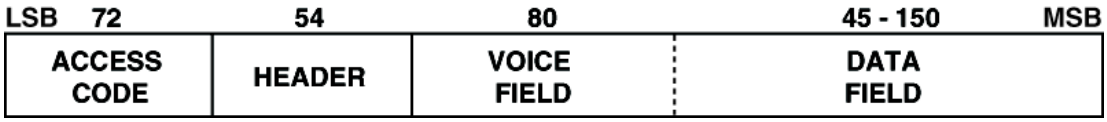


Figure 40—DV packet format

8.6.5.3 eSCO packets

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

8.6.5.3.1 EV3 packet

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

8.6.5.3.2 EV4 packet

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

8.6.5.3.3 EV5 packet

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

8.6.5.4 ACL packets

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

8.6.5.4.1 DM1 packet

The **DM1** packet carries data information only. The payload has between 1 and 18 information bytes (including the 1-byte payload header) plus a 16-bit CRC code. The **DM1** packet occupies a single time slot. The information plus CRC bits are coded with a rate 2/3 FEC. The payload header in the **DM1** packet is 1 byte long (see Figure 41). The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).

8.6.5.4.2 DH1 packet

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

8.6.5.4.3 DM3 packet

The **DM3** packet may occupy up to three time slots. The payload has between 2 and 123 information bytes (including the 2-byte payload header) plus a 16-bit CRC code. The information plus CRC bits are coded with a rate 2/3 FEC. The payload header in the **DM3** packet is 2 bytes long (see Figure 42). The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).

8.6.5.4.4 DH3 packet

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

8.6.5.4.5 DM5 packet

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

8.6.5.4.6 DH5 packet

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

8.6.5.4.7 AUX1 packet

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

8.6.6 Payload format

In the payload, two fields are distinguished: the Synchronous Data field and the Asynchronous Data field. The ACL packets have only the Asynchronous Data field, and the SCO and eSCO packets have only the Synchronous Data field—with the exception of the **DV** packets, which have both.

8.6.6.1 Synchronous Data field

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

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

8.6.6.1.1 Synchronous data body

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

8.6.6.1.2 CRC code

The 16-bit CRC in the payload is generated as specified in 8.7.1. The 8-bit UAP of the master is used to initialize the CRC generator.

8.6.6.2 Asynchronous data field

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

8.6.6.2.1 Payload header

The payload header is one or two bytes long. Packets in segments 1 and 2 (see Table 15) have a 1-byte payload header; packets in segments 3 and 4 (see Table 15) have a 2-byte payload header. The payload header specifies the logical link (2-bit LLID indication), controls the flow on the logical channels (1-bit FLOW indication), and has a payload length indicator (5 bits and 9 bits for 1-byte and 2-byte payload headers, respectively). In the case of a 2-byte payload header, the length indicator is extended by 4 bits into the next byte. The remaining 4 bits of the second byte are reserved for future use and shall be set to zero. The formats of the 1-byte and 2-byte payload headers are shown in Figure 41 and Figure 42.

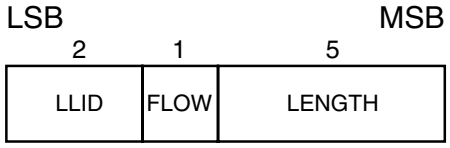


Figure 41—Payload header format for single-slot ACL packets

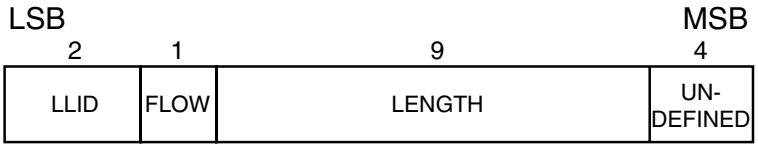


Figure 42—Payload header format for multislot ACL packets

The LLID field shall be transmitted first, the Length field last. In Table 19, more details about the contents of the LLID field are listed.

Table 19—LLID field contents

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

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

The flow indicator in the payload is used to control the flow at the L2CAP level. It is used to control the flow per logical link. FLOW = 1 means flow-on (go), and FLOW = 0 means flow-off (stop). After a new

connection has been established, the flow indicator shall be set to "go." When a device receives a payload header with the FLOW bit set to "stop," it shall stop the transmission of ACL packets before an additional amount of payload data is sent. This amount is defined as the *flow control lag*, expressed as a number of bytes. The shorter the flow control lag, the less buffering the other device must dedicate to this function. The flow control lag shall not exceed 1792 bytes (7 × 256 bytes). In order to allow devices to optimize the selection of packet length and buffer space, the flow control lag of a given implementation shall be provided in the LMP_features_res message.

If a packet containing the payload FLOW bit of "stop" is received with a valid packet header, but bad payload, the payload flow control bit shall be ignored. The BB 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 BB 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 8.6.4.3). 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 20). On the ACL-C logical link, no flow control is applied, and the payload FLOW bit shall always be set to one.

Table 20—Use of payload header FLOW bit on the logical links

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

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 41 and Figure 42, the MSB of the length field in a 1-byte header is the last (rightmost) bit in the payload header; the MSB of the length field in a 2-byte header is the fourth bit (from left) of the second byte in the payload header.

8.6.6.2.2 Payload body

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

8.6.6.2.3 CRC code generation

The 16-bit CRC code in the payload is generated as specified in 8.7.1. Before determining the CRC code, an 8-bit value is used to initialize the CRC generator. For the CRC code in the FHS packets sent in **master**

response substate, the UAP of the slave is used. For the FHS packet sent in **inquiry response** substate, the DCI (see 8.1.2.1) is used. For all other packets, the UAP of the master is used.

8.6.7 Packet summary

A summary of the packets and their characteristics are shown in Table 21, Table 22, and Table 23. The payload represents the packet payload excluding FEC, CRC, and payload header.

Table 21—Link control packets

Type	Payload (bytes)	FEC	CRC	Symmetric maximum rate	Asymmetric maximum rate
ID	NA	NA	NA	NA	NA
NULL	NA	NA	NA	NA	NA
POLL	NA	NA	NA	NA	NA
FHS	18	2/3	Yes	NA	NA

Table 22—ACL packets

Type	Payload header (bytes)	Payload (bytes)	FEC	CRC	Symmetric maximum rate (kb/s)	Asymmetric maximum rate (kb/s)	
						Forward	Reverse
DM1	1	0–17	2/3	Yes	108.8	108.8	108.8
DH1	1	0–27	No	Yes	172.8	172.8	172.8
DM3	2	0–121	2/3	Yes	258.1	387.2	54.4
DH3	2	0–183	No	Yes	390.4	585.6	86.4
DM5	2	0–224	2/3	Yes	286.7	477.8	36.3
DH5	2	0–339	No	Yes	433.9	723.2	57.6
AUX1	1	0–29	No	No	185.6	185.6	185.6

Table 23—Synchronous packets

Type	Payload header (bytes)	Payload (bytes)	FEC	CRC	Symmetric maximum rate (kb/s)
HV1	NA	10	1/3	No	64.0
HV2	NA	20	2/3	No	64.0
HV3	NA	30	No	No	64.0
DV[a]	1 D	10 + (0 – 9) D	2/3 D	Yes D	64.0 + 57.6 D

Table 23—Synchronous packets *(continued)*

Type	Payload header (bytes)	Payload (bytes)	FEC	CRC	Symmetric maximum rate (kb/s)
EV3	NA	1-30	No	Yes	96
EV4	NA	1–120	2/3	Yes	192
EV5	NA	1–180	No	Yes	288

[a]Items followed by D relate to data field only.

8.7 Bitstream processing

IEEE 802.15.1-2005 devices shall use the bitstream processing schemes as defined in 8.7.1 through 8.7.6.

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 43 shows the processes carried out for the packet header both at the transmit and the receive side. All header bit processes are mandatory.

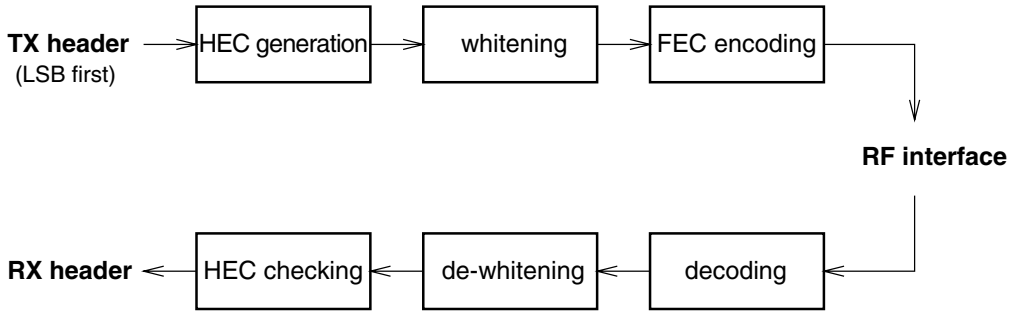


Figure 43—Header bit processes

Figure 44 shows the processes that may be carried out on the payload. In addition to the processes defined for the packet header, encryption may be applied on the payload. Only whitening and dewhitening, as explained in 8.7.2, are mandatory for every payload (with the exception of test mode); all other processes are optional and depend on the packet type (see 8.6.6) and whether encryption is enabled. In Figure 44, optional processes are indicated by dashed blocks.

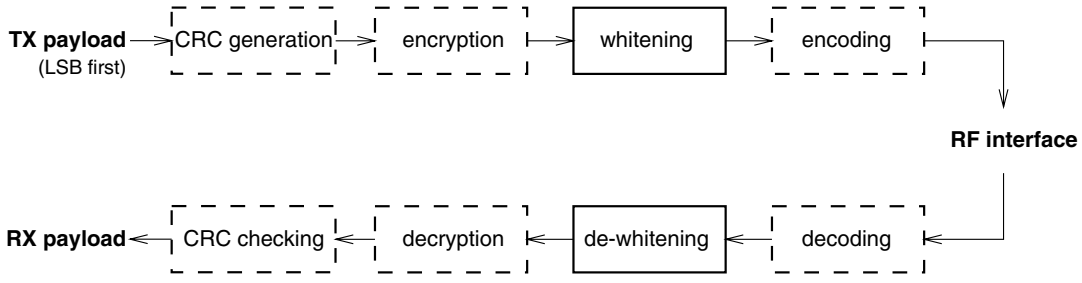


Figure 44—Payload bit processes

8.7.1 Error checking

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

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

The generation and check of the HEC and CRC are summarized in Figure 47 and Figure 50. Before calculating the HEC or CRC, the shift registers in the HEC/CRC generators shall be initialized with the 8-bit UAP (or DCI) value. Then the header and payload information shall be shifted into the HEC and CRC generators, respectively (with the LSB first).

8.7.1.1 HEC generation

The HEC generating LFSR is depicted in Figure 45. The generator polynomial is $g(D) = (D + 1)(D^7 + D^4 + D^3 + D^2 + 1) = D^8 + D^7 + D^5 + D^2 + D + 1$. Initially, this circuit shall be preloaded with the 8-bit UAP so that the LSB of the UAP (denoted UAP_0) goes to the leftmost shift register element, and UAP_7 goes to the rightmost element. The initial state of the HEC LFSR is depicted in Figure 46. Then the data shall be shifted in with the switch S set in position 1. When the last data bit has been clocked into the LFSR, the switch S shall be set in position 2, and the HEC can be read out from the register. The LFSR bits shall be read out from right to left (i.e., the bit in position 7 is the first to be transmitted, followed by the bit in position 6, and so on).

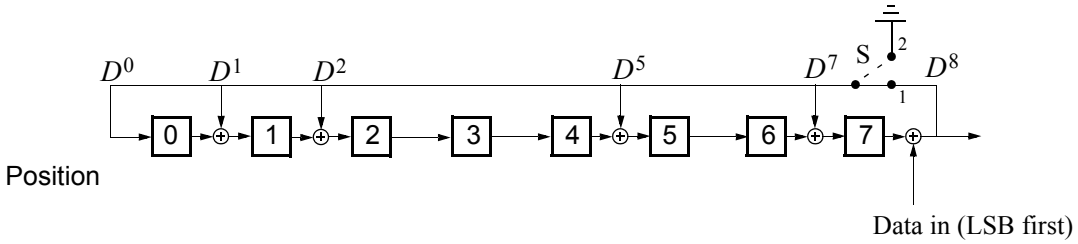


Figure 45—The LFSR circuit generating the HEC

Position:	0	1	2	3	4	5	6	7
LFSR:	UAP_0	UAP_1	UAP_2	UAP_3	UAP_4	UAP_5	UAP_6	UAP_7

Figure 46—Initial state of the HEC generating circuit

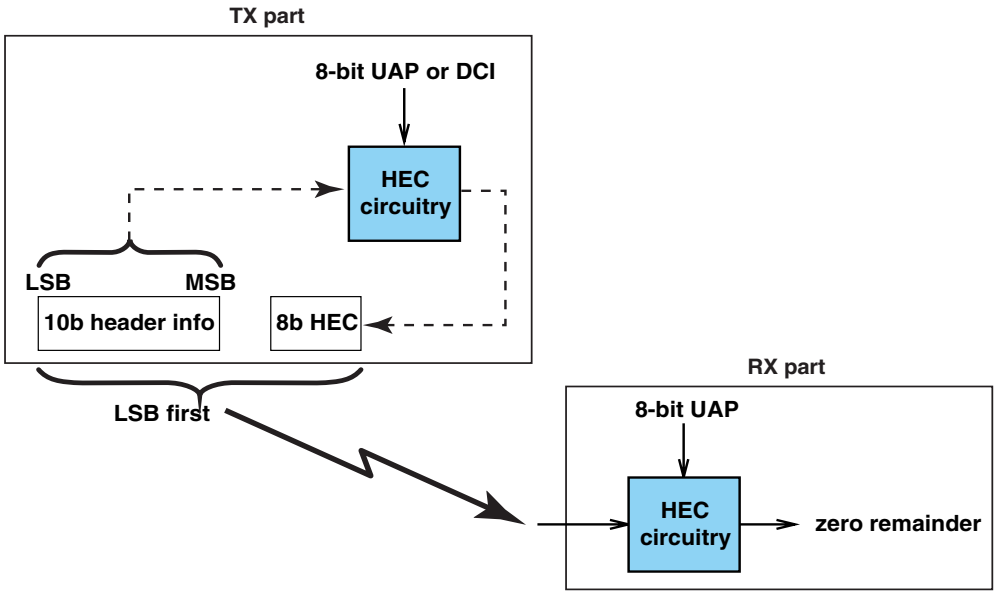


Figure 47—HEC generation and checking

8.7.1.2 CRC generation

The 16-bit LFSR for the CRC is constructed similarly to the HEC using the CRC-CCITT generator polynomial $g(D) = D^{16} + D^{12} + D^5 + 1$ (i.e., `210041` in octal representation) (see Figure 48). For this case, the 8 leftmost bits shall be initially loaded with the 8-bit UAP (UAP_0 to the left and UAP_7 to the right) while the 8 rightmost bits shall be reset to zero. The initial state of the 16-bit LFSR is specified in Figure 49. The switch S shall be set in position 1 while the data are shifted in. After the last bit has entered the LFSR, the switch shall be set in position 2, and the register's contents shall be transmitted, from right to left (i.e., starting with position 15, then position 14, and so on).

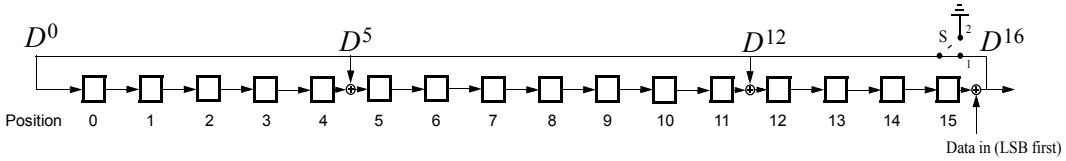


Figure 48—The LFSR circuit generating the CRC

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

Figure 49—Initial state of the CRC generating circuit

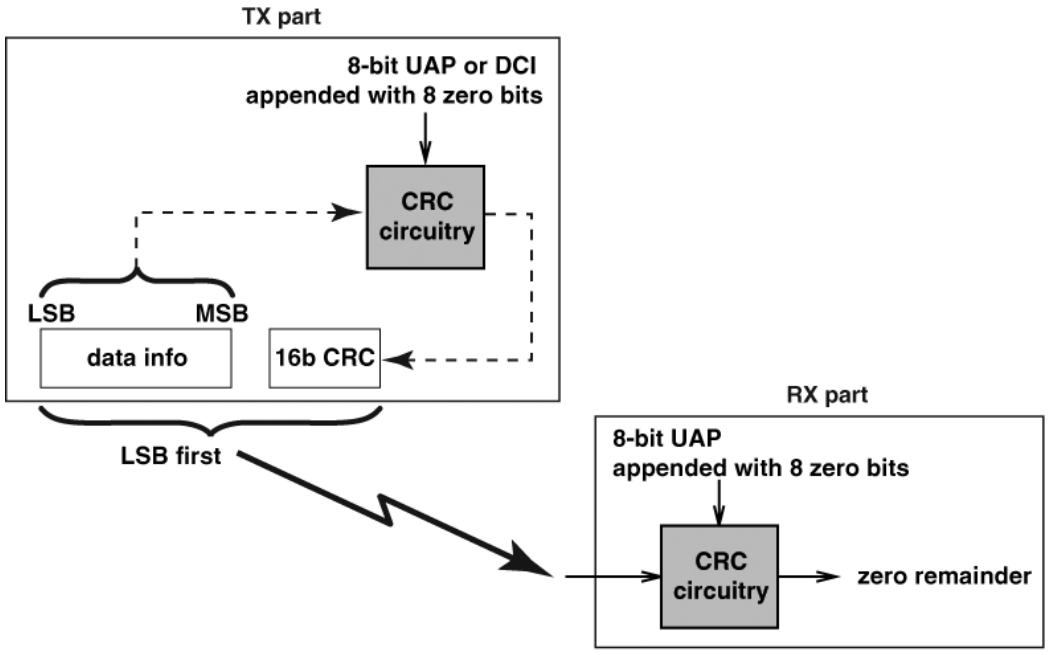


Figure 50—CRC generation and checking

8.7.2 Data whitening

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

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

The whitening word is generated with the polynomial $g(D) = D^7 + D^4 + 1$ (i.e., 221 in octal representation) and shall be subsequently XORed with the header and the payload. The whitening word is generated with the LFSR shown in Figure 51. Before each transmission, the shift register shall be initialized with a portion of the master clock CLK_{6-1}, extended with an MSB of value one. This initialization shall be carried out with CLK_1 written to position 0, CLK_2 written to position 1, etc. An exception is the FHS packet sent during page response or inquiry, where initialization of the whitening register shall be carried out differently. Instead of CLK, the input X used in the inquiry or page response (depending on current state) routine shall be used (see Table 13). The 5-bit value shall be extended with 2 MSBs of value 1. During register initialization, the LSB of X (i.e., X_0) shall be written to position 0, X_1 shall be written to position 1, etc.

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

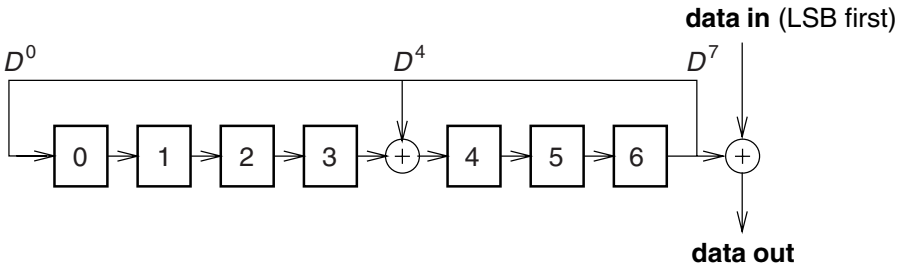


Figure 51—Data whitening LFSR

8.7.3 Error correction

There are three error correction schemes defined for this standard:

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

The purpose of the FEC scheme on the data payload is to reduce the number of retransmissions. However, in a reasonable error-free environment, FEC gives unnecessary overhead that reduces the throughput. Therefore, the packet definitions given in 8.6 have been kept flexible to use FEC in the payload or not, resulting in the **DM** and **DH** packets for the ACL logical transport, **HV** packets for the SCO logical transport, and **EV** packets for the eSCO logical transport. The packet header is always protected by a 1/3 rate FEC since it contains valuable link information and is designed to withstand more bit errors.

Correction measures to mask errors in the voice decoder are not included in this subclause. This matter is discussed in 8.9.3.

8.7.4 FEC code: rate 1/3

A simple 3-times repetition FEC code is used for the header. The repetition code is implemented by repeating each bit three times (see the illustration in Figure 52). The 3-times repetition code is used for the entire header as well as for the Synchronous Data field in the **HV1** packet.

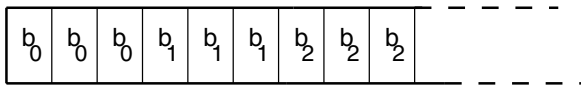


Figure 52—Bit-repetition encoding scheme

8.7.5 FEC code: rate 2/3

The other FEC scheme is a (15,10) shortened Hamming code. The generator polynomial is $g(D) = (D + 1)(D^4 + D + 1)$. This corresponds to 65 in octal notation. An LFSR that may be used to generate this code is depicted in Figure 53. Initially all register elements are set to zero. The 10 information bits are sequentially fed into the LFSR with the switches S1 and S2 set in position 1. Then, after the final input bit, the switches S1 and S2 are set in position 2, and the 5 parity bits are shifted out. The parity bits are appended to the information bits. Subsequently, each block of 10 information bits is encoded into a 15-bit code word. This code can correct all single errors and detect all double errors in each code word. This 2/3 rate FEC is used in the **DM** packets, in the Data field of the **DV** packet, in the FHS packet, in the **HV2** packet, and in the **EV4** packet. Since the encoder operates with information segments of length 10, tail bits with value zero shall be appended after the CRC bits to bring the total number of bits equal to a multiple of 10.

The number of tail bits to append shall be the least possible that achieves this (i.e., in the interval 0...9). These tail bits are not included in the payload length indicator for ACL packets or in the payload length field of the eSCO setup LMP command.

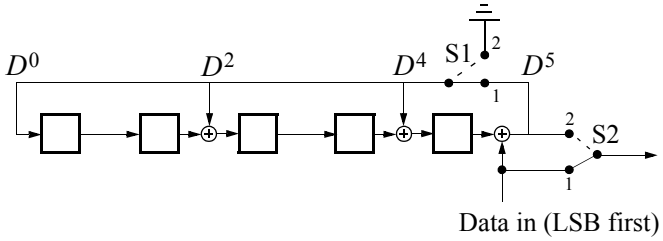


Figure 53—LFSR generating the (15,10) shortened Hamming code

8.7.6 Automatic repeat request (ARQ) scheme

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

The rules in 8.7.6.1 through 8.7.6.4 do not apply to broadcast packets.

8.7.6.1 Unnumbered ARQ

IEEE 802.15.1-2005 uses a fast, unnumbered acknowledgment scheme. An ACK (ARQN = 1) or a NAK (ARQN = 0) is returned in response to the receipt of previously received packet. The slave shall respond in the slave-to-master slot directly following the master-to-slave slot unless the slave has scatternet commitments in that timeslot. The master shall respond at the next event addressing the same slave (the master may have addressed other slaves between the last received packet from the considered slave and the master response to this packet). For a packet reception to be successful, at least the HEC must pass. In addition, the CRC must pass if present.

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

The ARQ bit shall be affected only by empty slots and data packets containing CRC. As shown in Figure 54, upon successful reception of a CRC packet, the ARQN bit shall be set to ACK. The ARQN bit shall be set to NAK if, in any receive slot in the slave or in a receive slot in the master following transmission of a packet, one or more of the following events applies:

— No access code is detected.
— The HEC fails.
— The CRC fails.

In eSCO, the receiving device may be able to use an erroneous packet, in which case it may set the ARQN bit to ACK even when the CRC on an **EV** packet has failed. This may be useful to save bandwidth or to save power.

Packets that have correct HEC, but that are addressed to other slaves, or packets other than **DH**, **DM**, **DV**, or **EV** packets shall not affect the ARQN bit, except as noted in 8.7.6.2.2. In these cases, the ARQN bit shall be left as it was prior to reception of the packet. For ACL packets, if a CRC packet with a correct header has the same SEQN as the previously received CRC packet, the ARQN bit shall be set to ACK, and the payload shall be ignored without checking the CRC. For eSCO packets, the SEQN shall not be used when determining the ARQN. If an eSCO packet has been received successfully within the eSCO window, subsequent receptions within the eSCO window shall be ignored. At the end of the eSCO window, the master's ARQN shall be retained for the first master-to-slave transmission in the next eSCO window.

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

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

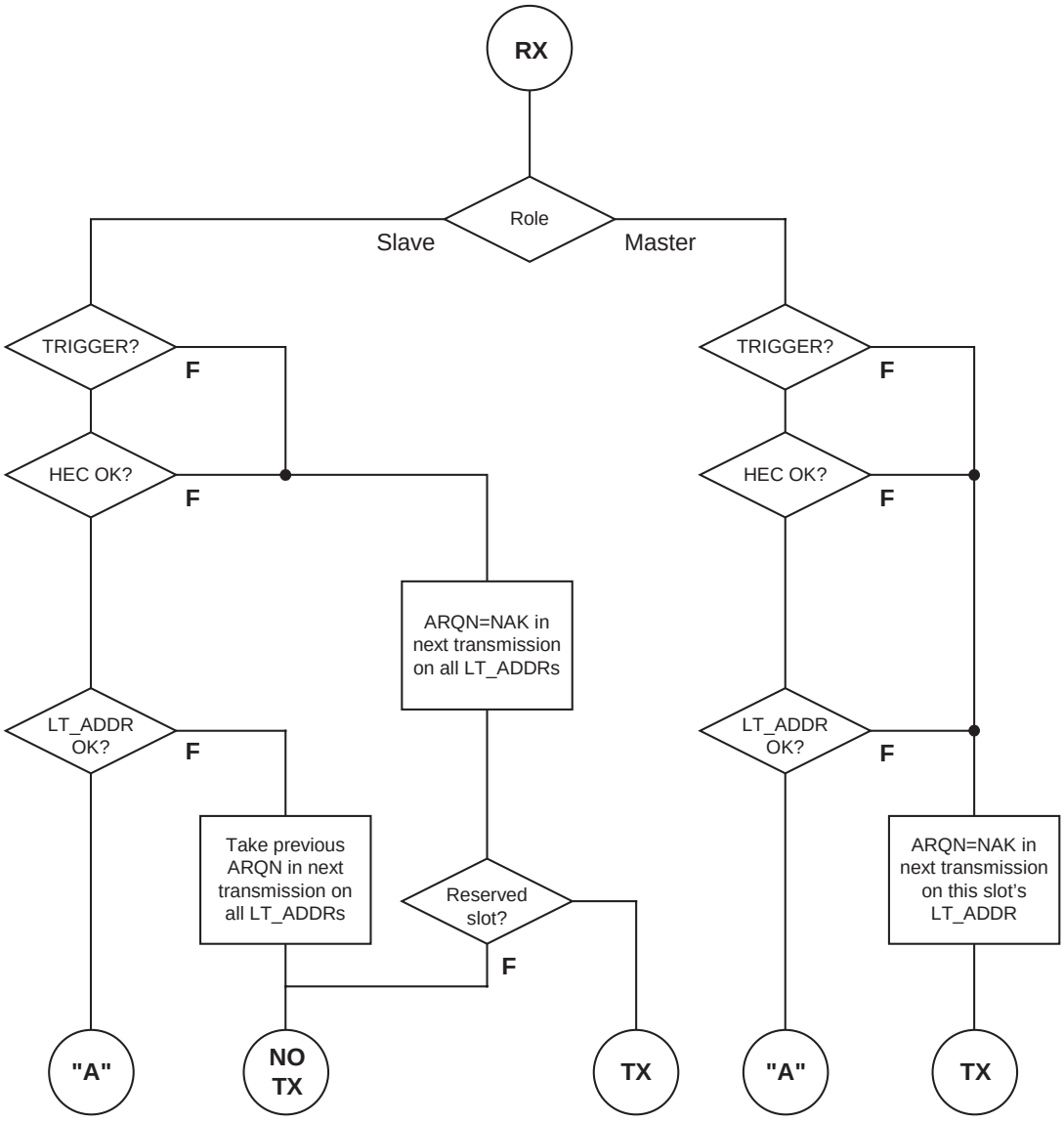


Figure 54—Stage 1 of the receive protocol for determining the ARQN bit

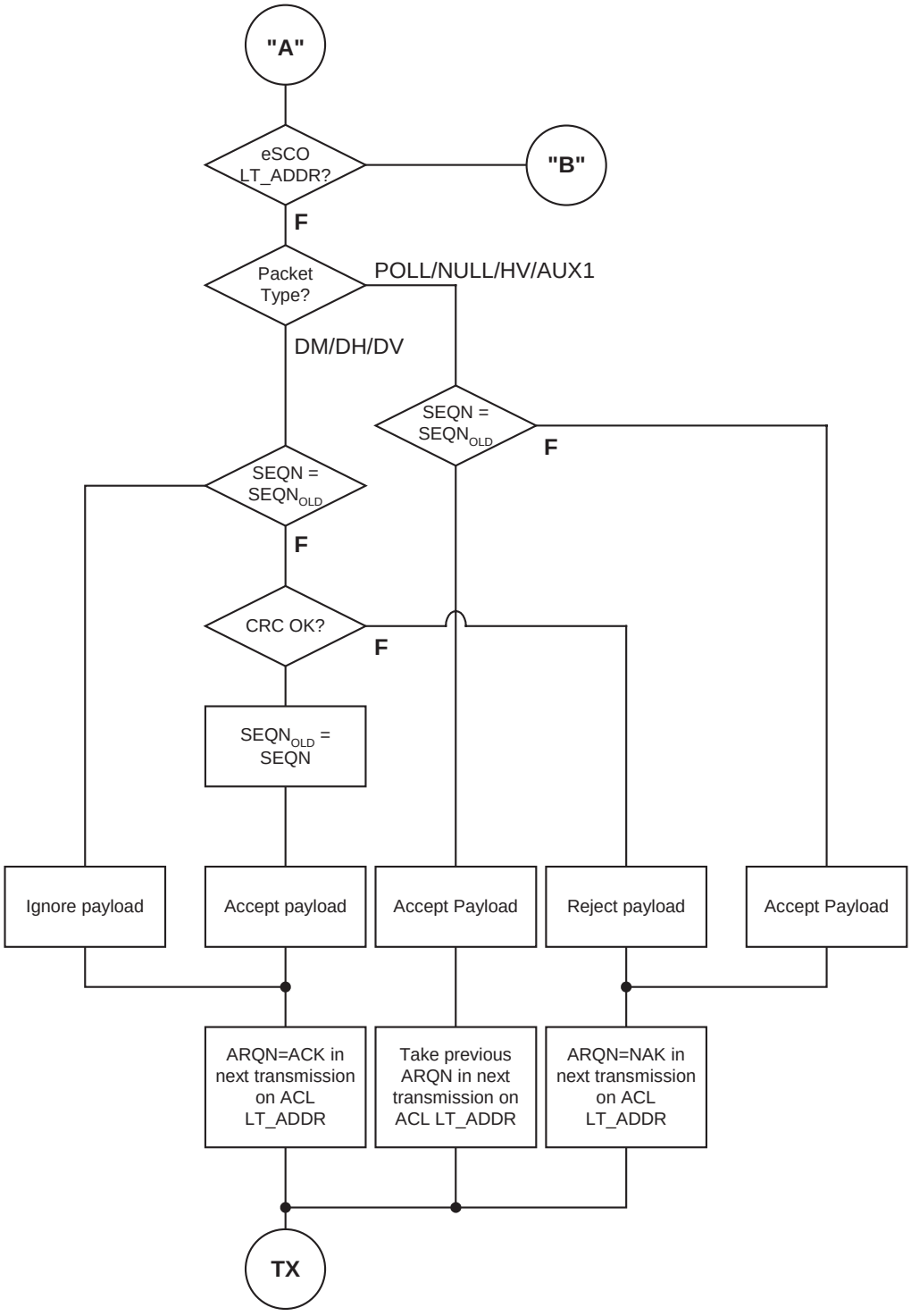


Figure 55—Stage 2 (ACL) of the receive protocol for determining the ARQN bit

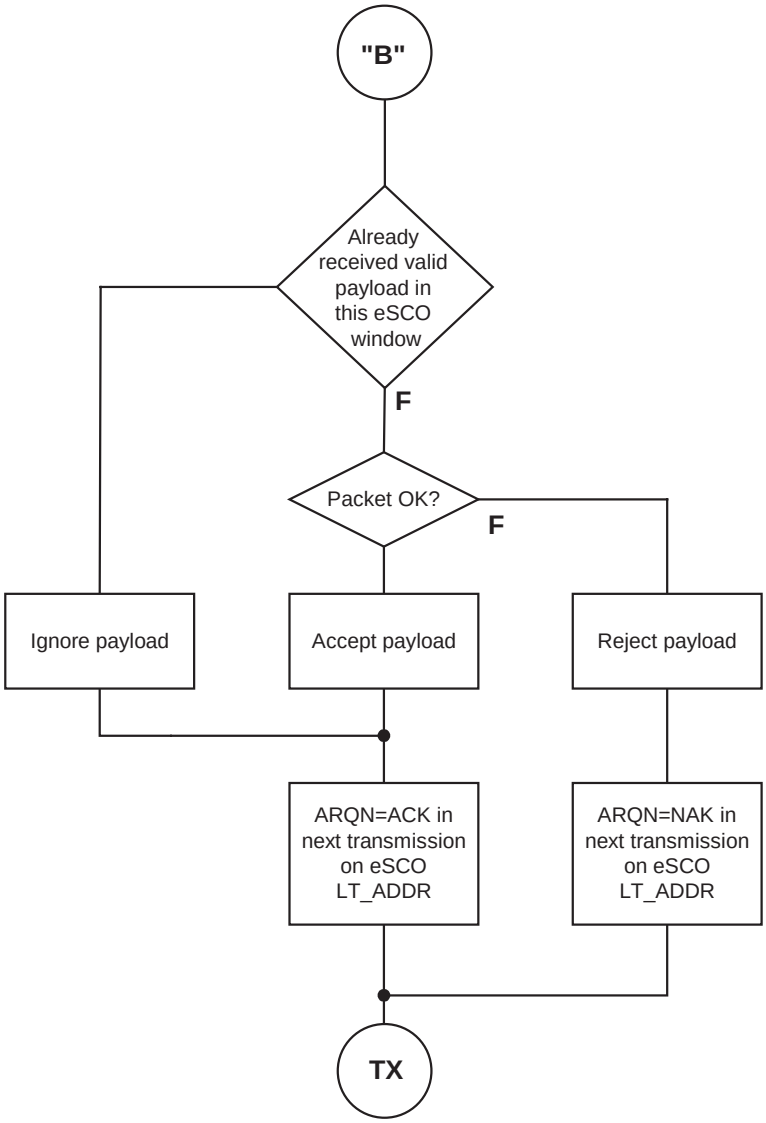


Figure 56—Stage 2 (eSCO) of the receive protocol for determining the ARQN bit

8.7.6.2 Retransmit filtering

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

8.7.6.2.1 Initialization of SEQN at start of new connection

The SEQN bit of the first CRC data packet at the start of a connection (as a result of page, page scan, role switch, or unpark) on both the master and the slave sides shall be set to 1. The subsequent packets shall use the rules in 8.7.6.2.2 through 8.7.6.2.5.

8.7.6.2.2 ACL and SCO retransmit filtering

The SEQN bit shall be affected only by the CRC data packets as shown in Figure 57. It shall be inverted every time a new CRC data packet is sent. The CRC data packet shall be retransmitted with the same SEQN number until an ACK is received or the packet is flushed. When an ACK is received, a new payload may be sent and on that transmission the SEQN bit shall be inverted. If a device decides to flush (see 8.7.6.3) and it has not received an acknowledgment for the current packet, it shall replace the current packet with an ACL-U continuation packet with the same SEQN as the current packet and length zero. If it replaces the current packet in this way, it shall not move on to transmit the next packet until it has received an ACK.

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

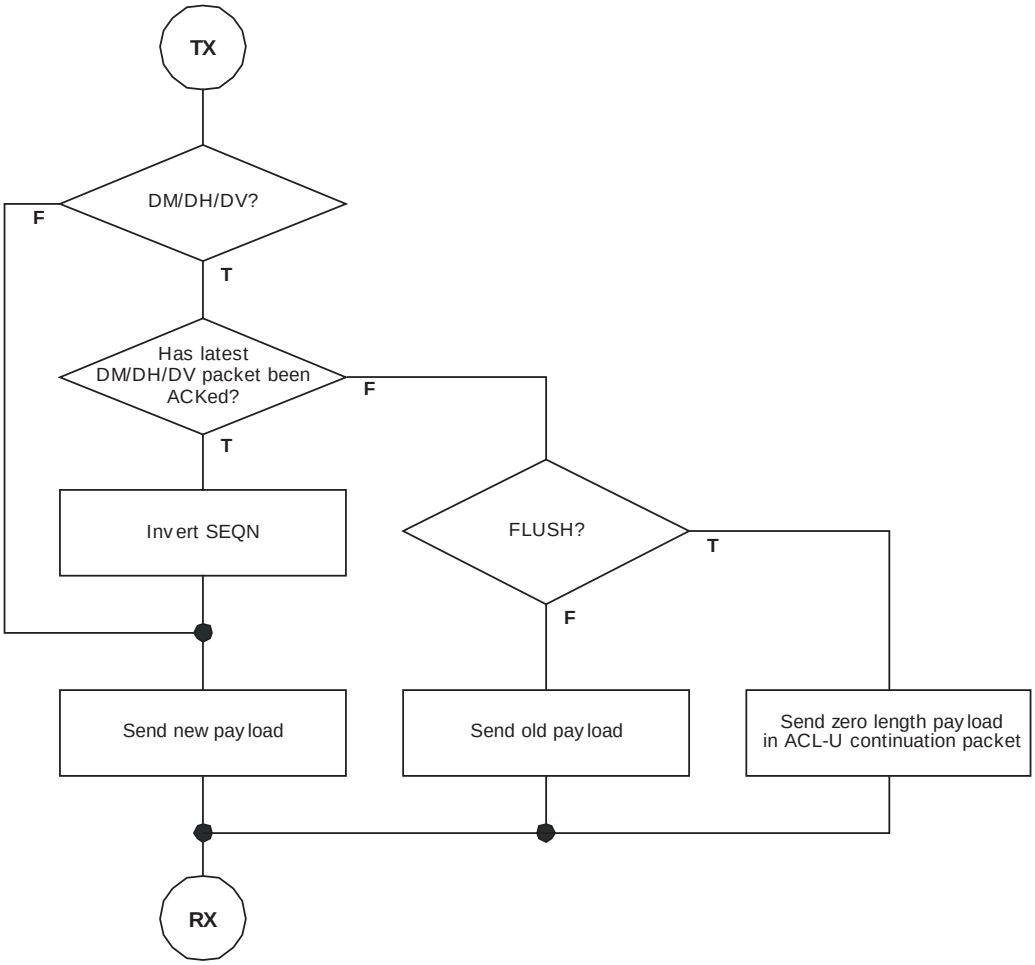


Figure 57—Transmit filtering for packets with CRC

8.7.6.2.3 eSCO retransmit filtering

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

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

8.7.6.2.4 FHS retransmit filtering

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

8.7.6.2.5 Packets without CRC retransmit filtering

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

8.7.6.3 Flushing payloads

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

Flushing results in loss of remaining portions of an L2CAP message. Therefore, the packet following the flush shall have a start packet indication of LLID = 10 in the payload header for the next L2CAP message. This informs the destination of the flush. (see 8.6.6). Flushing will not necessarily result in a change in the SEQN bit value (see 8.7.6.2).

The flush timeout defines a maximum period after which all segments of the ACL-U packet are flushed from the controller buffer. The flush timeout shall start when the first segment of the ACL-U packet is stored in the controller buffer. After the flush timeout has expired, the link controller may continue transmissions according to the procedure described in 8.7.6.2.2; however, the BB resource manager shall not continue the transmission of the ACL-U packet to the link controller. If the BB resource manager has further segments of the packet queued for transmission to the link controller, it shall delete the remaining segments of the ACL-U packet from the queue. In case the complete ACL-U packet was not stored in the controller buffer yet, any continuation segments, received for the ACL logical transport, shall be flushed until the first segment of the next ACL-U packet for the logical transport is received. When the complete ACL-U packet has been flushed, the LM shall continue transmission of the next ACL-U packet for the ACL logical transport. The default flush timeout shall be infinite, i.e., retransmissions are carried out until physical link loss occurs. This is also referred to as a *reliable channel*. All devices shall support the default flush timeout.

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

8.7.6.4 Multislave considerations

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

8.7.6.5 Broadcast packets

Broadcast packets are packets transmitted by the master to all the slaves simultaneously. If multiple hop sequences are being used, each transmission may be received only by some of the slaves. In this case the master shall repeat the transmission on each hop sequence. A broadcast packet shall be indicated by the all-zero LT_ADDR (note that the FHS packet is the only packet that may have an all-zero LT_ADDR, but is not a broadcast packet). Broadcast packets shall not be acknowledged.

Since broadcast messages are not acknowledged, each broadcast packet is transmitted at least a fixed number of times. A broadcast packet should be transmitted N_{BC} times before the next broadcast packet of the same broadcast message is transmitted (see Figure 58). Optionally, a broadcast packet may be transmitted $N_{BC} + 1$ times. Note that $N_{BC} = 1$ 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 8.8.9.5).

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.

Broadcast packets with a CRC shall have their own SEQN. The SEQN of the first broadcast packet with a CRC shall be set to 1 by the master and shall be inverted for each new broadcast packet with CRC thereafter. Broadcast packets without a CRC have no influence on the SEQN. 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 acknowledgment of broadcast messages and there is no end packet indication, it is important to receive the start packets correctly.[10] 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 8.6.6. Only repetitions of the L2CAP continuation packets shall be filtered out.

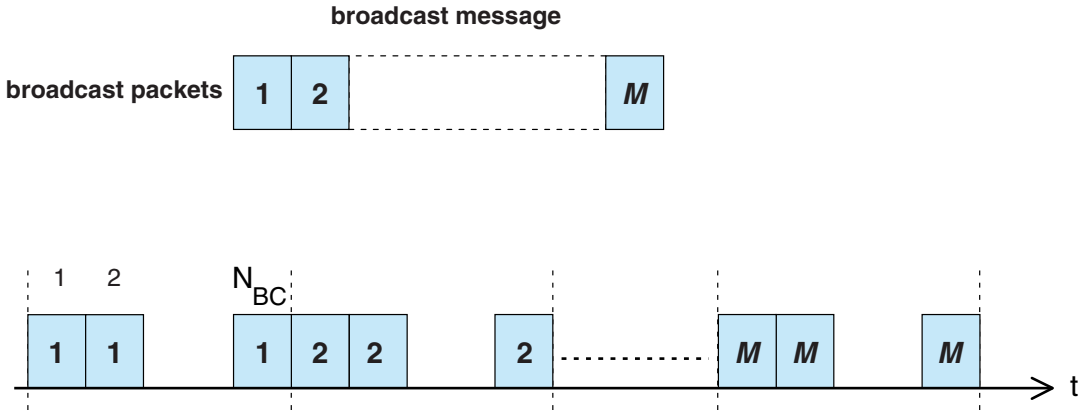


Figure 58—Broadcast repetition scheme

[10]Previous versions of the standard reset sequence numbering each time the first fragment of an ACL-U packet began broadcast. This meant that if an ACL-U packet fit into a single BB packet, it was impossible to distinguish it from the start of the next ACL-U packet. To avoid losing start fragments of ACL-U packets, the solution was to not filter out any start fragments of ACL-U packets. To keep behavior the same in all versions of the specification, this behavior has been retained.

8.8 Link controller operation

This subclause 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.

8.8.1 Overview of states

Figure 59 shows a state diagram illustrating the different states used in the link controller. There are three major states: STANDBY, CONNECTION, and PARK; in addition, there are seven substates, **page**, **page scan**, **inquiry**, **inquiry scan**, **master response**, **slave response**, and **inquiry response**. The substates are interim states that are used to establish connections and enable device discovery. To move from one state or substate to another, either commands from the LM are used, or internal signals in the link controller are used (such as the trigger signal from the correlator and the timeout signals). It should be noted that Figure 62 reflects the fact that a device may have more than one simultaneous connection and may be in different states with respect to each connection. So, for example, a single connection cannot move straight from the **inquiry** substate to the CONNECTION state, but a device that has an active connection may hold or sniff that connection and then move to the **inquiry** substate.

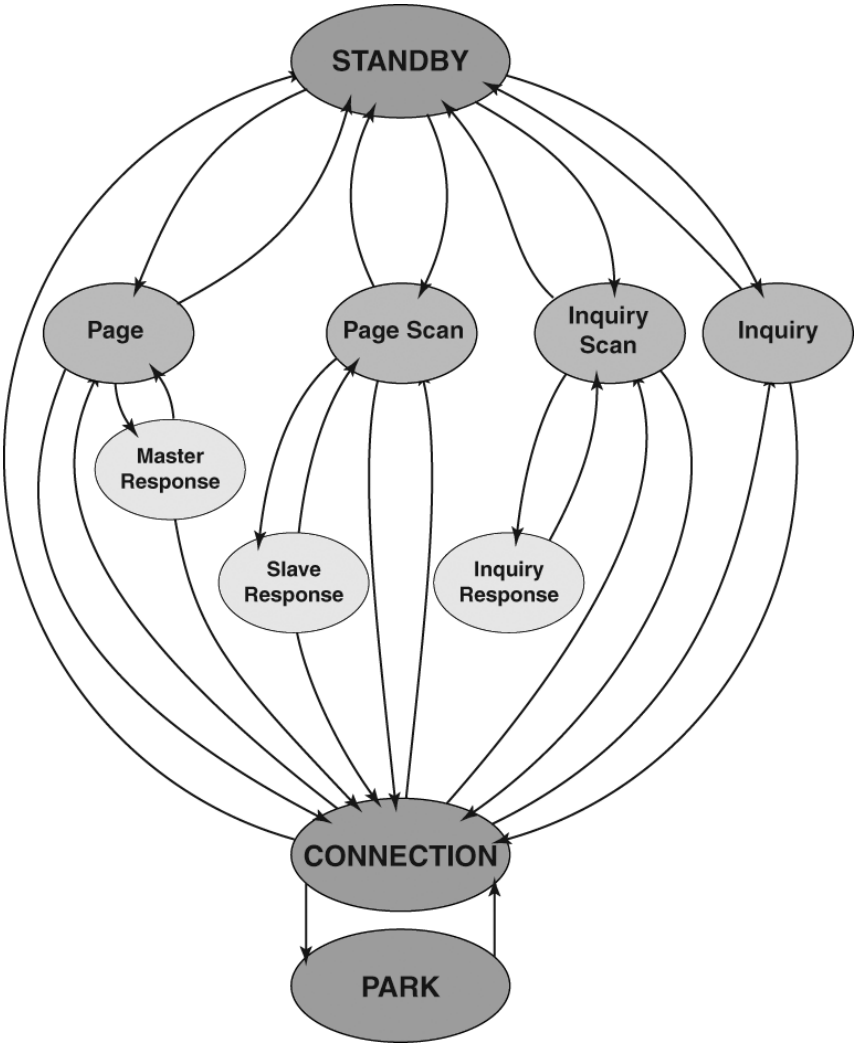


Figure 59—State diagram of link controller

Figure 60 shows a typical progress through the states for a single connection.

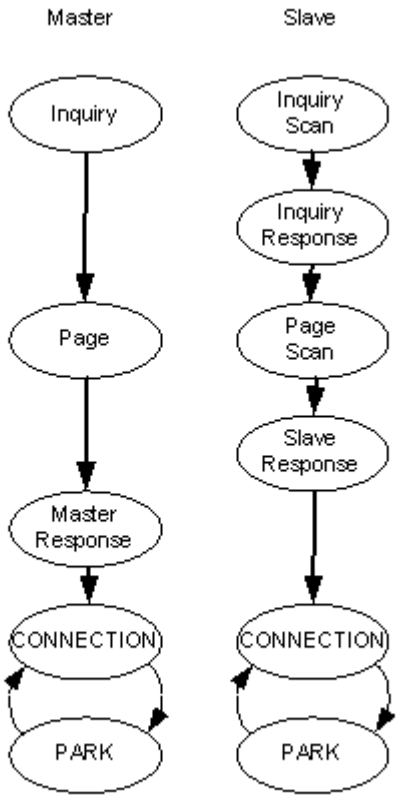


Figure 60—Typical progress though states

8.8.2 STANDBY state

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

The controller may leave the STANDBY state to scan for page or inquiry messages or to conduct a page or an inquiry.

8.8.3 Connection establishment substates

In order to establish new connections, the paging procedure is used. Only the device address is required to set up a connection. Knowledge about the clock, obtained from the inquiry procedure (see 8.8.4) or from a previous connection with this device, and the page scanning mode of the other device will accelerate the setup procedure. A device that establishes a connection carries out a page procedure and will automatically become the master of the connection.

8.8.3.1 Page scan substate

In the **page scan** substate, a device may be configured to use either the standard or interlaced scanning procedure. During a standard scan, a device listens for the duration of the scan window $T_{w\_page\_scan}$ (11.25 ms default) (see 11.7.3.20), while the interlaced scan is performed as two back-to-back scans of $T_{w\_page\_scan}$. If the scan interval is not at least twice the scan window, then interlaced scan shall not be used. During each scan window, the device shall listen at a single hop frequency, its correlator matched to its DAC. The scan window shall be long enough to completely scan 16 page frequencies.

When a device enters the **page scan** substate, it shall select the scan frequency according to the page hopping sequence determined by the device's device address (see 8.2.6.4.1). The phase in the sequence shall be determined by $CLKN_{16-12}$ of the device's CLKN, i.e., every 1.28 s a different frequency is selected.

In the case of a standard scan, if the correlator exceeds the trigger threshold during the page scan, the device shall enter the **slave response** substate described in 8.8.3.3.1. The scanning device may also use interlaced scan. In this case, if the correlator does not exceed the trigger threshold during the first scan, it shall scan a second time using the phase in the sequence determined by $[CLKN_{16-12} + 16] \bmod 32$. If, on this second scan, the correlator exceeds the trigger threshold, the device shall enter the **slave response** substate using $[CLKN_{16-12} + 16] \bmod 32$ as the frozen CLKN* in the calculation for $Xprs^{(79)}$ (see 8.2.6.4.3 for details). If the correlator does not exceed the trigger threshold during a scan in normal mode or during the second scan in interlaced scan mode, it shall return to either the STANDBY or CONNECTION state.

The **page scan** substate can be entered from the STANDBY state or the CONNECTION state. In the STANDBY state, no connection has been established, and the device can use all the capacity to carry out the page scan. Before entering the **page scan** substate from the CONNECTION state, the device should reserve as much capacity as possible for scanning. If desired, the device may place ACL connections in HOLD mode, PARK state, or SNIFF mode (see 8.8.8 and 8.8.9). Synchronous connections should not be interrupted by the page scan, although eSCO retransmissions should be paused during the scan. The page scan may be interrupted by the reserved synchronous slots, which should have higher priority than the page scan. SCO packets that require the least amount of capacity (**HV3** packets) should be used. The scan window shall be increased to minimize the setup delay. If one SCO logical transport is present using **HV3** packets and $T_{SCO} = 6$ slots or if one eSCO logical transport is present using **EV3** packets and $T_{ESCO} = 6$ slots, a total scan window $T_{w\_page\_scan}$ of at least 36 slots (22.5 ms) is recommended. If two SCO links are present using **HV3** packets and $T_{SCO} = 6$ slots or if two eSCO links are present using **EV3** packets and $T_{ESCO} = 6$ slots, a total scan window of at least 54 slots (33.75 ms) is recommended.

The scan interval $T_{page\ scan}$ is defined as the interval between the beginnings of two consecutive page scans. A distinction is made between the case where the scan interval is equal to the scan window $T_{w\_page\_scan}$ (continuous scan), the scan interval is maximal 1.28 s, or the scan interval is maximal 2.56 s. These three cases shall determine the behavior of the paging device, i.e., whether the paging device shall use R0, R1, or R2 (see also 8.8.3.2). Table 24 illustrates the relationship between $T_{page\ scan}$ and modes R0, R1, and R2. Although scanning in the R0 mode is continuous, the scanning may be interrupted, for example, by reserved synchronous slots. The scan interval information is included in the SR field in the FHS packet.

Table 24—Relationship between scan interval and paging modes R0, R1, and R2

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

8.8.3.2 Page substate

The **page** substate is used by the master (source) to activate and connect to a slave (destination) in the **page scan** substate. The master tries to coincide with the slave's scan activity by repeatedly transmitting the paging message consisting of the slave's DAC in different hop channels. Since the clocks of the master and the slave are not synchronized, the master does not know exactly when the slave wakes up and on which hop

frequency. Therefore, it transmits a train of identical paging messages at different hop frequencies and listens between the transmit intervals until it receives a response from the slave.

The page procedure in the master consists of a number of steps. On systems with separate host and controller, the host first communicates the BD_ADDR of the slave to the controller. This BD_ADDR shall be used by the master to determine the page hopping sequence (see 8.2.6.4.2). The slave's BD_ADDR shall be used to determine the page hopping sequence (see 8.2.6.4.2). For the phase in the sequence, the master shall use an estimate of the slave's clock. For example, this estimate can be derived from timing information that was exchanged during the last encounter with this particular device (which could have acted as a master at that time) or from an inquiry procedure. With this CLKE of the slave's CLKN, the master can predict on which hop channel the slave starts page scanning.

The estimate of the clock in the slave can be completely wrong. Although the master and the slave use the same hopping sequence, they use different phases in the sequence and might never select the same frequency. To compensate for the clock drifts, the master shall send its page message during a short time interval on a number of wake-up frequencies. It shall transmit also on hop frequencies just before and after the current, predicted hop frequency. During each TX slot, the master shall sequentially transmit on two different hop frequencies. In the following RX slot, the receiver shall listen sequentially to two corresponding RX hops for an ID packet. The RX hops shall be selected according to the page response hopping sequence. The page response hopping sequence is strictly related to the page hopping sequence: for each page hop there is a corresponding page response hop. The RX/TX timing in the **page** substate is described in 8.2.2.5 (see also Figure 18). In the next TX slot, it shall transmit on two hop frequencies different from the former ones. The hop rate is increased to 3200 hop/s.

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

$$f(k-8), f(k-7), \ldots, f(k), \ldots, f(k+7)$$

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

When the difference between the clocks of the master and the slave is less than -8×1.28 s or larger than $+7 \times 1.28$ s, the remaining 16 hops are used to form the new 10 ms train **B.** The second train consists of hops

$$f(k-16), f(k-15), \ldots, f(k-9), f(k+8) \ldots, f(k+15)$$

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

The **page** substate may be entered from the STANDBY state or the CONNECTION state. In the STANDBY state, no connection has been established, and the device can use all the capacity to carry out the page. Before entering the **page** substate from the CONNECTION state, the device should free as much capacity as possible for scanning. To ensure this, it is recommended that the ACL connections are put in HOLD mode or PARK state. However, the synchronous connections shall not be disturbed by the page. This means that the

page will be interrupted by the reserved SCO and eSCO slots, which have higher priority than the page. In order to obtain as much capacity for paging, it is recommended to use the SCO packets that use the least amount of capacity (**HV3** packets). If SCO or eSCO links are present, the repetition number N_{page} of a single train shall be increased (see Table 25). Here it has been assumed that the **HV3** packets are used with an interval T_{SCO} = 6 slots or **EV3** packets are used with an interval of T_{ESCO} = 6 slots, which would correspond to a 64 kb/s synchronous link.

Table 25—Relationship between train repetition and paging modes R0, R1, and R2 when synchronous links are present

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

The construction of the page train shall be independent of the presence of synchronous links. In other words, synchronous packets are sent on the reserved slots, but shall not affect the hop frequencies used in the unreserved slots (see Figure 61).

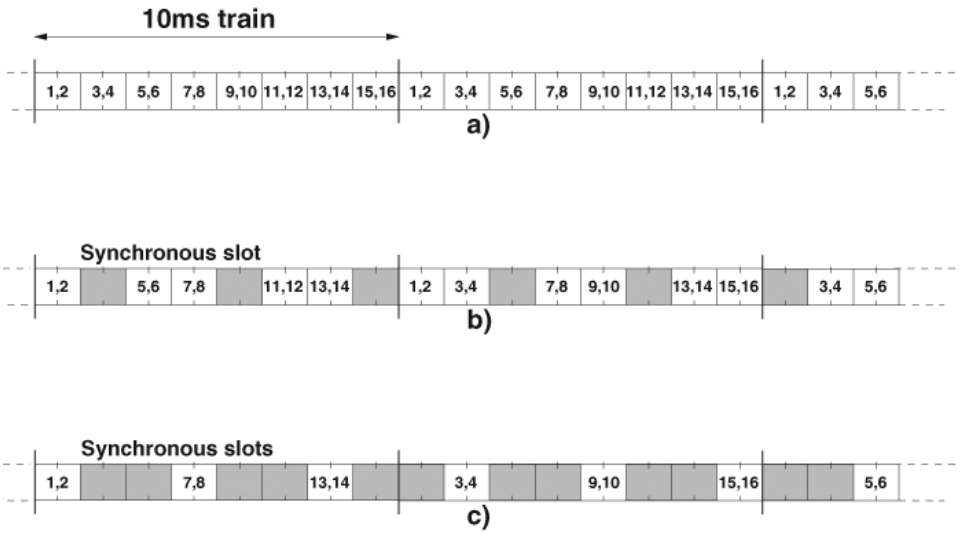


Figure 61—Conventional page (a), page while one synchronous link present (b), page while two synchronous links present (c)

8.8.3.3 Page response substates

When a page message is successfully received by the slave, there is a coarse frequency hopping synchronization between the master and the slave. Both the master and the slave enter a response substate to exchange vital information to continue the connection setup. It is important for the piconet connection that both devices shall use the same CAC, that they shall use the same channel hopping sequence, and that their clocks are synchronized. These parameters shall be derived from the master device. The device that initializes the connection (starts paging) is defined as the master device (which is thus valid only during the time the piconet exists). The CAC and channel hopping sequence shall be derived from the device address (BD_ADDR)

of the master. The timing shall be determined by CLK. An offset shall be added to the slave's CLNE to temporarily synchronize the slave clock to the CLK. At startup, the master parameters are transmitted from the master to the slave. The messaging between the master and the slave at startup is specified in this subclause.

The initial messaging between master and slave is shown in Table 26 and in Figure 62 and Figure 63. In those two figures, frequencies $f(k)$, $f(k + 1)$, etc., are the frequencies of the page hopping sequence determined by the slave's BD_ADDR. The frequencies $f'(k)$, $f'(k + 1)$, etc., are the corresponding page response frequencies (slave-to-master). The frequencies $g(m)$ belong to the basic channel hopping sequence.

Table 26—Initial messaging during startup

Step	Message	Packet type	Direction	Hopping sequence	Access code and clock
1	Page	ID	Master to slave	Page	Slave
2	First slave page response	ID	Slave to master	Page response	Slave
3	Master page response	FHS	Master to slave	Page	Slave
4	Second slave page response	ID	Slave to master	Page response	Slave
5	1st packet master	POLL	Master to slave	Channel	Master
6	1st packet slave	Any type	Slave to master	Channel	Master

In step 1 (see Table 26), the master device is in **page** substate, and the slave device in the **page scan** substate. Assume in this step that the page message sent by the master reaches the slave. On receiving the page message, the slave enters the **slave response** substate in step 2. The master waits for a reply from the slave, and when this arrives in step 2, it will enter the **master response** substate in step 3. Note that during the initial message exchange, all parameters are derived from the slave's device address and that only the page hopping and page response hopping sequences are used (these are also derived from the slave's device address). Note that when the master and slave enter the response substates, their clock input to the page and page response hop selection is frozen as is described in 8.2.6.4.3.

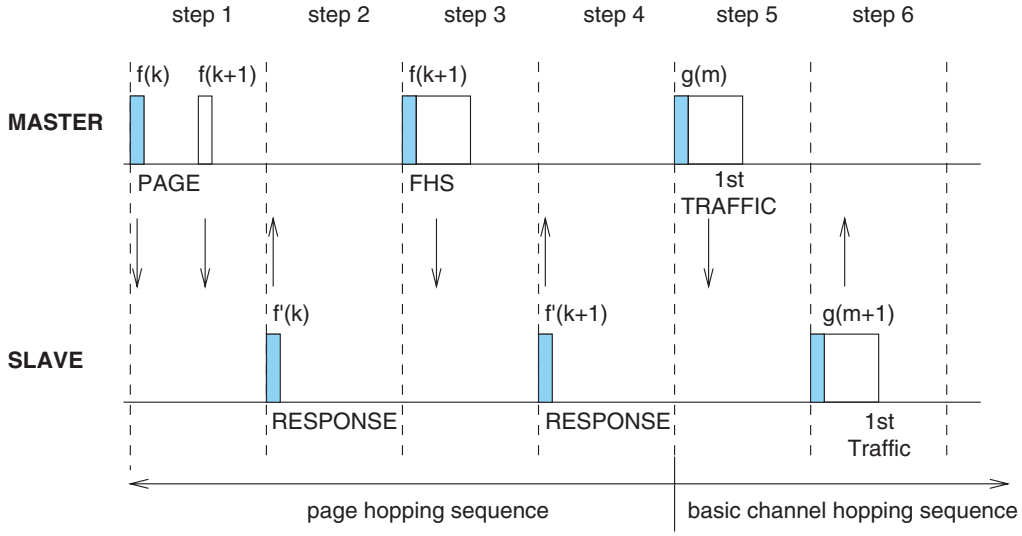


Figure 62—Messaging at initial connection when slave responds to first page message

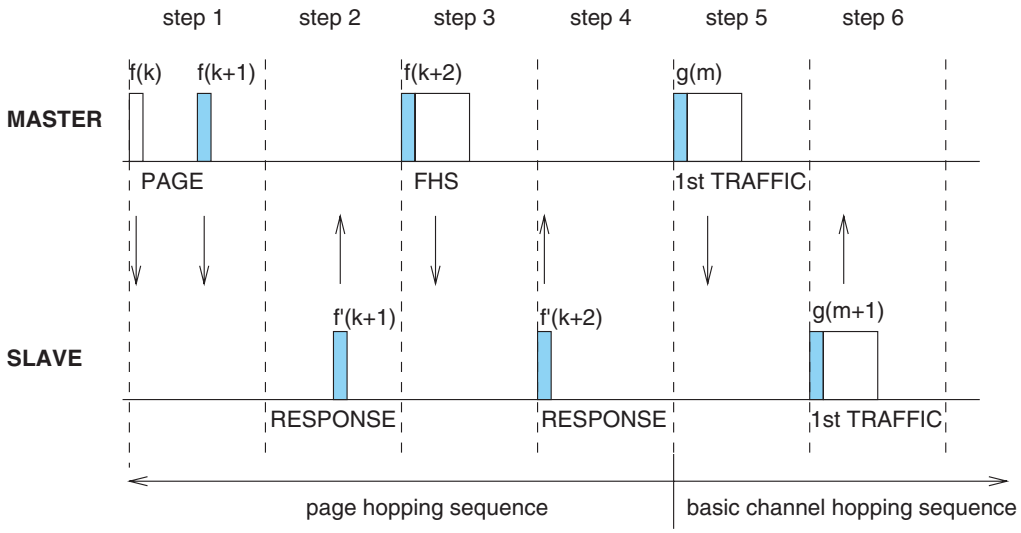


Figure 63—Messaging at initial connection when slave responds to second page message

8.8.3.3.1 Slave response substate

After having received the page message in step 1, the slave device shall transmit a slave page response message (the slave's DAC) in step 2. This response message shall be the slave's DAC. The slave shall transmit this response 625 μs after the beginning of the received page message and at the response hop frequency that corresponds to the hop frequency in which the page message was received. The slave transmission is, therefore, time aligned to the master transmission. During initial messaging, the slave shall still use the page response hopping sequence to return information to the master. The clock input $CLKN_{16-12}$ shall be frozen at the value it had at the time the page message was received.

After having sent the response message, the slave's receiver shall be activated 312.5 μs after the start of the response message and shall await the arrival of an FHS packet. Note that an FHS packet can arrive 312.5 μs after the arrival of the page message as shown in Figure 63 and not after 625 μs as is usually the case in the piconet physical channel RX/TX timing. More details about the timing can be found in 8.2.4.4.

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

If an FHS packet is received by the slave in the **slave response** substate, the slave shall return a slave page response message in step 4 to acknowledge reception of the FHS packet. This response shall use the page response hopping sequence. The transmission of the slave page response packet is based on the reception of the FHS packet. Then the slave shall change to the master's CAC and clock as received from the FHS packet. Only the 26 MSBs of the CLK are transferred: the timing shall be such that CLK_1 and CLK_0 are both zero at the time the FHS packet was received as the master transmits in even slots only. The offset between the master's clock and the slave's clock shall be determined from the master's clock in the FHS packet and reported to the slave's BB resource manager.

Finally, the slave enters the CONNECTION state in step 5. From then on, the slave shall use the master's clock and the master's BD_ADDR to determine the basic channel hopping sequence and the CAC. The slave

shall use the LT_ADDR in the FHS payload as the primary LT_ADDR in the CONNECTION state. The connection mode shall start with a POLL packet transmitted by the master. The slave may respond with any type of packet. If the POLL packet is not received by the slave or the response packet is not received by the master within *newconnectionTO* number of slots after FHS packet acknowledgment, the master and the slave shall return to **page** and **page scan** substates, respectively. See 8.8.5.

8.8.3.3.2 Master response substate

When the master has received a slave page response message in step 2, it shall enter the master response routine. It shall freeze the current clock input to the page hop selection scheme. The master shall then transmit an FHS packet in step 3 containing the master's CLKN, the master's BD_ADDR, the BCH parity bits, and the class of device. The FHS packet contains all information to construct the CAC without requiring a mathematical derivation from the master's 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 packet 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 as shown in Figure 63 and not 625 μs after the received packet as is usual in the piconet physical channel RX/TX timing (see also 8.2.4.4).

After the master has sent its FHS packet, it shall wait for a second slave page response message in step 4 acknowledging the reception of the FHS packet. This response shall be the slave's DAC. If no response is received, the master shall retransmit the FHS packet with an updated clock and still using the slave's parameters. It shall retransmit the FHS packet with the clock updated each time until a second slave page response message is received or the timeout of *pageRespTO* is exceeded. In the latter case, the master shall return to the **page** substate and send an error message to the BB resource manager. During the retransmissions of the FHS packet, the master shall use the page hopping sequence.

If the slave's response is received, the master shall change to using the master parameters, so it shall use the CAC and the CLK. The FHS packet transmission shall start when the lower clock bits CLK0 and CLK1 are reset to zero. These clock bits are not included in the FHS packet. Finally, the master enters the CONNECTION state in step 5. The master BD_ADDR shall be used to change to a new hopping sequence, the *basic channel hopping sequence*. The basic channel hopping sequence uses all 79 hop channels in a pseudo-random fashion (see also 8.2.6.4.7). The master shall now send its first traffic packet in a hop determined with the new (master) parameters. This first packet shall be a POLL packet. See 8.8.5. This packet shall be sent within *newconnectionTO* number of slots after reception of the FHS packet acknowledgment. The slave may respond with any type of packet. If the POLL packet is not received by the slave or the POLL packet response is not received by the master within *newconnectionTO* number of slots, the master and the slave shall return to **page** and **page scan** substates, respectively.

8.8.4 Device discovery substates

In order to discover other devices, a device shall enter the **inquiry** substate. In this substate, it shall repeatedly transmit the inquiry message (ID packet) (see 8.6.5.1.1) at different hop frequencies. The inquiry hop sequence is derived from the LAP of the GIAC. Thus, even when DIACs are used, the applied hopping sequence is generated from the GIAC LAP. A device that allows itself to be discovered should regularly enter the **inquiry scan** substate to respond to inquiry messages. The message exchange and contention resolution during inquiry response are described in 8.8.4.1 through 8.8.4.3. The inquiry response is optional: a device is not forced to respond to an inquiry message.

During the **inquiry** substate, the discovering device collects the device addresses and clocks of all devices that respond to the inquiry message. It may then, if desired, make a connection to any one of them by means of the page procedure described in 8.8.3.

The inquiry message broadcast by the source does not contain any information about the source. However, it may indicate which classes of device should respond. There is one GIAC to inquire for any device and a number of DIACs that inquire only for a certain type of device. The IACs are derived from reserved device addresses and are further described in 8.6.3.1.

8.8.4.1 Inquiry scan substate

The **inquiry scan** substate is very similar to the **page scan** substate. However, instead of scanning for the device's DAC, the receiver shall scan for the IAC long enough to completely scan for 16 inquiry frequencies. Two types of scans are defined: standard and interlaced. In the case of a standard scan, the length of this scan period is denoted $T_{w\_inquiry\_scan}$ (11.25 ms default) (see 11.7.3.22). The standard scan is performed at a single hop frequency as defined by Xir_{4-0} (see 8.2.6.4.6). The interlaced scan is performed as two back-to-back scans of $T_{w\_inquiry\_scan}$ where the first scan is on the normal hop frequency and the second scan is defined by $[Xir_{4-0} + 16] \bmod 32$. If the scan interval is not at least twice the scan window, then interlaced scan shall not be used. The inquiry procedure uses 32 dedicated inquiry hop frequencies according to the inquiry hopping sequence. These frequencies are determined by the general inquiry address. The phase is determined by CLKN of the device carrying out the inquiry scan; the phase changes every 1.28 s.

Instead of, or in addition to, the GIAC, the device may scan for one or more DIACs. However, the scanning shall follow the inquiry scan hopping sequence determined by the general inquiry address. If an inquiry message is received during an inquiry wake-up period, the device shall enter the **inquiry response** substate.

The **inquiry scan** substate can be entered from the STANDBY state or the CONNECTION state. In the STANDBY state, no connection has been established, and the device can use all the capacity to carry out the inquiry scan. Before entering the **inquiry scan** substate from the CONNECTION state, the device should reserve as much capacity as possible for scanning. If desired, the device may place ACL logical transports in SNIFF mode, HOLD mode, or PARK state. Synchronous logical transports are preferably not interrupted by the inquiry scan, although eSCO retransmissions should be paused during the scan. In this case, the inquiry scan may be interrupted by the reserved synchronous slots. SCO packets that require the least amount of capacity (**HV3** packets) should be used. The scan window $T_{w\_inquiry\_scan}$ shall be increased to increase the probability to respond to an inquiry message. If one SCO logical transport is present using **HV3** packets and $T_{SCO} = 6$ slots or if one eSCO logical transport is present using **EV3** packets and $T_{ESCO} = 6$ slots, a total scan window of at least 36 slots (22.5 ms) is recommended. If two SCO links are present using **HV3** packets and $T_{SCO} = 6$ slots or if two eSCO links are present using **EV3** packets and $T_{ESCO} = 6$ slots, a total scan window of at least 54 slots (33.75 ms) is recommended.

The scan interval $T_{inquiry\_scan}$ is defined as the interval between two consecutive inquiry scans. The inquiry scan interval shall be less than or equal to 2.56 s.

8.8.4.2 Inquiry substate

The **inquiry** substate is used to discover new devices. This substate is very similar to the **page** substate; the TX/RX timing shall be the same as in paging (see 8.2.4.4 and Figure 18). The TX and RX frequencies shall follow the inquiry hopping sequence and the inquiry response hopping sequence and are determined by the GIAC and the CLKN of the discovering device. Between inquiry transmissions, the receiver shall scan for inquiry response messages. When a response is received, the entire packet (an FHS packet) is read, after which the device shall continue with inquiry transmissions. The device in an **inquiry** substate shall not acknowledge the inquiry response messages. If enabled by the host (see 11.7.3.54), the RSSI value of the inquiry response message shall be measured. The device shall keep probing at different hop channels and between, listening for response packets. As in the **page** substate, two 10 ms trains, **A** and **B**, are defined, splitting the 32 frequencies of the inquiry hopping sequence into two 16-hop parts. A single train shall be repeated for at least $N_{inquiry} = 256$ times before a new train is used, except for the first train in a series, which may be repeated fewer times. In order to collect all responses in an error-free environment, at least three train switches must have taken place. As a result, the **inquiry** substate may have to last for 10.24 s unless the

inquirer collects enough responses and aborts the **inquiry** substate earlier. If desired, the inquirer may also prolong the **inquiry** substate to increase the probability of receiving all responses in an error-prone environment. If an inquiry procedure is automatically initiated periodically (e.g., a 10 s period every minute), then the interval between two inquiry instances shall be determined randomly. This is done to avoid two devices synchronizing their inquiry procedures.

The **inquiry** substate is continued until stopped by the BB resource manager (when it decides that it has sufficient number of responses), when a timeout has been reached (*inquiryTO*), or by a command from the host to cancel the inquiry procedure.

The **inquiry** substate can be entered from the STANDBY state or the CONNECTION state. In the STANDBY state, no connection has been established, and the device can use all the capacity to carry out the inquiry. Before entering the **inquiry** substate from the CONNECTION state, the device should free as much capacity as possible for scanning. To ensure this, it is recommended that the ACL logical transports are placed in SNIFF mode, HOLD mode, or PARK state. However, the reserved slots of synchronous logical transports shall not be disturbed by the inquiry. This means that the inquiry will be interrupted by the reserved SCO and eSCO slots, which have higher priority than the inquiry. In order to obtain as much capacity as possible for inquiry, it is recommended to use the SCO packets that use the least amount of capacity (**HV3** packets). If SCO or eSCO links are present, the repetition number N_{inquiry} shall be increased (see Table 27). Here it has been assumed that **HV3** packets are used with an interval $T_{\text{SCO}} = 6$ slots or **EV3** packets are used with an interval of $T_{\text{ESCO}} = 6$ slots, which would correspond to a 64 kb/s synchronous link.

Table 27—Increase of train repetition when synchronous links are present

	No synchronous links	One synchronous link (HV3)	Two synchronous links (HV3)
N_{inquiry}	≥ 256	≥ 512	≥ 768

8.8.4.3 Inquiry response substate

The **slave response** substate for inquiries differs completely from the **slave response** substate applied for pages. When the inquiry message is received in the **inquiry scan** substate, the recipient shall return an inquiry response (FHS) packet containing the recipient's device address (BD_ADDR) and other parameters.

The following protocol in the slave's inquiry response shall be used. On the first inquiry message received in this substate, the slave shall enter the **inquiry response** substate and shall return an FHS response packet to the master 625 μs after the inquiry message was received. A contention problem may arise when several devices are in close proximity to the inquiring device and all respond to an inquiry message at the same time. However, because every device has a free-running clock, it is highly unlikely that they all use the same phase of the inquiry hopping sequence. In order to avoid repeated collisions between devices that wake up in the same inquiry hop channel simultaneously, a device shall back off for a random period of time. Thus, if the device receives an inquiry message and returns an FHS packet, it shall generate a random number, RAND, between 0 and MAX_RAND. For scanning intervals ≥ 1.28 s, MAX_RAND shall be 1023; however, for scanning intervals < 1.28 s, MAX_RAND may be as small as 127. A profile that uses a special DIAC may choose to use a smaller MAX_RAND than 1023 even when the scanning interval is ≥ 1.28 s. The slave shall return to the CONNECTION or STANDBY state for the duration of at least RAND time slots. Before returning to the CONNECTION and STANDBY state, the device may go through the **page scan** substate. After at least RAND slots, the device shall add an offset of 1 to the phase in the inquiry hop sequence (the phase has a 1.28 s resolution) and return to the **inquiry scan** substate again. If the slave is triggered again, it shall repeat the procedure using a new random number. The offset to the clock accumulates each time an FHS packet is returned. During a probing window, a slave may respond multiple times, but on

different frequencies and at different times. Reserved synchronous slots should have priority over response packets. In other words, if a response packet overlaps with a reserved synchronous slot, it shall not be sent, but the next inquiry message is awaited.

The messaging during the inquiry routines is summarized in Table 28. In step 1, the master transmits an inquiry message using the IAC and its own clock. The slave responds with the FHS packet containing the slave's device address, CLKN, and other slave information. This FHS packet is returned at times that tend to be random. The FHS packet is not acknowledged in the inquiry routine, but it is retransmitted at other times and frequencies as long as the master is probing with inquiry messages.

Table 28—Messaging during inquiry routines

Step	Message	Packet type	Direction	Hopping sequence	Access code and clock
1	Inquiry	ID	Master to slave	Inquiry	Inquiry
2	Inquiry response	FHS	Slave to master	Inquiry response	Inquiry

8.8.5 Connection state

In the CONNECTION state, the connection has been established, and packets can be sent back and forth. In both devices, the CAC (master), the CLK, and the AFH_channel_map are used. The CONNECTION state uses the basic or adapted channel hopping sequence.

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

The first information packets in the CONNECTION state contain control messages that characterize the link and give more details regarding the devices. These messages are exchanged between the LMs 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 11.7.3.2.

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 64).

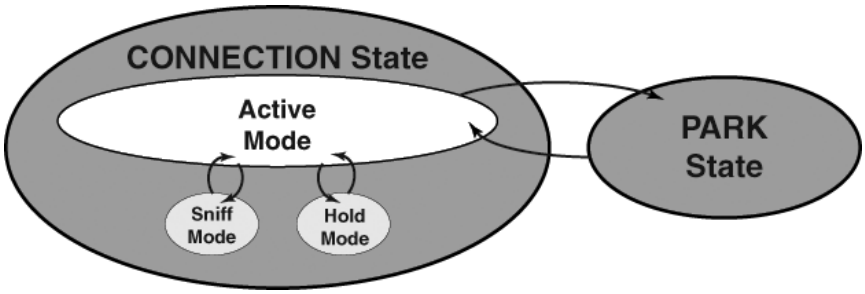


Figure 64—Connection state

8.8.6 Active mode

In the active mode, both master and slave actively participate on the channel. Up to seven slaves may be in the active mode at any given time. The master schedules the transmissions based on traffic demands to and from the different slaves. In addition, it supports regular transmissions to keep slaves synchronized to the channel. Slaves in the active mode listen in the master-to-slave slots for packets. These devices are known as *active slaves*. If an active slave is not addressed, it may sleep until the next new master transmission. Slaves can derive the number of slots the master has reserved for transmission from the TYPE field in the packet header; during this time, the nonaddressed slaves do not have to listen on the master-to-slave slots. When a device is participating in multiple piconets, it should listen in the master-to-slave slot for the current piconet. It is recommended that a device not be away from each piconet in which it is participating for more than T_{poll} slots. A periodic master transmission is required to keep the slaves synchronized to the channel. Since the slaves need only the CAC to synchronize, any packet type can be used for this purpose.

Only the slave that is addressed by one of its LT_ADDRs (primary or secondary) may return a packet in the next slave-to-master slot. If no valid packet header is received, the slave may respond only in its reserved SCO or eSCO slave-to-master slot. In the case of a broadcast message, no slave shall return a packet (an exception is the access window for access requests in the PARK state; see 8.8.9).

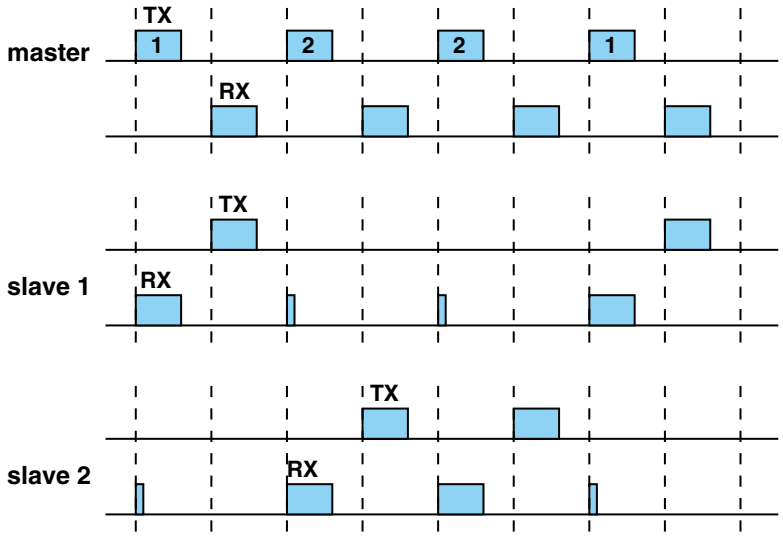


Figure 65—RX/TX timing in multislave configuration

8.8.6.1 Polling in the active mode

The master always has full control over the piconet. Due to the TDD scheme, slaves can communicate only with the master and not with other slaves. In order to avoid collisions on the ACL logical transport, a slave is allowed to transmit in the slave-to-master slot only when addressed by the LT_ADDR in the packet header in the preceding master-to-slave slot. If the LT_ADDR in the preceding slot does not match or a valid packet header was not received, the slave shall not transmit unless the slot is reserved for the slave on a synchronous logical transport.

The master normally attempts to poll a slave's ACL logical transport no less often than once every T_{poll} slots. T_{poll} is set by the LM (see 9.3.1.8).

The slave's ACL logical transport may be polled with any packet type except for FHS and ID. For example, polling during SCO may use **HV** packets, since the slave may respond to an **HV** packet with a **DM1** packet (see 8.8.6.2).

8.8.6.2 SCO

The SCO logical transport shall be established by the master sending an SCO setup message via the LMP. This message contains timing parameters including the SCO interval T_{SCO} and the offset D_{SCO} to specify the reserved slots.

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

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

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

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

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

The master will send SCO packets to the slave at regular intervals (the SCO interval T_{SCO} counted in slots) in the reserved master-to-slave slots. An **HV1** packet can carry 1.25 ms of speech at a 64 kb/s rate. An **HV1** packet shall, therefore, be sent every two time slots ($T_{SCO} = 2$). An **HV2** packet can carry 2.5 ms of speech at a 64 kb/s rate. An **HV2** packet shall, therefore, be sent every four time slots ($T_{SCO} = 4$). An **HV3** packet can carries 3.75 ms of speech at a 64 kb/s rate. An **HV3** packet shall, therefore, be sent every six time slots ($T_{SCO} = 6$).

The slave is allowed to transmit in the slot reserved for its SCO logical transport unless the (valid) LT_ADDR in the preceding slot indicates a different slave. If no valid LT_ADDR can be derived in the preceding slot, the slave may still transmit in the reserved SCO slot.

Since the **DM1** packet is recognized on the SCO logical transport, it may be sent during the SCO reserved slots if a valid packet header with the primary LT_ADDR is received in the preceeding slot. **DM1** packets sent during SCO reserved slots shall be used only to send ACL-C data.

The slave shall not transmit anything other than an **HV** packet in a reserved SCO slot unless it decodes its own slave address in the packet header of the packet in the preceding master-to-slave transmission slot.

8.8.6.3 eSCO

The eSCO logical transport is established by the master sending an eSCO setup message via the LMP. 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 LMP. This message shall contain the eSCO interval T_{ESCO} and an offset D_{ESCO}. In order to prevent clock wraparound problems, an

initialization flag in the LMP setup message indicates whether initialization procedure 1 or 2 shall be used. The slave shall apply the initialization method as indicated by the initialization flag. The master shall use initialization 1 when the MSB of the current master clock CLK_{27} is 0; it shall use initialization 2 when the MSB of the current master clock CLK_{27} is 1. The master-to-slave eSCO slots reserved by the master and the slave shall be initialized on the slots for which the clock satisfies the applicable equation:

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

$(\overline{CLK}_{27}, CLK_{26-1}) \bmod T_{ESCO} = D_{ESCO}$ 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 the rule used in the SCO reserved slots. The master may send a packet in the master slot. The slave may transmit on the eSCO LT_ADDR in the following slot either if it received a packet on the eSCO LT_ADDR in the previous slot or if it did not receive a valid packet header in the previous slot. When the master-to-slave packet type is a three-slot packet, the slave's transmit slot is the fourth slot of the eSCO reserved slots. A master shall transmit in an eSCO retransmission window on a given eSCO LT_ADDR only if it addressed that eSCO LT_ADDR in the immediately preceeding eSCO reserved slots. A slave may transmit on an eSCO LT_ADDR in the eSCO reserved slots only if the slave did not received a valid packet header with a different LT_ADDR in the eSCO reserved slots. Inside retransmission windows, the same polling rule as for ACL traffic is used: the slave shall transmit on the eSCO channel only if it received a valid packet header and correct LT_ADDR on the eSCO channel in the previous master-to-slave transmission slot. The master may transmit on any non-eSCO LT_ADDR in any master-to-slave transmission slot inside the eSCO retransmission window. The master shall transmit on an eSCO LT_ADDR in the retransmission window only 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 are 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 acknowledged, the master shall use the POLL packet. When the master-to-slave packet has been acknowledged 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 are 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 acknowledged, the slave shall use NULL packets. eSCO traffic should be given priority over ACL traffic in the retransmission window.

Figure 66 shows the eSCO window when single-slot packets are used.

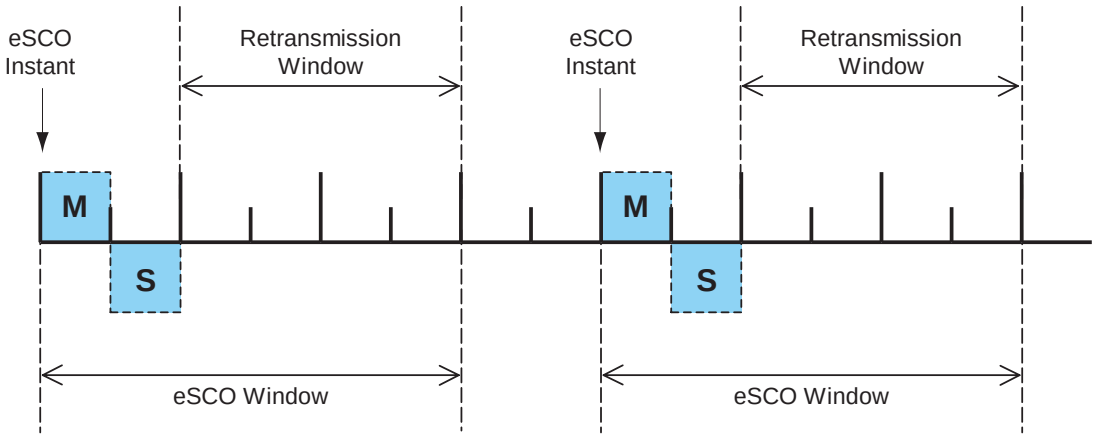


Figure 66—eSCO window details for single-slot packets

When multislot packets are used in either direction of the eSCO logical transport, the first transmission continues into the following slots. The retransmission window in this case starts the slot after the end of the slave-to-master packet, i.e., two, four, or six slots immediately following the eSCO instant are reserved and should not be used for other traffic. Figure 67 shows the eSCO window when multislot packets are used in one direction and single-slot packets are used in the other direction.

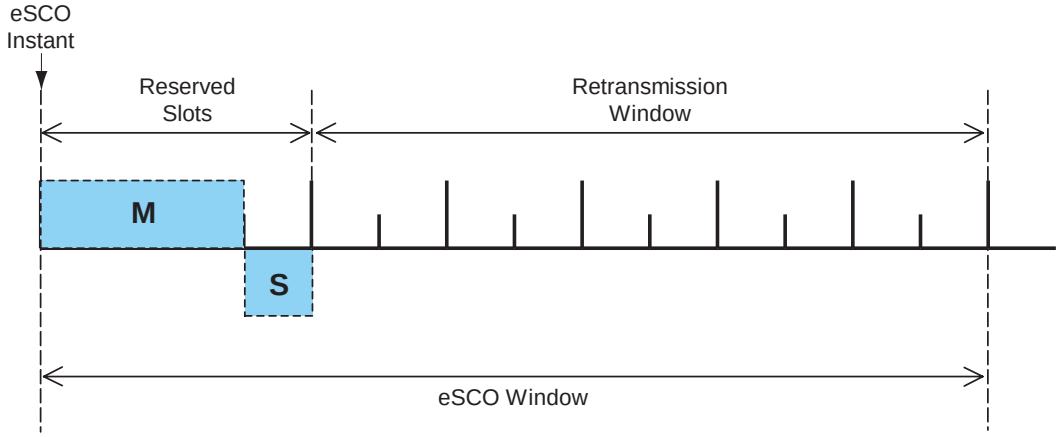


Figure 67—eSCO window details for asymmetric traffic

eSCO windows may overlap on the master, but shall not overlap on an individual slave.

In the reserved slots, both master and slave should listen and transmit at their allocated slots at the first transmission time of each eSCO window. Intermittent lapses due to, for instance, time-critical signaling during connection establishment are allowed. Both master and slave may refrain from sending data and may use instead POLL and NULL packets, respectively. When the master transmits a POLL packet instead of an **EV4** or **EV5** packet, the slave shall transmit, starting in the same slot as if the master transmitted an **EV4** or **EV5** packet. If the slave does not receive anything in the reserved master-to-slave transmission slot, it shall transmit in the same slot as if the master had transmitted the negotiated packet type. For example, if the master had negotiated an **EV5** packet, the slave would transmit three slots later. If the master does not receive a slave transmission in response to an eSCO packet, it causes an implicit NAK of the packet in question. The listening requirements for the slave during the retransmission window are the same as for an active ACL logical transport.

8.8.6.4 Broadcast scheme

The master of the piconet can broadcast messages to all slaves. A broadcast packet shall have an LT_ADDR set to all zero. Each new broadcast message (which may be carried by a number of packets) shall start with the start of L2CAP message indication (LLID = 10).

A broadcast packet shall never be acknowledged. In an error-prone environment, the master may carry out a number of retransmissions to increase the probability for error-free delivery (see also 8.7.6.5).

In order to support the PARK state (as described in 8.8.9), 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 8.8.9.

8.8.6.5 Role switch

There are several occasions when a role switch is used:

— A role switch is necessary when joining an existing piconet by paging since, by definition, the paging device is initially master of a “small” piconet involving only the pager (master) and the paged (slave) device.
— A role switch is necessary in order for a slave in an existing piconet to set up a new piconet with itself as master and the original piconet master as slave. If the original piconet had more than one slave, then this implies a double role for the original piconet master; it becomes a slave in the new piconet while still maintaining the original piconet as master.

Prior to the role switch, encryption, if present, shall be stopped in the old piconet. A role switch shall not be performed if the physical link is in SNIFF mode, HOLD mode, or PARK state or has any synchronous logical transports.

For the master and slave involved in the role switch, the switch results in a reversal of their TX and RX timing: a *TDD switch*. Additionally, since the piconet parameters are derived from the device address and clock of the master, a role switch inherently involves a redefinition of the piconet as well: a *piconet switch*. The new piconet’s parameters shall be derived from the former slave’s device address and clock.

Assume device A is to become master; device B was the former master. Then there are two alternatives: either the slave initiates the role switch or the master initiates the role switch. These alternatives are described in 9.3.4.2. The BB procedure is the same regardless of which alternative is used.

In step 1, the device A (slave) and devide B (master) shall perform a TDD switch using the former hopping scheme (still using the device address and clock of device B) so there is no piconet switch yet. The slot offset information sent by device A shall not be used yet, but shall be used in step 3. Device A now becomes the master; device B, the slave. The LT_ADDR formerly used by device A in its slave role shall now be used by devide B (now slave).

At the moment of the TDD switch, both devices A and B shall start a timer with a timeout of *newconnectionTO*. The timer shall be stopped in device B (slave) as soon as it receives an FHS packet from device A (master) on the TDD-switched channel. The timer shall be stopped in device A (master) as soon as it receives an ID packet from device B (slave). If the *newconnectionTO* expires, the master and slave shall return to the old piconet timing and AFH state, taking their old roles of master and slave. The FHS packet shall be sent by device A (master) using the old piconet parameters. The LT_ADDR in the FHS packet header shall be the former LT_ADDR used by device A. The LT_ADDR carried in the FHS payload shall be the new LT_ADDR intended for device B when operating on the new piconet. After the FHS acknowledgment, which is the ID packet and shall be sent by device B (slave) on the old hopping sequence, both device A (master) and device B (slave) shall use the new channel parameters of the new piconet as indicated by the

FHS with the sequence selection set to basic channel hopping sequence. If the new master has physical links that are AFH enabled, following the piconet switch, the new master is responsible for controlling the AFH operational mode of its new slave.

Since the old and new masters' clocks are synchronized, the clock information sent in the FHS payload shall indicate the new master's clock at the begining of the FHS packet transmission. Furthermore, the 1.25 ms resolution of the clock information given in the FHS packet is not sufficient for aligning the slot boundaries of the two piconets. The slot offset information in the LMP message previously sent by device A shall be used to provide more accurate timing information. The slot offset indicates the delay between the start of the master-to-slave slots of the old and new piconet physical channels. This timing information ranges from 0 to 1249 μs with a resolution of 1 μs. It shall be used together with the clock information in the FHS packet to accurately position the correlation window when switching to the new master's timing after acknowledgment of the FHS packet.

After reception of the FHS packet acknowledgment, device A (new master) shall switch to its own timing with the sequence selection set to the basic channel hopping sequence and shall send a POLL packet to verify the switch. Both the master and the slave shall start a new timer with a timeout of *newconnectionTO* on FHS packet acknowledgment. The start of this timer shall be aligned with the beginning of the first master TX slot boundary of the new piconet, following the FHS packet acknowledgment. The slave shall stop the timer when the POLL packet is received; the master shall stop the timer when the POLL packet is acknowledged. The slave shall respond with any type of packet to acknowledge the POLL. Any pending AFH_Instant shall be cancelled once the POLL packet has been received by the slave. If no response is received, the master shall resend the POLL packet until *newconnectionTO* is reached. If this timer expires, both the slave and the master shall return to the old piconet timing with the old master and slave roles. Expiry of the timer shall also restore the state associated with AFH (including any pending AFH_Instant), CQDDR (see 9.3.1.7), and power control (see 9.3.1.3). The procedure may then start again beginning at step 1. Aligning the timer with TX boundaries of the new piconet ensures that no device returns to the old piconet timing in the middle of a master RX slot.

After the role switch, the ACL logical transport is reinitialized as if it were a new connection. For example, the SEQN of the first data packet containing a CRC on the new piconet physical channel shall be set according to the rules in 8.7.6.2.

A parked slave must be unparked before it can participate in a role switch.

8.8.6.6 Scatternet

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

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

A master or slave can become a slave in another piconet by being paged by the master of the other piconet. On the other hand, a device participating in one piconet can page the master or slave of another piconet. Since the paging device always starts out as master, a master-slave role switch is required if a slave role is desired. This is described in the 8.8.6.5.

8.8.6.6.1 Interpiconet communications

Time multiplexing must be used to switch between piconets. Devices may achieve the time multiplexing necessary to implement scatternet by using SNIFF mode or by remaining in an active ACL connection. For an ACL connection in piconets where the device is a slave in the CONNECTION state, the device may choose to not listen in every master slot. In this case, it should be recognized that the QoS on this link may degrade abruptly if the slave is not present enough to match up with the master polling that slave. Similarly, in piconets where the device is master, the device may choose to not transmit in every master slot. In this case, it is important to honor T_{poll} as much as possible. Devices in SNIFF mode may have sufficient time to visit another piconet between sniff slots. When the device is a slave using SNIFF mode and there are not sufficient idle slots, the device may choose to not listen to all master transmission slots in the sniff_attempts period or during the subsequent sniff_timeout period. A master is not required to transmit during sniff slots and, therefore, has flexibility for scatternet. If SCO or eSCO links are established, other piconets shall be visited only in the nonreserved slots between reserved slots. This is possible only if there is a single SCO logical transport using **HV3** packets or eSCO links where at least four slots remain between the reserved slots. Since the multiple piconets are not synchronized, guard time must be left to account for misalignment. This means that only two slots can effectively be used to visit another piconet between the **HV3** packets. Since clocks are unsycnhronized and accurate only to ± 20 ppm when a device is active, or ± 250 ppm in low-power modes, the unsynchronized clocks of piconets will drift with respect to one another. This means that if a device has a synchronous link in one piconet, the timing of slots available to it in another piconet will drift. If SNIFF mode is used to manage bandwidth in the piconets, then sniff instants may have to be periodically renegotiated.

Since the clocks of two masters of different piconets are not synchronized, a slave device participating in two piconets shall maintain two offsets that, added to its own CLKN, create the two CLKs. Since the two CLKs drift independently, the slave must regularly update the offsets in order to keep synchronization to both masters.

8.8.6.7 Hop sequence switching

Hop sequence adaptation is controlled by the master and can be set to either enabled or disabled. Once enabled, hop sequence adaptation shall apply to all logical transports on a physical link. Once enabled, the master may periodically update the set of used and unused channels as well as disable hop sequence adaptation on a physical link. When a master has multiple physical links, the state of each link is independent of all other physical links.

When hop sequence adaptation is enabled, the sequence selection hop selection kernel input is set to adapted channel hopping sequence and the input AFH_channel_map is set to the appropriate set of used and unused channels. Additionally, the same channel mechanism shall be used. When hop sequence adaptation is enabled with all channels used, this is known as AHS(79).

When hop sequence adaptation is disabled, the sequence selection input of the hop selection kernel is set to basic channel hopping sequence (the input AFH_channel_map is unused in this case), and the same channel mechanism shall not be used.

The hop sequence adaptation state shall be changed when the master sends the LMP_set_AFH PDU and a BB acknowledgment is received. When the BB acknowledgment is received prior to the hop sequence switch instant AFH_Instant (see 9.3.1.4), the hop sequence proceeds as shown in Figure 68.

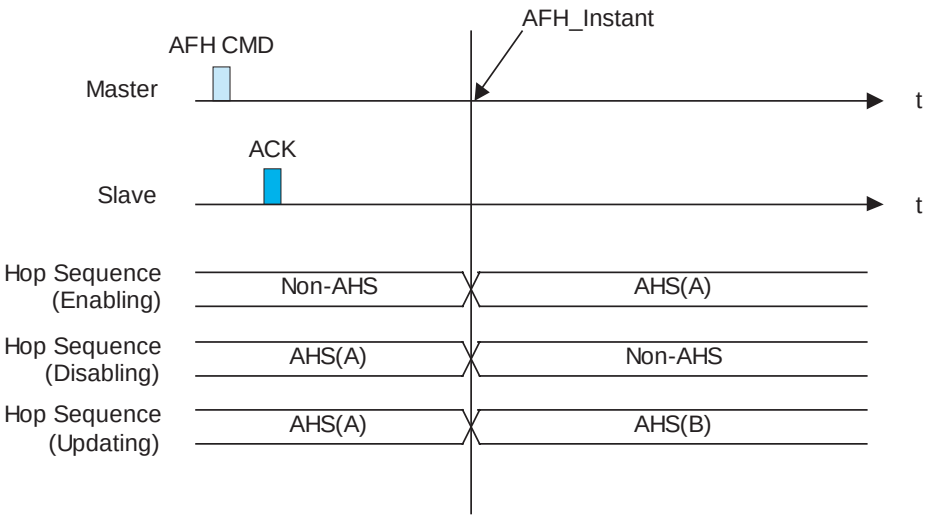


Figure 68—Successful hop sequence switching

When the BB acknowledgment is not received prior to the AFH_Instant, the master shall use a recovery hop sequence for the slave(s) that did not respond with an acknowledgment (this may be because the slave did not hear the master's transmission or the master did not hear the slave's transmission). When hop sequence adaptation is being enabled or disabled, the recovery sequence shall be the AFH_channel_map specified in the LMP_set_AFH PDU. When the AFH_channel_map is being updated, the master shall choose a recovery sequence that includes all of the RF channels marked as used in either the old or new AFH_channel_map, e.g., AHS(79). Once the BB acknowledgment is received, the master shall use the AFH_channel_map in the LMP_set_AFH PDU starting with the next transmission to the slave. See Figure 69.

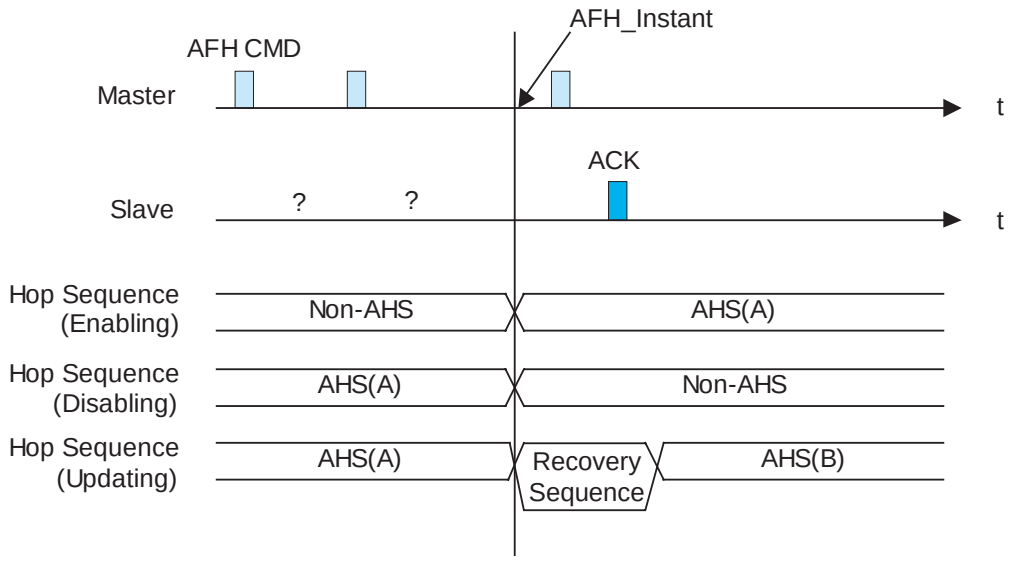


Figure 69—Recovery hop sequences

When the AFH_Instant occurs during a multislot packet transmitted by the master, the slave shall use the same hopping sequence parameters as the master used at the start of the multislot packet. An example of this is shown in Figure 70. In this figure the basic channel hopping sequence is designated *f*. The first adapted channel hopping sequence is designated with *f′*, and the second adapted channel hopping sequence is designated *f″*.

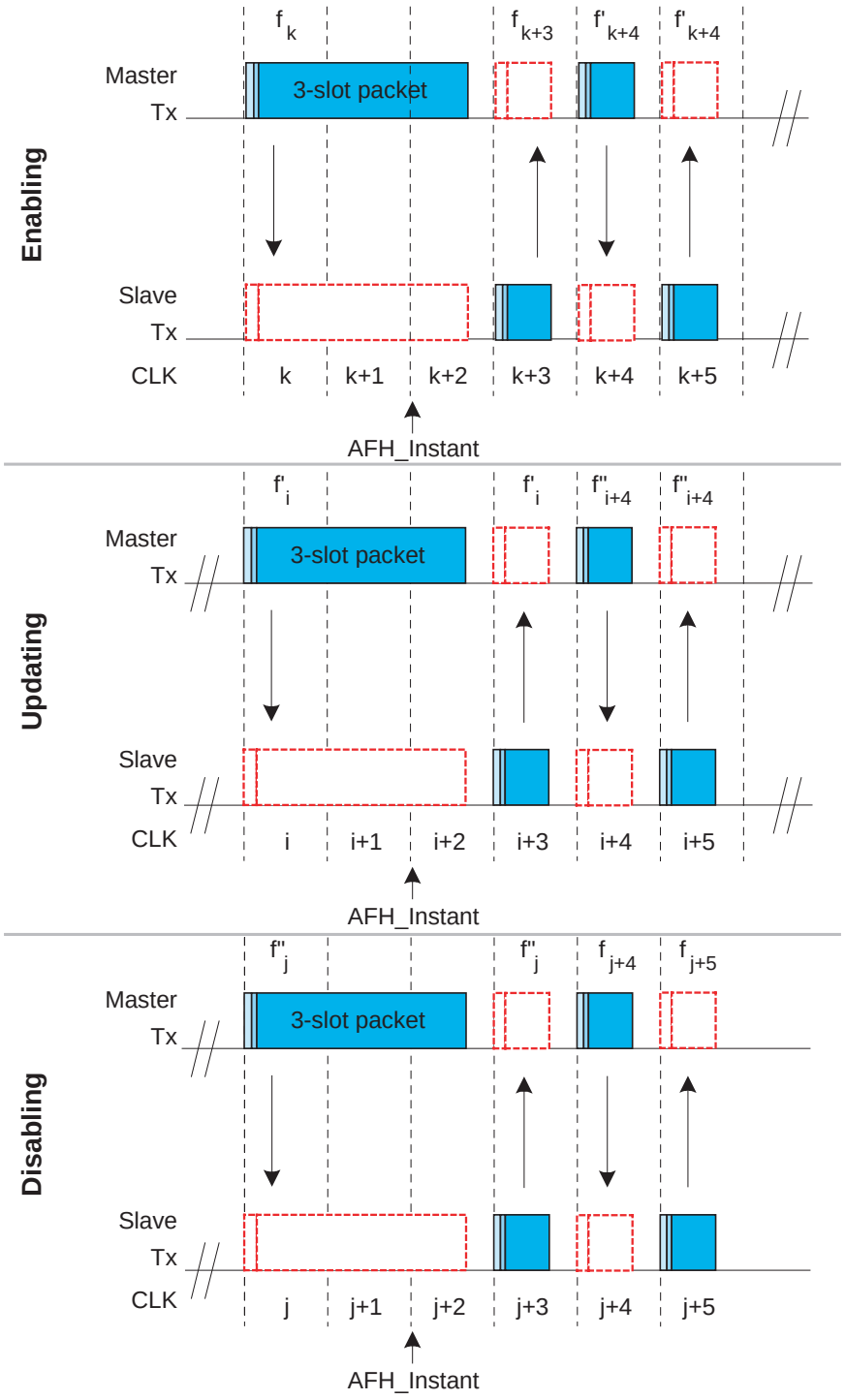


Figure 70—AFH_Instant changes during multislot packets transmitted by the master

8.8.6.8 Channel classification and channel map selection

RF channels are classified as being unknown, bad, or good. These classifications are determined individually by the master and slaves based on local information (e.g., active or passive channel assessment methods or from the host via HCI). Information received from other devices via LMP (e.g., an AFH_channel_map from a master or a channel classification report from a slave) shall not be included in a device's channel classification.

The three possible channel classifications shall be as defined in Table 29.

Table 29—Channel classification descriptions

Classification	Definition
Unknown	The channel assessment measurements are insufficient to reliably classify the channel, and the channel is not marked as bad in the most recent HCI Set_AFH_Channel_Classification command.
Bad	a) An ACL or synchronous throughput failure measure associated with a channel has exceeded a threshold (defined by the particular channel assessment algorithm employed). b) An interference-level measure associated with a channel, determining the interference level that the link poses upon other systems in the vicinity, has exceeded a threshold (defined by the particular channel assessment algorithm employed). c) A channel is marked as bad in the most recent HCI Set_AFH_Channel_Classification command.
Good	A channel is neither unknown or bad.

A master with AFH-enabled physical links shall determine an AFH_channel_map based on any combination of the following information:

— Channel classification from local measurements (e.g., active or passive channel assessment in the controller), if supported and enabled. The host may enable or disable local measurements using the HCI Write_AFH_Channel_Classification_Mode command, defined in 11.7.3.58, if HCI is present.
— Channel classification information from the host using the HCI Set_AFH_Channel_Classification command, defined in 11.7.3.58, if HCI is present. Channels classified as bad in the most recent AFH_Host_Channel_Classification command shall be marked as unused in the AFH_channel_map.
— Channel classification reports received from slaves in LMP_channel_classification PDUs, defined in 9.3.1.5.

The algorithm used by the master to combine these information sources and generate the AFH_channel_map is not defined in this standard and will be implementation specific. At no time shall the number of channels used be less than N_{min}, defined in 8.2.3.1.

If a master determines that all channels should be used, it may keep AFH operation enabled using an AFH_channel_map of 79 used channels, i.e., AHS(79).

8.8.6.9 Power management

Features are provided to allow low-power operation. These features are both at the microscopic level when handling the packets and at the macroscopic level when using certain operation modes.

8.8.6.9.1 Packet handling

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

8.8.6.9.2 Slot occupancy

As was described in 8.6.5, the packet type indicates how many slots a packet may occupy. A slave not addressed in the packet header may go to sleep for the remaining slots the packet occupies. This can be read from the TYPE code.

8.8.6.9.3 Recommendations for low-power operation

The most common and flexible methods for reducing power consumption are the use of SNIFF mode and PARK state. HOLD mode can also be used by repeated negotiation of HOLD periods.

8.8.7 SNIFF mode

In SNIFF mode, the duty cycle of the slave's activity in the piconet may be reduced. If a slave is in active mode on an ACL logical transport, it shall listen in every even ACL slot to that master traffic, unless

— A previous transmission indicated the slot would be occupied by a multislot packet for another slave, or
— That link is being treated as a scatternet link, or
— The device is absent due to HOLD mode

With SNIFF mode, the time slots when a slave is listening are reduced, so the master shall transmit to a slave only in specified time slots. The sniff anchor points are spaced regularly with an interval of $T_{sniff.}$

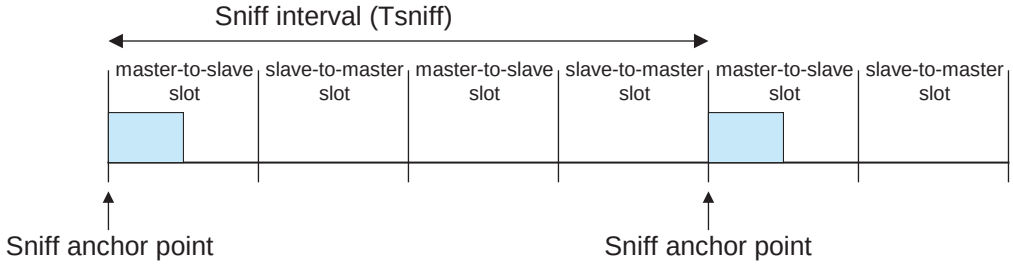


Figure 71—Sniff anchor points

The slave listens in master-to-slave transmission slots starting at the sniff anchor point. It shall continue listening if any of the following events occurs:

— If fewer than $N_{sniff\_attempt}$ master-to-slave transmission slots have elapsed since the sniff anchor point.

— If the slave has received a packet with a matching LT_ADDR that contains ACL data (**DM**, **DH**, **DV**, or **AUX1** packets) in the preceding $N_{sniff\_timeout}$ master-to-slave transmission slots.
— If the slave has transmitted a packet containing ACL data (**DM**, **DH**, **DV**, or **AUX1** packets) in the preceding $N_{sniff\_timeout}$ slave-to-master transmission slots.

A slave may continue listening if the slave has received any packet with a matching LT_ADDR in the preceding $N_{sniff\_timeout}$ master-to-slave transmission slots. Also, a device may override the rules above and stop listening prior to $N_{sniff\_timeout}$ or the remaining $N_{sniff\_attempt}$ slots if it has activity in another piconet.

It is possible that activity from one sniff timeout may extend to the next sniff anchor point. Any activity from a previous sniff timeout shall not affect activity after the next sniff anchor point. So in the above rules, only the slots since the last sniff anchor point are considered.

Note that $N_{sniff\_attempt} = 1$ and $N_{sniff\_timeout} = 0$ cause the slave to listen only at the slot beginning at the sniff anchor point, irrespective of packets received from the master.

$N_{sniff\_attempt} = 0$ shall not be used.

SNIFF mode applies only to asynchronous logical transports and their associated LT_ADDR. SNIFF mode shall not apply to synchronous logical transports; therefore, both masters and slaves shall still respect the reserved slots and retransmission windows of synchronous links.

To enter SNIFF mode, the master or slave shall issue a sniff command via the LMP. This message includes the sniff interval T_{sniff} and an offset D_{sniff}. In addition, an initialization flag indicates whether initialization procedure 1 or 2 shall be used. The device shall use initialization 1 when the MSB of the current master clock CLK_{27} is 0; it shall use initialization 2 when the MSB of the current master clock CLK_{27} is 1. The slave shall apply the initialization method as indicated by the initialization flag irrespective of its clock bit value CLK_{27}. The sniff anchor point determined by the master and the slave shall be initialized on the slots for which the clock satisfies the applicable equation:

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

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

This implies that D_{sniff} must be even.

After initialization, the clock value CLK(k + 1) for the next sniff anchor point shall be derived by adding the fixed interval T_{sniff} to the clock value of the current sniff anchor point:

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

8.8.7.1 SNIFF TRANSITION mode

SNIFF TRANSITION mode is a special mode that is used during the transition between SNIFF and active modes. It is required because during this transition the mode (SNIFF or active) in which the slave is located is unclear and it is necessary to ensure that the slave is polled correctly regardless of the mode in which it is located.

In SNIFF TRANSITION mode, the master shall maintain the active mode poll interval in case the slave is in active mode. In addition the master shall poll the slave at least once in the sniff attempt transmit slots starting at each sniff instant. Note that this transmission counts for the active mode polling as well. The master must use its high-power accurate clock when in SNIFF TRANSITION mode.

The precise circumstances under which the master enters SNIFF TRANSITION mode are defined in 9.3.5.3.1.

8.8.8 HOLD mode

During the CONNECTION state, the ACL logical transport to a slave can be put in a HOLD mode. In HOLD mode, the slave temporarily shall not support ACL packets on the channel. Any synchronous packet during reserved synchonous slots (from SCO and eSCO links) shall be supported. With the HOLD mode, capacity can be made free to do other things like scanning, paging, inquiring, or attending another piconet. The device in HOLD mode can also enter a low-power sleep mode. During HOLD mode, the slave device keeps its LT_ADDR(s).

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

8.8.9 PARK state

When a slave does not need to participate on the piconet physical channel, but still needs to remain synchronized to the channel, it can enter PARK state. PARK state is a state with very little activity in the slave. In the PARK state, the slave shall give up its LT_ADDR. Instead, it shall receive two new addresses to be used in the PARK state:

— PM_ADDR: 8-bit parked member address
— AR_ADDR: 8-bit access request address

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

The parked slave wakes up at regular intervals to listen to the channel in order to resynchronize and to check for broadcast messages. To support the synchronization and channel access of the parked slaves, the master supports a beacon train described in 8.8.9.1. The beacon structure is communicated to the slave when it is parked. When the beacon structure changes, the parked slaves are updated through broadcast messages.

The master shall maintain separate, nonoverlapping park beacon structures for each hop sequence. The beacon structures shall not overlap either their beacon slots or their access windows.

In addition to using PARK state for low power consumption, PARK state is used to connect more than seven slaves to a single master. At any one time, only seven slaves can be in the CONNECTION state. However, by swapping active and parked slaves in and out, respectively, of the piconet, the number of slaves can be much larger (255 if the PM_ADDR is used, and an arbitrarily large number if the BD_ADDR is used).

8.8.9.1 Beacon train

To support parked slaves, the master establishes a beacon train when one or more slaves are parked. The beacon train consists of one beacon slot or a train of equidistant beacon slots that is transmitted periodically with a constant time interval. The beacon train is illustrated in Figure 72. A train of N_B ($N_B \geq 1$) beacon slots is defined with an interval of T_B slots. The beacon slots in the train are separated by Δ_B. The start of the first beacon slot is referred to as the *beacon instant* and serves as the beacon timing reference. The beacon

parameters N_B and T_B are chosen so that there are sufficient beacon slots for a parked slave to synchronize to during a certain time window in an error-prone environment.

When parked, the slave shall receive the beacon parameters through an LMP command. In addition, the timing of the beacon instant is indicated through the offset D_B. As with the SCO logical transport (see 8.8.6.2), two initialization procedures (1 or 2) are used. The master shall use initialization 1 when the MSB of the current master clock CLK_{27} is 0; it shall use initialization 2 when the MSB of the current master clock CLK_{27} is 1. The chosen initialization procedure shall also be carried by an initialization flag in the LMP command. The slave shall apply the initialization method as indicated by the initialization flag irrespective of its clock bit CLK_{27}. The master-to-slave slot positioned at the beacon instant shall be initialized on the slots for which the clock satisfies the applicable equation:

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

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

This implies that D_B will be even.

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

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

The beacon train serves four purposes:

a) Transmitting master-to-slave packets that the parked slaves can use for resynchronization
b) Carrying messages to the parked slaves to change the beacon parameters
c) Carrying general broadcast messages to the parked slaves
d) Unparking one or more parked slaves

Since a slave can synchronize to any packet that is preceded by the proper CAC, the packets carried on the beacon slots do not have to contain specific broadcast packets for parked slaves to be able to synchronize; any packet may be used. The only requirement placed on the beacon slots is that there is a master-to-slave transmission present on the hopping sequence associated with the park structure. If there is no information to be sent, NULL packets may be transmitted by the master. If there is broadcast information to be sent to the parked slaves, the first packet of the broadcast message shall be repeated in every beacon slot of the beacon train. However, synchronous traffic in the synchronous reserved slots may interrupt the beacon transmission if it is on the same hopping sequence as the parked slaves. The master shall configure its park beacon structure so that reserved slots of synchronous logical transports do not cause slaves to miss synchronisation on a beacon slot. For example, a master that has active slaves using AHS, and parked slaves using non-AHS shall ensure that the park beacons cannot be interrupted by AHS synchronous reserved slots.

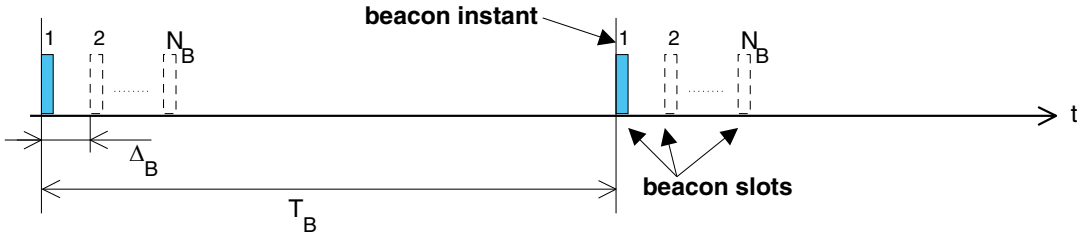


Figure 72—General beacon train format

The master can place parked slaves in any of the AFH operating modes, but shall ensure that all parked slaves use the same hop sequence. Masters should use AHS(79) or AHS when all the slaves in the piconet are AFH capable.

A master that switches a slave between AFH-enabled, AFH-disabled, or AHS(79) operation shall ensure that the AFH_Instant occurs before transmission of the beacon train using this hop sequence.

The master communicates with parked slaves using broadcast messages. Since these messages can be time-critical, an ongoing repetition train of broadcast message may be prematurely aborted by broadcast information destined to parked slaves in beacon slots and in access windows (see 8.8.9.2).

8.8.9.2 Beacon access window

In addition to the beacon slots, an access window is defined where the parked slaves can send requests to be unparked. To increase reliability, the access window may be repeated M_{access} times ($M_{access} \geq 1$) (see Figure 73). The access window starts a fixed delay D_{access} after the beacon instant. The width of the access window is T_{access}.

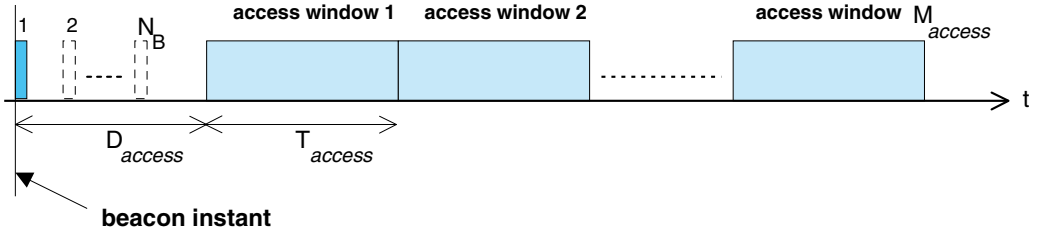


Figure 73—Definition of access window

The access window supports a polling slave access technique. The format of the polling technique is shown in Figure 74. The same TDD structure is used as on the piconet physical channel, i.e., master-to-slave transmission is alternated with slave-to-master transmission. The slave-to-master slot is divided into two half slots of 312.5 µs each. The half slot in which a parked slave is allowed to respond corresponds to its AR_ADDR (see also 8.8.9.6). For counting the half slots to determine the access request slot, the start of the access window is used (see Figure 74). The slave shall send an access request in the proper slave-to-master half slot only if a broadcast packet has been received in the preceding master-to-slave slot. In this way, the master polls the parked slaves.

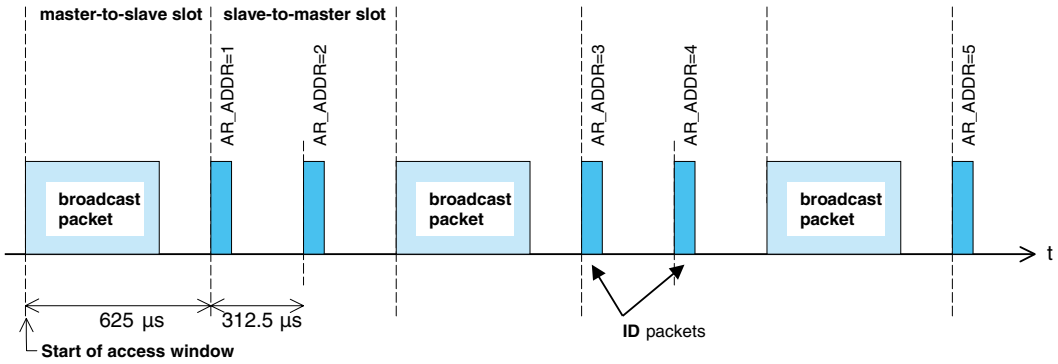


Figure 74—Access procedure applying the polling technique

The slots of the access window may also be used for traffic on the piconet if required. For example, if a synchronous connection has to be supported, the slots reserved for the synchronous link may carry synchronous information instead of being used for access requests, i.e., if the master-to-slave slot in the access window contains a packet different from a broadcast packet, the following slave-to-master slot shall not be used for slave access requests. If the master transmits a broadcast packet in the access window, then it shall use the hop sequence associated with the park structure. Slots in the access window not affected by piconet traffic may still be used according to the defined access structure (an example is shown in Figure 75), and the access procedure shall be continued as if no interruption had taken place.

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

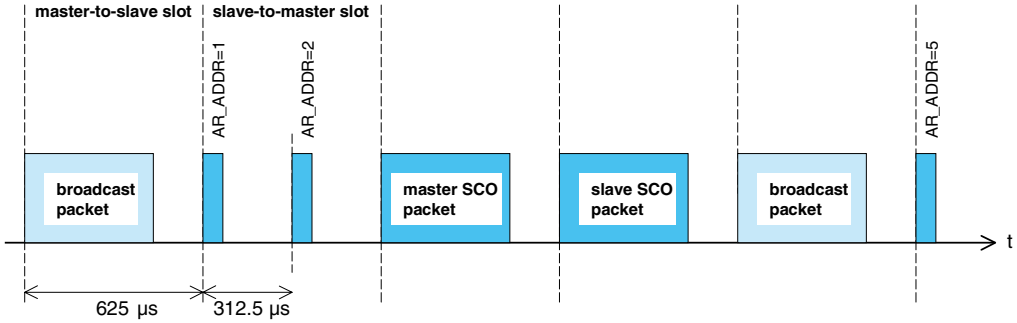


Figure 75—Disturbance of access window by SCO traffic

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

8.8.9.3 Parked slave synchronization

Parked slaves wake up periodically to resynchronize to the channel. Any packet exchanged on the channel can be used for synchronization. Since master transmission is mandatory on the beacon slots, parked slaves will use the beacon train to resynchronize. A parked slave may wake up at the beacon instant to read the packet sent on the first beacon slot. If this fails, it may retry on the next beacon slot in the beacon train; in total, there are N_B opportunities per beacon instant to resynchronize. During the search, the slave may increase its search window (see also 8.2.2.5.2). The separation between the beacon slots in the beacon train Δ_B shall be chosen so that consecutive search windows will not overlap.

The parked slave may not wake up at every beacon instant. Instead, a sleep interval may be applied that is longer than the beacon interval T_B (see Figure 76). The slave sleep window shall be a multiple N_{B_sleep} of T_B. The precise beacon instant on which the slave may wake up shall be indicated by the master with D_{B_sleep}, which indicates the offset (in multiples of T_B) with respect to the beacon instant ($0 < D_{B\_sleep} < N_{B\_sleep} - 1$). To initialize the wake-up period, the applicable equation shall be used:

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

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

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

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

A parked slave shall read the broadcast messages sent in the beacon slot(s) in which it wakes up. If the parked slave wakes up, the minimum wake-up activity shall be to read the CAC for resynchronization and the packet header to check for broadcast messages.

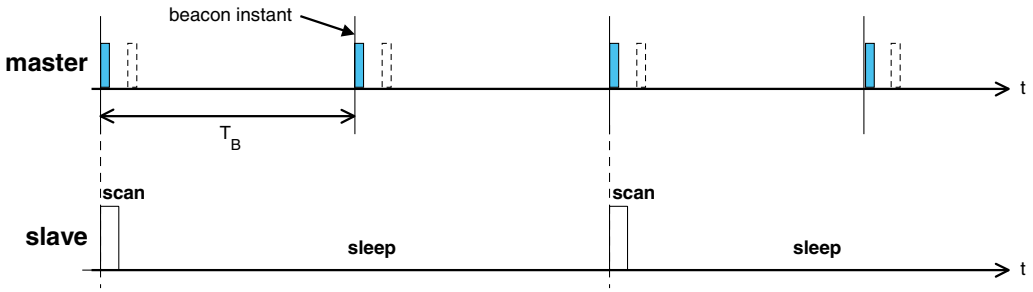


Figure 76—Extended sleep interval of parked slaves

8.8.9.4 Parking

A master can park an active slave through the exchange of LMP commands. Before being put into PARK state, the slave shall be assigned a PM_ADDR and an AR_ADDR. Every parked slave shall have a unique PM_ADDR or a PM_ADDR of 0. The AR_ADDR is not necessarily unique. The beacon parameters shall be given by the master when the slave is parked. The slave shall then give up its LT_ADDR and shall enter PARK state. A master can park only a single slave at a time. The park message is carried with a **DM1** packet and addresses the slave through its LT_ADDR.

8.8.9.5 Master-initiated unparking

The master can unpark a parked slave by sending a dedicated LMP unpark command including the parked slave's address. This message shall be sent in a broadcast packet on the beacon slots. The master shall use either the slave's PM_ADDR or its BD_ADDR. The message also includes the LT_ADDR the slave shall use after it has reentered the piconet. The unpark message may include a number of slave addresses so that multiple slaves may be unparked simultaneously. For each slave, a different LT_ADDR shall be assigned.

After having received the unpark message, the parked slave matching the PM_ADDR or BD_ADDR shall leave the PARK state and enter the CONNECTION state. It shall keep listening to the master until it is addressed by the master through its LT_ADDR. The first packet sent by the master shall be a POLL packet. The return packet in response to the POLL packet confirms that the slave has been unparked. If no response packets from the slave are received for *newconnectionTO* number of slots after the end of beacon repetition period, the master shall unpark the slave again. The master shall use the same LT_ADDR on each unpark attempt until the link supervision timer for that slave has elapsed or the unpark has completed successfully. If the slave does not receive the POLL packet for *newconnectionTO* number of slots after the end of beacon repetition period, it shall return to PARK state, with the same beacon parameters. After confirming that the slave is in the CONNECTION state, the master decides in which mode the slave will continue.

When a device is unparked, the SEQN bit for the link shall be reset to 1 on both the master and the slave (see 8.7.6.2.1).

8.8.9.6 Slave-initiated unparking

A slave can request access to the channel through the access window defined in 8.8.9.2. As shown in Figure 74 (in 8.8.9.2), the access window includes several slave-to-master half slots where the slave may send an access request message. The specific half slot in which the slave is allowed to respond corresponds to its AR_ADDR, which it received when it was parked. The order of the half slots (in Figure 74 the AR_ADDR numbers linearly increase from 1 to 5) is not fixed: an LMP command sent in the beacon slots may reconfigure the access window. When a slave desires access to the channel, it shall send an access request message in the proper slave-to-master half slot. The access request message of the slave is the ID packet containing the DAC of the master (which is the CAC without the trailer). The parked slave shall transmit an access request message in the half slot only when, in the preceding master-to-slave slot, a broadcast packet has been received. This broadcast message may contain any kind of broadcast information not necessarily related to the parked slave(s). If no broadcast information is available, a broadcast NULL or broadcast POLL packet may be sent to enable the access window.

After having sent an access request, the parked slave shall listen for an unpark message from the master. As long as no unpark message is received, the slave shall repeat the access requests in the subsequent access windows. After the last access window (there are M_{access} windows in total; see 8.8.9.2), the parked slave shall listen for an additional N_{poll} time slots for an unpark message. If no unpark message is received within N_{poll} slots after the end of the last access window, the slave may return to sleep and retry an access attempt after the next beacon instant.

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

When a device is unparked, the SEQN bit for the link shall be reset to 1 on both the master and the slave (see 8.7.6.2.1).

8.8.9.7 Broadcast scan window

In the beacon train, the master can support broadcast messages to the parked slaves. However, it may extend its broadcast capacity by indicating to the parked slaves that more broadcast information is following after the beacon train. This is achieved by an LMP command ordering the parked slaves (as well as the active slaves) to listen to the channel for broadcast messages during a limited time window. This time window starts at the beacon instant and continues for the period indicated in the LMP command sent in the beacon train.

8.8.9.8 Polling in the PARK state

In the PARK state, parked slaves may send access requests in the access window provided a broadcast packet is received in the preceding master-to-slave slot. Slaves in the CONNECTION state shall not send in the slave-to-master slots following the broadcast packet since they are allowed to send only if addressed specifically.

8.9 Audio

On the air-interface, either a 64 kb/s log pulse code modulation (PCM) format (A-law or μ-law) may be used or a 64 kb/s continuous variable slope delta (CVSD) modulation may be used. The latter format applies an adaptive delta modulation algorithm with syllabic companding.

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

Table 30 summarizes the voice coding schemes supported on the air interface. The appropriate voice coding scheme is selected after negotiations between the LMs.

Table 30—Voice coding schemes supported on the air interface

Voice codecs	
Linear	CVSD
8-bit logarithmic	A-law
	μ-law

8.9.1 Log PCM coder decoder (CODEC)

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

8.9.2 CVSD CODEC

A more robust format for voice-over-air interface is delta modulation. This modulation scheme follows the waveform where the output bits indicate whether the prediction value is smaller or larger than the input waveform. To reduce slope overload effects, syllabic companding is applied: the step size is adapted according to the average signal slope. The input to the CVSD encoder shall be 64 Ksample/s linear PCM (typically 16 bits, but actual value is implementation specific). Block diagrams of the CVSD encoder and CVSD decoder are shown in Figure 77, Figure 78, and Figure 79. The system shall be clocked at 64 kHz.

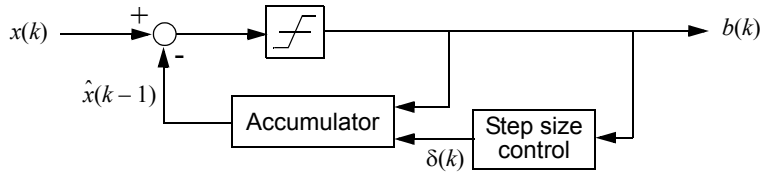


Figure 77—Block diagram of CVSD encoder with syllabic companding

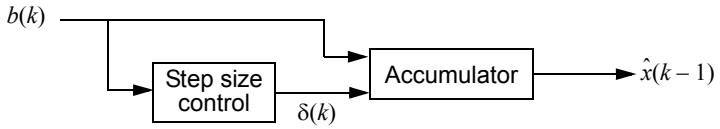


Figure 78—Block diagram of CVSD decoder with syllabic companding

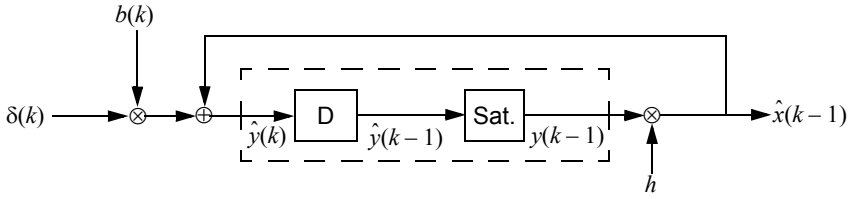


Figure 79—Accumulator procedure

Let $\mathrm{sgn}(x) = 1$ for $x \geq 0$; otherwise, $\mathrm{sgn}(x) = -1$. On air, these numbers shall be represented by the sign bit, i.e., negative numbers are mapped on 1, and positive numbers are mapped on 0.

Denote the CVSD encoder output bit $b(k)$, the encoder input $x(k)$, the accumulator contents $y(k)$, and the step size $\delta(k)$. Furthermore, let h be the decay factor for the accumulator, let β denote the decay factor for the step size, and, let α be the syllabic companding parameter. The last parameter monitors the slope by considering the K most recent output bits.

Let

$$\hat{x}(k) = hy(k). \tag{13}$$

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

$$b(k) = \mathrm{sgn}\{x(k) - \hat{x}(k-1)\}, \tag{14}$$

$$\alpha = \begin{cases} 1, & \text{if } J \text{ bits in the last } K \text{ output bits are equal,} \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

$$\delta(k) = \begin{cases} \min\{\delta(k-1) + \delta_{\min}, \delta_{\max}\}, & \alpha = 1, \\ \max\{\beta\delta(k-1), \delta_{\min}\}, & \alpha = 0, \end{cases} \tag{16}$$

$$y(k) = \begin{cases} \min\{\hat{y}(k), y_{\max}\}, & \hat{y}(k) \geq 0. \\ \max\{\hat{y}(k), y_{\min}\}, & \hat{y}(k) < 0. \end{cases} \tag{17}$$

where

$$\hat{y}(k) = \hat{x}(k-1) + b(k)\delta(k). \tag{18}$$

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

For a 64 kb/s CVSD, the parameters as shown in Table 31 shall be used. The numbers are based on a 16-bit signed number output from the accumulator. These values result in a time constant of 0.5 ms for the accumulator decay and a time constant of 16 ms for the step size decay.

Table 31—CVSD parameter values[a]

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

[a]The values are based on a 16-bit signed number output from the accumulator.

8.9.3 Error handling

In the **DV**, **HV3**, **EV3**, and **EV5** packets, the voice is not protected by FEC. The quality of the voice in an error-prone environment then depends on the robustness of the voice coding scheme and, in the case of eSCO, the retransmission scheme. CVSD, in particular, is rather insensitive to random bit errors, which are experienced as white background noise. When a packet is rejected because the CAC is incorrect, the HEC test was unsuccessful, or the CRC has failed, measures have to be taken to "fill" in the lost speech segment.

The voice payload in the **HV2** and **EV4** packets is protected by a 2/3 rate FEC. For errors that are detected, but cannot be corrected, the receiver should try to minimize the audible effects. For instance, from the 15-bit FEC segment with uncorrected errors, the 10-bit information part as found before the FEC decoder should be used. The **HV1** packet is protected by a 3-bit repetition FEC. For this code, the decoding scheme will always assume zero or 1-bit errors. Thus, there exist no detectable, but uncorrectable, error events for **HV1** packets.

8.9.4 General audio requirements

8.9.4.1 Signal levels

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

Full swing at the 16-bit linear PCM interface to the CVSD encoder is defined to be 3 dBm0.

8.9.4.2 CVSD audio quality

The requirements for audio quality are put on the TX side. The 64 Ksample/s linear PCM input signal should have negligible spectral power density above 4 kHz. The power spectral density in the 4–32 kHz band of the decoded signal at the 64 Ksample/s linear PCM output should be more than 20 dB below the maximum in the 0–4 kHz range.

8.10 General audio recommendations

8.10.1 Maximum sound pressure

It is the sole responsibility of each manufacturer to design its audio products in a safe way with regard to injury to the human ear. This standard does not specify maximum sound pressure from an audio device.

8.10.2 Other telephony network requirements

It is the sole responsibility of each manufacturer to design the audio product so that it meets the regulatory requirements of all telephony networks to which it may be connected.

8.10.3 Audio levels

Audio levels shall be calculated as send loudness rating (SLR) and receive loudness rating (RLR). The calculation methods are specified in ITU-T Recommendation P.79 [B14].

The physical test setup for handsets and headsets is described in ITU-T Recommendations P.51 [B12] and P.57 [B13].

The physical test setup for speakerphones and vehicle handsfree systems is specified in ITU-T Recommendation P.34 [B11].

A general equation for computation of loudness rating (LR) for telephone sets is given by ITU-T Recommendations P.79 and is given by

$$R = -\frac{10}{m}\log 10\left(\sum_{i=N_1}^{N_2} 10^{ms_i - w_i/10}\right), \tag{19}$$

where

m is a constant (~ 0.2);
w_i is weighting coefficient (different for the various LRs);
S_i is the sensitivity at frequency F_i, of the electro-acoustic path;
N_1,N_2 are consecutive filter bank numbers (Art. Index: 200–4000 Hz).

Equation (19) is used for calculating the SLR according to Figure 80 and RLR according to Figure 81. When calculating LRs, one must include only the parts of the frequency band where an actual signal transmission can occur in order to ensure that the additive property of LRs is retained. Therefore, ITU-T Recommendation P.79 uses only the 200–4000 Hz frequency band in LR computations.

8.10.4 Microphone path

The SLR measurement model is shown in Figure 80.

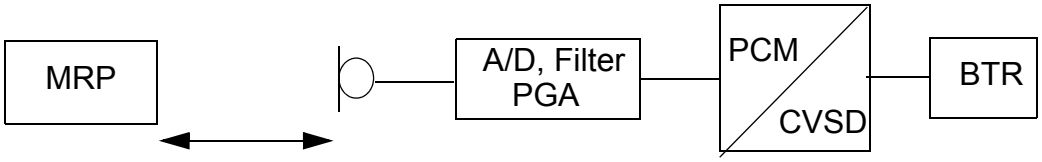


Figure 80—SLR measurement setup

8.10.5 Loudspeaker path

The RLR measurement model is shown in Figure 81.

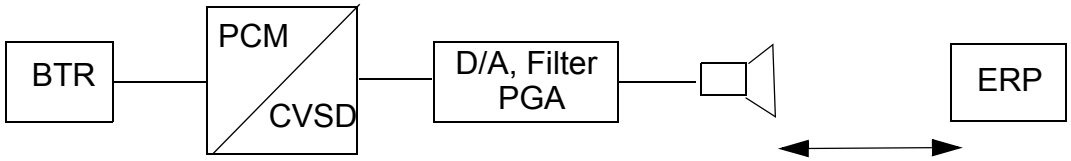


Figure 81—RLR measurement setup

8.10.6 Voice interface

The voice interface should follow in the first place the ITU-T Recommendations P.79 [B14], which specifies the LRs for telephone sets. These recommendations give general guidelines and specific algorithms used for calculating the LRs of the audio signal with respect to ear reference point (ERP).

For voice interfaces to the different cellular system terminals, loudness and frequency recommendations based on the cellular standards should be used. Since this standard is based on the Bluetooth specification, any recommendations for Bluetooth devices may also be applied to IEEE 802.15.1-2005 devices. For example, GSM 03.50 gives recommendation for both the LRs and frequency mask for a GSM terminal interconnection with IEEE 802.15.1-2005.

a) The output of the CVSD decoder are 16-bit linear PCM digital samples, at a sampling frequency of 8 ksample/s. IEEE 802.15.1-2005 also supports 8-bit log PCM samples of A-law and μ-law type. The sound pressure at the ERP for a given 16-bit CVSD sample should follow the sound pressure level given in the cellular standard specification.
b) A maximum sound pressure that can be represented by a 16-bit linear PCM sample at the output of the CVSD decoder should be specified according to the LR in ITU Recommendation P.79 and at the programmable gain amplifier (PGA) value of 0 dB. PGAs are used to control the audio level at the terminals by the user. For conversion between various PCM representations (A-law, μ-law, and linear PCM), ITU-T Recommendations G.711, G.712 [B9], and G.714 [B10] give guidelines and PCM value relationships. Zero-code suppression based on ITU-T Recommendation G.711 is also recommended to avoid network mismatches.

8.10.7 Frequency mask

For interfacing a device to a digital cellular mobile terminal, a compliance of the CVSD decoder signal to the frequency mask given in the cellular standard is recommended to guarantee correct function of the

speech coders. A recommendation for a frequency mask is given in Table 32. Figure 82 shows a plot of the frequency mask for IEEE 802.15.1-2005 (solid line). The GSM frequency mask (dotted line) is shown in Figure 82 for comparison.

Table 32—Recommended frequency mask for IEEE 802.15.1-2005

Frequency (Hz)	Upper limit (dB)	Lower limit (dB)
50	–20	—
300	4	–12
1000	4	–9
2000	4	–9
3000	4	–9
3400	4	–12
4000	0	—

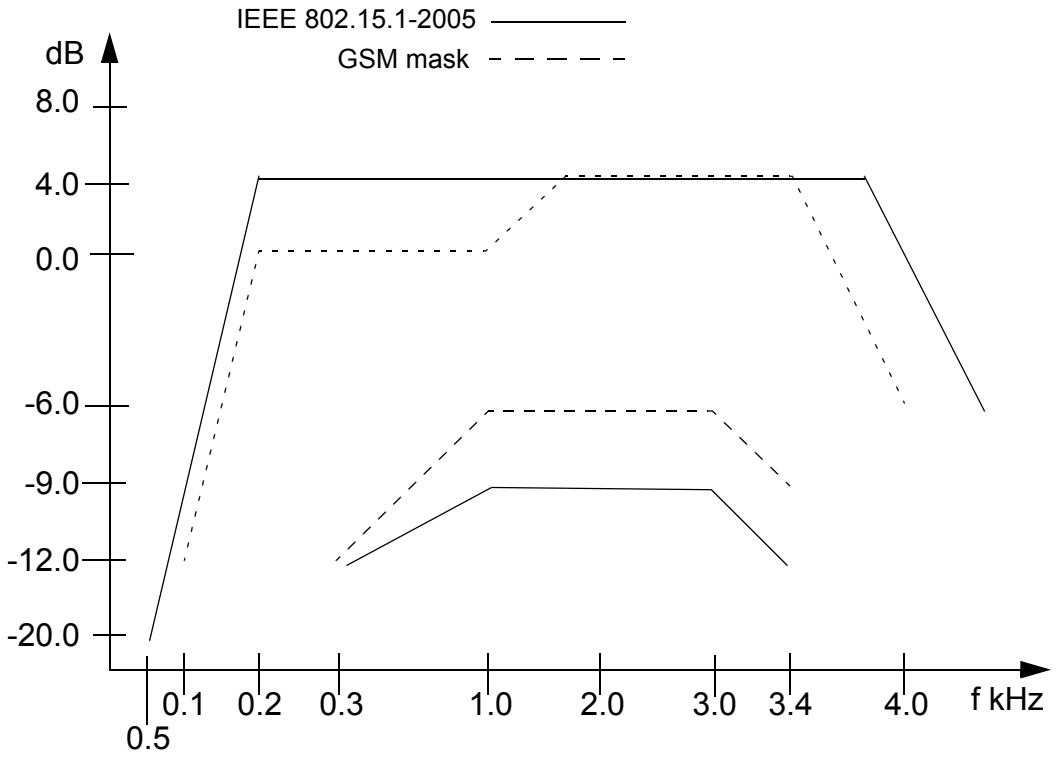


Figure 82—Plot of recommended frequency mask for IEEE 802.15.1-2005 with GSM send frequency mask given for comparison

8.11 Timers

This subclause contains a list of BB timers related to inactivity timeout defined in this standard. Definitions and default values of the timers are listed in 8.11.1 through 8.11.5.

All timer values are given in slots.

8.11.1 inquiryTO

The *inquiryTO* timer defines the number of slots the **inquiry** substate will last. The timer value may be changed by the host. HCI provides a command to change the timer value.

8.11.2 pageTO

The *pageTO* timer defines the number of slots the **page** substate can last before a response is received. The timer value may be changed by the host. HCI provides a command to change the timer value.

8.11.3 pagerespTO

In the slave, the *pageresp*TO timer defines the number of slots the slave awaits the master's response (FHS packet) after sending the page acknowledgment ID packet. In the master, the *pageresp*TO timer defines the number of slots the master should wait for the FHS packet acknowledgment before returning to the **page** substate. Both master and slave devices should use the same value for this timeout to ensure common page/scan intervals after reaching *pageresp*TO.

The value of the *pageresp*TO timer is 8 slots.

8.11.4 newconnectionTO

Every time a new connection is started through paging, scanning, role switching, or unparking, the master sends a POLL packet as the first packet in the new connection. Transmission and acknowledgment of this POLL packet are used to confirm the new connection. If the POLL packet is not received by the slave or the response packet is not received by the master for *newconnectionTO* number of slots, both the master and the slave will return to the previous substate.

The value of the *newconnectionTO* timer is 32 slots.

8.11.5 supervisionTO

The *supervisionTO* timer is used by both the master and slave to monitor link loss. If a device does not receive any packets that pass the HEC check and have the proper LT_ADDR for a period of *supervisionTO*, it will reset the link. The supervision timer keeps running in HOLD mode, SNIFF mode, and PARK state.

The *supervisionTO* value may be changed by the host. HCI provides a command to change the timer value. At the BB level, a default value that is equivalent to 20 s will be used.

8.12 Recommendations for AFH operation in PARK, HOLD, and SNIFF

The three possible AFH operation modes for an AFH-capable slave in PARK state, HOLD mode, and SNIFF mode are the same three AFH operation modes used during CONNECTION state:

— Enabled (using the same AHS as slaves in the CONNECTION state)
— Enabled [using AHS(79)]

— Disabled

The master may place an AFH-capable slave in any of the three AFH operating modes.

8.12.1 Operation at the master

A master that has one or more slaves in PARK state, HOLD mode, or SNIFF mode and decides to update them simultaneously shall schedule an AFH_Instant for a time that allows it to update all these slaves (as well as its active slaves) with the new adaptation.

A master that has multiple slaves with nonoverlapping “wake” times (e.g., slaves in SNIFF mode with different timing parameters) may keep them enabled on the same adaptation provided that its scheduling of the AFH_Instant allows enough time to update them all.

This timing is summarized in Figure 83. In this example, the master decides that a hop sequence adaptation is required. However, it cannot schedule an AFH_Instant until it has informed its active slave, its slave in HOLD mode, and its slave in SNIFF mode and had time to unpark its parked slaves.

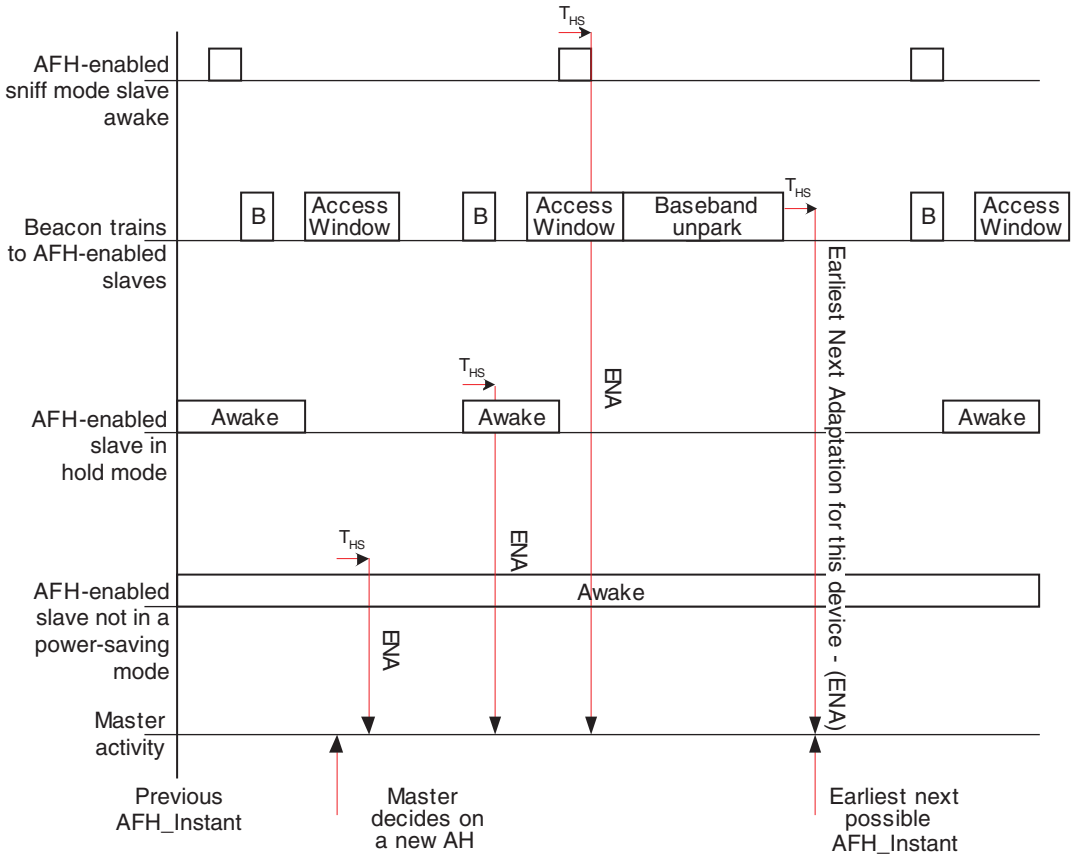


Figure 83—Timing constraint on AFH_Instant with slaves in PARK, HOLD, and SNIFF

8.12.2 Operation in PARK state

A slave that is in the PARK state cannot send or receive any AFH LMP messages (see 9.3.5.2). Once the slave has left the PARK state, the master may subsequently update the slave’s adaptation.

8.12.3 AFH operation in SNIFF mode

Once a slave has been placed in SNIFF mode, the master may periodically change its AHS without taking the slave out of SNIFF mode.

8.12.4 AFH operation in HOLD mode

A slave that is in HOLD mode cannot send or receive any LMP messages. Once the slave has left HOLD mode, the master may subsequently update the slave's adaptation.

9. Link Manager Protocol (LMP)

This clause describes the LMP, which is used for link setup and control. The signals are interpreted and filtered out by the LM on the receiving side and are not propagated to higher layers.

The LMP is used to control and negotiate all aspects of the operation of the IEEE 802.15.1-2005 connection between two devices. This includes the setup and control of logical transports and logical links and the control of physical links.

The LMP is used to communicate between the LMs on the two devices that are connected by the ACL logical transport. All LMP messages shall apply solely to the physical link and associated logical links and logical transports between the sending and receiving devices.

The protocol is made up of a series of messages that shall be transferred over the ACL-C logical link on the default ACL logical transport between two devices. LMP messages shall be interpreted and acted upon by the LM and shall not be directly propagated to higher protocol layers.

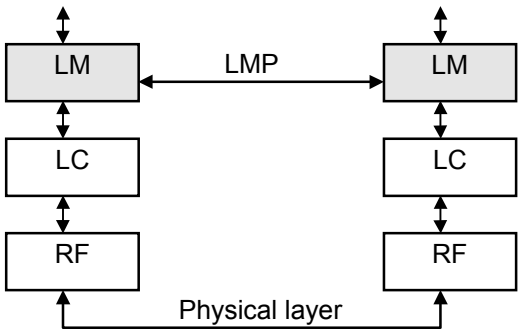


Figure 84—LMP signalling

9.1 General rules

9.1.1 Message transport

LMP messages shall be exchanged over the ACL-C logical link that is carried on the default ACL logical transport (see 8.4.4). The ACL-C logical link is distinguished from the ACL-U (which carries L2CAP and user data) by the LLID field carried in the payload header of variable-length packets (see 8.6.6.2).

The ACL-C has a higher priority than other traffic (see 8.5.5).

The error detection and correction capabilities of the BB ACL logical transport are generally sufficient for the requirements of the LMP. Therefore, LMP messages do not contain any additional error detection information beyond what can be realized by means of sanity checks performed on the contents of LMP messages. Any such checks and protections to overcome undetected errors in LMP messages are an implementation matter.

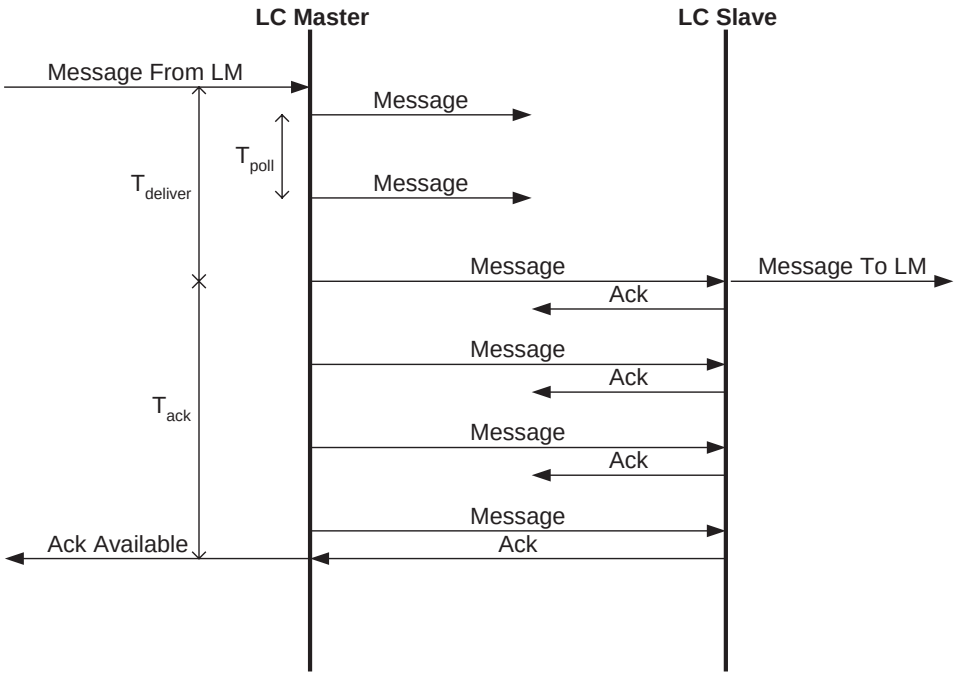
9.1.2 Synchronization

This subclause is informative and explains why many of the LMP message sequences are defined as they are.

LMP messages are carried on the ACL-C logical link, which does not guarantee a time to deliver or time to acknowledge packets. LMP procedures take account of this when synchronizing state changes in the two devices. For example, criteria are defined that specify when a LT_ADDR may be reused after it becomes available due to a device leaving the piconet or entering the PARK state. Other LMP procedures, such as HOLD or role switch, include the CLK as a parameter in order to define a fixed synchronization point. The transitions into and out of SNIFF mode are protected with a transition mode.

The link controller normally attempts to communicate with each slave no less often than every T_{poll} slots (see 9.3.1.8) based on the T_{poll} for that slave.

Figure 85 illustrates the fundamental problem. It shows the transmission of a packet from the master to the slave in conditions of heavy interference for illustrative purposes. It is obvious that neither side can determine the value of either $T_{deliver}$ or T_{ack}. It is, therefore, not possible to use simple messages to identify uniquely the instant at which a state change occurs in the other device.



Note that the diagram shows the limiting case where the master transmits the message at intervals of T_{poll}. In the case of heavy interference, improved performance is gained by transmitting more often.

Figure 85—Transmission of message from master to slave

9.1.3 Packet format

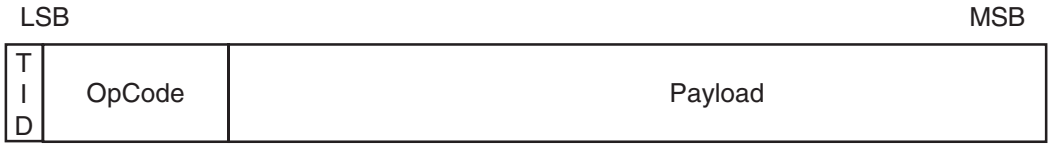
Each PDU is assigned either a 7-bit or a 15-bit opcode used to uniquely identify different types of PDUs (see Table 67 in 9.4). The first 7 bits of the opcode and a transaction ID are located in the first byte of the payload body. If the initial 7 bits of the opcode have one of the special escape values 124–127, then an additional byte of opcode is located in the second byte of the payload, and the combination uniquely identifies the PDU.

The FLOW bit in the packet header is always 1 and is ignored on reception.

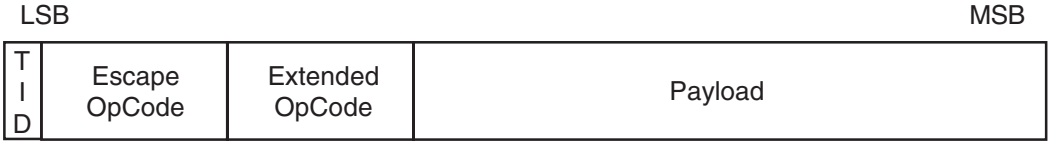
If the PDU contains one or more parameters, these are placed in the payload starting immediately after the opcode, i.e., at byte 2 if the PDU has a 7-bit opcode or byte 3 if the PDU has a 15-bit opcode. The number of

bytes used depends on the length of the parameters. All parameters have a little-endian format, i.e., the least significant byte is transferred first.

LMP messages shall be transmitted using **DM1** packets; however, if an **HV1** SCO link is in use and the length of the payload is less than 9 bytes, then **DV** packets may be used.



LMP PDU with 7 bit opCode



LMP PDU with 15 bit opCode

Figure 86—Payload body when LMP PDUs are sent

9.1.4 Transactions

The LMP operates in terms of transactions. A transaction is a connected set of message exchanges that achieve a particular purpose. All PDUs that form part of the same transaction shall have the same value for the transaction ID that is stored as part of the first byte of the opcode (see 9.1.3).

The transaction ID is in the LSB. It shall be 0 if the PDU forms part of a transaction that was initiated by the master and 1 if the transaction was initiated by the slave.

Each sequence described in 9.3 shall be defined as a transaction. For pairing (see 9.3.2.2) and encryption (see 9.3.2.5), all sequences belonging to each subclause shall be counted as one transaction and shall use the same transaction ID. For connection establishment (see 9.3.1.1), LMP_host_connection_req and the response with LMP_accepted or LMP_not_accepted shall form one transaction and have the transaction ID of 0. LMP_setup_complete is a stand-alone PDU, which forms a transaction by itself.

For error handling (see 9.1.5), the PDU to be rejected and LMP_not_accepted or LMP_not_accepted_ext shall form a single transaction.

9.1.4.1 LMP response timeout

The time between receiving a BB packet carrying an LMP PDU and sending a BB packet carrying a valid response PDU, according to the procedure rules in 9.3, shall be less than the LMP response timeout. The value of this timeout is 30 s. Note that the LMP response timeout is applied not only to sequences described in 9.3, but also to the series of the sequences defined as the transactions in 9.3. It shall also be applied to the series of LMP transactions that take place during a period when traffic on the ACL-U logical link is disabled for the duration of the series of LMP transactions, e.g., during the enabling of encryption.

The LMP response timeout shall restart each time an LMP PDU that requires a reply is queued for transmission by the BB.

9.1.5 Error handling

If the LM receives a PDU with an unrecognized opcode, it shall respond with an LMP_not_accepted or LMP_not_accepted_ext PDU with the error code *unknown LMP PDU*. The opcode parameter that is echoed back is the unrecognized opcode.

If the LM receives a PDU with invalid parameters, it shall respond with an LMP_not_accepted or LMP_not_accepted_ext PDU with the error code *invalid LMP parameters*.

If the maximum response time (see 9.1.4) is exceeded or if a link loss is detected (see 8.3.1), the party that waits for the response shall conclude that the procedure has terminated unsuccessfully.

Erroneous LMP messages can be caused by errors on the channel or systematic errors at the TX side. To detect the latter case, the LM should monitor the number of erroneous messages and disconnect if it exceeds a threshold, which is implementation dependent.

When the LM receives a PDU that is not allowed and the PDU normally expects a PDU reply, e.g., LMP_host_connection_req or LMP_unit_key, the LM shall return an LMP_not_accepted or LMP_not_accepted_ext PDU with the error code *PDU not allowed*. If the PDU normally does not expect a reply, e.g., LMP_sres or LMP_temp_key, the PDU will be ignored.

The reception of an optional PDU that is not supported shall be handled in one of two ways: If the LM simply does not know the opcode (e.g., it was added at a later version of this standard), it shall respond with an LMP_not_accepted or LMP_not_accepted_ext PDU with the error code *unknown LMP PDU*. If the LM recognises the PDU as optional, but unsupported, then it shall reply with an LMP_not_accepted or LMP_not_accepted_ext PDU with the error code *unsupported LMP feature* if the PDU would normally generate a reply; otherwise, no reply is generated.

9.1.5.1 Transaction collision resolution

Since LMP PDUs are not interpreted in real time, collision situations can occur where both LMs initiate the same procedure and both procedures cannot be completed. In this situation, the master shall reject the slave-initiated procedure by sending an LMP_not_accepted or LMP_not_accepted_ext PDU with the error code *LMP error transaction collision*. The master-initiated procedure shall then be completed.

Collision situations can also occur where both LMs initiate different procedures and both procedures cannot be completed. In this situation, the master shall reject the slave-initiated procedure by sending an LMP_not_accepted or LMP_not_accepted_ext PDU with the error code *LMP error different transaction collision*. The master- initiated procedure shall then be completed.

9.1.6 Procedure rules

Each procedure is described and depicted with a sequence diagram. The symbols in Figure 87 are used in the sequence diagrams.

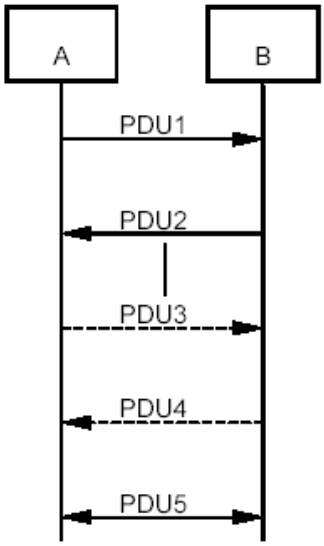


Figure 87—Symbols used in sequence diagrams

PDU1 is a PDU sent from A to B. PDU2 is a PDU sent from B to A. PDU3 is a PDU that is optionally sent from A to B. PDU4 is a PDU that is optionally sent from B to A. PDU5 is a PDU sent from either A or B. A vertical line indicates that more PDUs can optionally be sent.

9.1.7 General response messages

The PDUs LMP_accepted, LMP_accepted_ext, LMP_not_accepted, and LMP_not_accepted_ext are used as response messages to other PDUs in a number of different procedures. LMP_accepted or LMP_accepted_ext includes the opcode of the message that is accepted. LMP_not_accepted or LMP_not_accepted_ext includes the opcode of the message that is not accepted and the error code indicating why it is not accepted. See Table 33

LMP_accepted_ext and LMP_not_accepted_ext shall be used when the opcode is 15 bits in length. LMP_accepted and LMP_not_accepted shall be used when the opcode is 7 bits in length.

Table 33—General response messages

M/O[a]	PDU	Contents
M	LMP_accepted	Opcode
M	LMP_not_accepted	Opcode Error code
O	LMP_accepted_ext	Escape opcode Extended opcode
O	LMP_not_accepted_ext	Escape opcode Extended opcode Error code

[a]M = mandatory; O = optional.

9.1.8 LMP message constraints

This subclause is informative.

— No LMP message shall exceed the maximum payload length of a single **DM1** packet, i.e., 17 bytes in length (see 8.6.5.4.1).
— All LM messages are of fixed length apart from those sent using broadcast in PARK state.
— The LMP version number is not be used to indicate the presence or absence of functionality.

9.2 Device features

Each PDU is either mandatory or optional as defined by the M/O columns in the tables of 9.3. An M in this column shall indicate that support for the feature is mandatory. An O in this column shall indicate that support for the PDU is optional, and it shall be followed by the number(s) of the feature(s) involved (in parentheses).

All features added after IEEE Std 802.15.1-2002 have associated LMP feature bits. Support of these features may be made “mandatory” by the qualification process, but the LM still considers them to be optional since it must interoperate with older devices that do not support them.

The LM does not need to be able to transmit a PDU that is optional. Support of optional PDUs is indicated by a device’s features mask. The features mask can be read (see 9.3.3.4). An LM shall not send or be sent any PDU that is incompatible with the features signaled in its features mask or the features mask of its peer, as detailed in 9.2.1.

9.2.1 Feature definitions

Table 34 provides summary definitions of the features provided by LMP and references the subclauses containing more detailed definitions. Some features have only local meaning and do not imply support for any additional LMP PDUs or sequences. Although local features have no meaning for the remore LM, they are still included in the feature definitions because they are meaningful to the local host. In systems implementing HCI, the host may read the LM features using the HCI_Read_Local_Supported_Features command.

Table 34—Feature definitions

Feature	Definition
Extended features	This feature indicates whether the device is able to support the extended features mask using the LMP sequences defined in 9.3.3.4.
Timing accuracy	This feature indicates whether the LM supports requests for timing accuracy using the sequence defined in 9.3.3.1.
Enhanced inquiry scan	This feature indicates whether the device is capable of supporting the enhanced inquiry scan mechanism as defined in 8.8.4.1. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Interlaced inquiry scan	This feature indicates whether the device is capable of supporting the interlaced inquiry scan mechanism as defined in 8.8.4.1. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Interlaced page scan	This feature indicates whether the device is capable of supporting the interlaced page scan mechanism as defined in 8.8.3.1. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.

Table 34—Feature definitions *(continued)*

Feature	Definition
RSSI with inquiry results	This feature indicates whether the device is capable of reporting the RSSI with inquiry results as defined in 8.8.4.2. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Paging parameter negotiation	This feature indicates whether the LM is capable of supporting the signaling of changes in the paging scheme as defined in 9.3.1.9.
3-slot packets	This feature indicates whether the device supports the transmission and reception of both **DM3** and **DH3** packets for the transport of traffic on the ACL-U logical link.
5-slot packets	This feature indicates whether the device supports the transmission and reception of both **DM5** and **DH5** packets for the transport of traffic on the ACL-U logical link.
Flow control lag	This is defined as the total amount of ACL-U data that can be sent following the receipt of a valid payload header with the payload header FLOW bit set to 0 and is in units of 256 bytes. See further details in 8.6.6.2.
AFH-capable slave	This feature indicates whether the device is able to support the AFH mechanism as a slave as defined in 8.2.3 using the LMP sequences defined in 9.3.1.4.
AFH classification slave	This feature indicates whether the device is able to support the AFH classification mechanism as a slave as defined in 8.8.6.8 using the LMP sequences defined in 9.3.1.5.
AFH-capable master	This indicates whether the device is able to support the AFH mechanism as a master as defined in 8.2.3 using the LMP sequences defined in 9.3.1.4.
AFH classification master	This feature indicate whether the device is able to support the AFH classification mechanism as a master as defined in 8.8.6.8 using the LMP sequences defined in 9.3.1.5.
Power control	This feature indicates whether the device is capable of adjusting its transmission power. This feature indicates the ability to receive the LMP_incr_power and LMP_decr_power PDUs and transmit the LMP_max_power and LMP_min_power PDUs, using the sequences defined in 9.3.1.3. These sequences may be used even if the remote device does not support the power control feature, as long as it supports the power control requests feature.
Power control requests	This feature indicates whether the device is capable of determining if the transmit power level of the other device should be adjusted and will send the LMP_incr_ power and LMP_decr_power PDUs to request the adjustments. It indicates the ability to receive the LMP_max_power and LMP_min_power PDUs, using the sequences in 9.3.1.3. These sequences may be used even if the remote device does not support the RSSI feature, as long as it supports the power control feature.
CQDDR	This feature indicates whether the LM is capable of recommending a packet type (or types) depending on the channel quality using the LMP sequences defined in 9.3.1.7.
Broadcast encryption	This feature indicates whether the device is capable of supporting broadcast encryption as defined in 13.4.2 and also the sequences defined in 9.3.2.5.5 and 9.3.2.4. NOTE—Devices compliant to IEEE Std 802.15.1-2002 may support broadcast encryption even though this feature bit is not set.
Encryption	This feature indicates whether the device supports the encryption of packet contents using the sequence defined in 9.3.2.5.
Slot offset	This feature indicates whether the LM supports the transfer of the slot offset using the sequence defined in 9.3.4.1.
Role switch	This feature indicates whether the device supports the change of master and slave roles as defined by 8.8.6.5 using the sequence defined in 9.3.4.2.

Table 34—Feature definitions *(continued)*

Feature	Definition
HOLD mode	This feature indicates whether the device is able to support HOLD mode as defined in 8.8.8 using the LMP sequences defined in 9.3.5.1.
SNIFF mode	This feature indicates whether the device is able to support SNIFF mode as defined in 8.8.7 using the LMP sequences defined in 9.3.5.3.
PARK state	This feature indicates whether the device is able to support PARK state as defined in 8.8.9 using the LMP sequences defined in 9.3.5.2.
SCO link	This feature indicates whether the device is able to support the SCO logical transport as defined in 8.4.3, the **HV1** packet defined in 8.6.5.2.1, and the **DV** packet defined in 8.6.5.2.4 using the LMP sequence in 9.3.6.1.
HV2 packets	This feature indicates whether the device is capable of supporting the **HV2** packet type as defined in 8.6.5.2.2 on the SCO logical transport.
HV3 packets	This feature indicates whether the device is capable of supporting the **HV3** packet type as defined in 8.6.5.2.3 on the SCO logical transport.
μ-law log synchronous data	This feature indicates whether the device is capable of supporting μ-law log format data as defined in 8.9.1 on the SCO and eSCO logical transports.
A-law log synchronous data	This feature indicates whether the device is capable of supporting A-law log format data as defined in 8.9.1 on the SCO and eSCO logical transports.
CVSD synchronous data	This feature indicates whether the device is capable of supporting CVSD format data as defined in 8.9.2 on the SCO and eSCO logical transports.
Transparent synchronous data	This feature indicates whether the device is capable of supporting transparent synchronous data as defined in 8.6.4.3 on the SCO and eSCO logical transports.
Extended SCO link	This feature indicates whether the device is able to support the eSCO logical transport as defined in 8.5.4 and the **EV3** packet defined in 8.6.5.3.1 using the LMP sequences defined in 9.3.6.2.
EV4 packets	This feature indicates whether the device is capable of supporting the **EV4** packet type defined in 8.6.5.3.2 on the eSCO logical transport.
EV5 packets	This feature indicates whether the device is capable of supporting the **EV5** packet type defined in 8.6.5.3.3 on the eSCO logical transport.

9.2.2 Features mask definition

The features are represented as a bit mask when they are transferred in LMP messages. For each feature, a single bit is specified, which shall be set to 1 if the feature is supported and set to 0 otherwise. The single exception is the flow control lag, which is coded as a 3-bit field with the LSB in byte 2 bit 4 and the MSB in byte 2 bit 6. All unknown or unassigned feature bits shall be set to 0. See Table 35.

Table 35—Features mask definition

Number	Supported feature	Byte	Bit
0	3-slot packets	0	0
1	5-slot packets	0	1
2	Encryption	0	2

Table 35—Features mask definition *(continued)*

Number	Supported feature	Byte	Bit
3	Slot offset	0	3
4	Timing accuracy	0	4
5	Role switch	0	5
6	HOLD mode	0	6
7	SNIFF mode	0	7
8	PARK state	1	0
9	Power control requests	1	1
10	CQDDR	1	2
11	SCO link	1	3
12	**HV2** packets	1	4
13	**HV3** packets	1	5
14	μ-law log synchronous data	1	6
15	A-law log synchronous data	1	7
16	CVSD synchronous data	2	0
17	Paging parameter negotiation	2	1
18	Power control	2	2
19	Transparent synchronous data	2	3
20	Flow control lag(LSB)	2	4
21	Flow control lag(middle bit)	2	5
22	Flow control lag(MSB)	2	6
23	Broadcast encryption	2	7
24	Reserved	3	0
25	Reserved	3	1
26	Reserved	3	2
27	Enhanced inquiry scan	3	3
28	Interlaced inquiry scan	3	4
29	Interlaced page scan	3	5
30	RSSI with inquiry results	3	6
31	Extended SCO link (**EV3** packets)	3	7
32	**EV4** packets	4	0
33	**EV5** packets	4	1
34	Reserved	4	2
35	AFH-capable slave	4	3

Table 35—Features mask definition *(continued)*

Number	Supported feature	Byte	Bit
36	AFH classification slave	4	4
37	Reserved	4	5
38	Reserved	4	6
43	AFH-capable master	5	3
44	AFH classification master	5	4
63	Extended features	7	7

9.2.3 LM interoperability policy

LMs of any version will interoperate using the lowest common subset of functionality by reading the LMP features mask (defined in Table 35).

An optional LMP PDU shall be sent to a device only if the corresponding feature bit is set in its features mask. The exception to this are certain PDUs (see 9.3.1.1) that can be sent before the features mask is requested.

NOTE—A later version device with a restricted feature set is indistinguishable from an earlier version device with the same features.[11]

9.3 Procedure rules

This subclause describes the rules for carrying out LMP procedures.

9.3.1 Connection control

This subclause describes the LMP procedures for controlling connections, including connection establishment, detach, and power control within a connection.

9.3.1.1 Connection establishment

After the paging procedure, LMP procedures for clock offset request, LMP version, supported features, name request, and detach may be initiated.

An LM connection may be established during a device discovery phase in order to retrieve information such as the LM version, supported features, and the device's name. Such a connection does not involve higher layers and, therefore, does not need to proceed to an LM_host_connection_req PDU. After the desired information has been retrieved, an LM_detach PDU is sent. Figure 88 illustrates this transaction.

A typical connection establishment that is initiated by host request is shown in Figure 89.

[11]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement this standard.

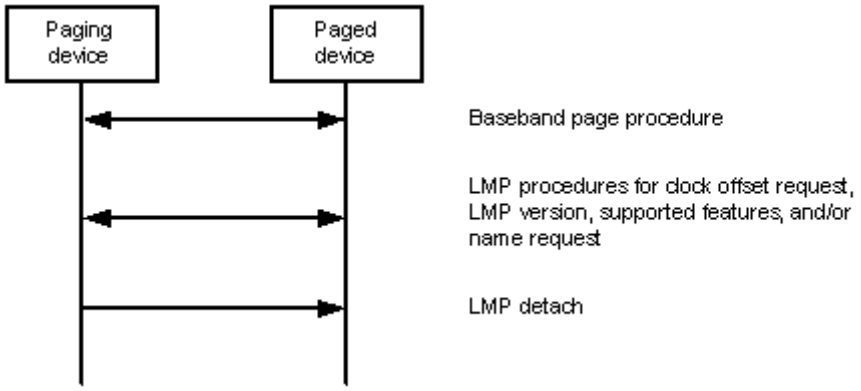


Figure 88—Connection establishment without host request

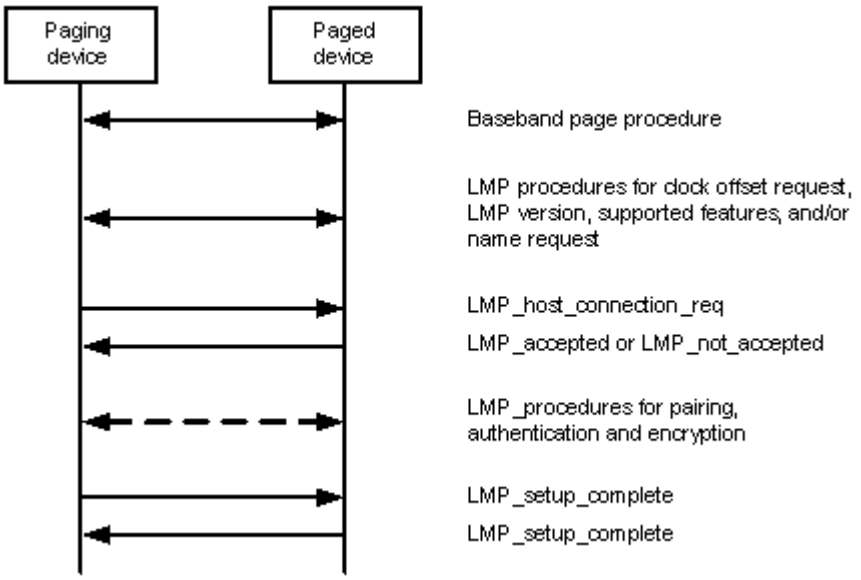


Figure 89—Connection establishment with host request

When the paging device wishes to create a connection involving layers above LM, it sends an LMP_host_connection_req PDU. When the other side receives this message, the host is informed about the incoming connection. The remote device can accept or reject the connection request by sending an LMP_accepted PDU or an LMP_not_accepted PDU. Alternatively, if the slave needs a role switch (see 9.3.4.2), it sends an LMP_slot_offset PDU and LMP_switch_req PDU after it has received an LMP_host_connection_req PDU. If the role switch fails, the LM shall continue with the creation of the connection unless this cannot be supported due to limited resources. In this case, the connection shall be terminated with an LMP_detach PDU with error code *other end terminated connection: low resources*. When the role switch has been successfully completed, the old slave will reply with an LMP_accepted PDU or an LMP_not_accepted PDU to the LMP_host_connection_req PDU (with the transaction ID set to 0).

If the paging device receives an LMP_not_accepted PDU in response to an LMP_host_connection_req PDU, it shall immediately disconnect the link using the mechanism described in 9.3.1.2.

If the LMP_host_connection_req PDU is accepted, LMP security procedures (pairing, authentication, and encryption) may be invoked. When a device is not going to initiate any more security procedures during connection establishment, it sends an LMP_setup_complete PDU. When both devices have sent LMP_setup_complete PDUs, the traffic can be transferred on the ACL-U logical transport. See Table 36.

Table 36—PDUs used for connection establishment

M/O	PDU	Contents
M	LMP_host_connection_req	—
M	LMP_setup_complete	—

9.3.1.2 Detach

The connection between two devices may be detached anytime by the master or the slave. An error code parameter is included in the message to inform the other party of why the connection is detached. See Table 37.

Table 37—PDU used for detach

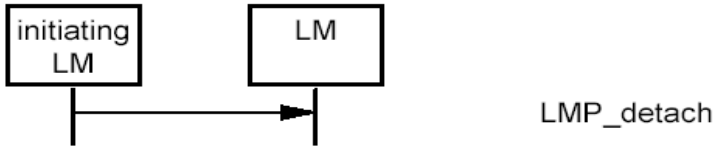
M/O	PDU	Contents
M	LMP_detach	Error code

The initiating LM shall pause traffic on the ACL-U logical link (see 8.5.3.1). The initiating LM then queues the LMP_detach PDU for transmission, and it shall start a timer for $6*T_{poll}$ slots where T_{poll} is the poll interval for the connection. If the initiating LM receives the BB acknowledgment before the timer expires, it starts a timer for $3*T_{poll}$ slots. When this timer expires and if the initiating LM is the master, the LT_ADDR(s) may be reused immediately. If the initial timer expires, then the initiating LM drops the link and starts a timer for $T_{linksupervisiontimeout}$ slots after which the LT_ADDR(s) may be reused if the initiating LM is the master.

When the receiving LM receives the LMP_detach PDU, it shall start a timer for $6*T_{poll}$ slots if it is the master and $3*T_{poll}$ if it is the slave. On timer expiration, the link shall be detached and, if the receiving LM is the master, the LT_ADDR(s) may be reused immediately. If the receiver never receives the LMP_detach PDU, then a link supervision timeout will occur, the link will be detached, and the LT_ADDR may be reused immediately.

If, at any time during this or any other LMP sequence, the link supervision timeout expires, then the link shall be terminated immediately, and the LT_ADDR(S) may be reused immediately.

If the connection is in HOLD mode, the initiating LM shall wait for the HOLD mode to end before initiating the procedure defined above (in this subclause). If the connection is in SNIFF mode or PARK state, the initiating LM shall perform the procedure to exit SNIFF mode or PARK state before initiating the procedure defined above. If the procedure to exit SNIFF mode or PARK state does not complete within the LMP response timeout (30 s), the procedure defined above shall be initiated anyway. See Sequence 1.



Sequence 1: Connection closed by sending LMP_detach.

9.3.1.3 Power control

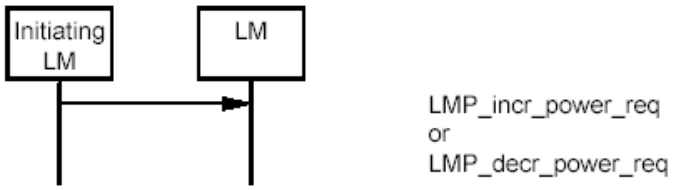
If the received signal characteristics differs too much from the preferred value of a device, it may request an increase or a decrease of the other device's TX power. The power adjustment requests may be made at any ime following a successful BB paging procedure.

If a device does not support power control requests, this is indicated in the supported features list, and thus no power control requests shall be sent after the supported features response has been processed. Prior to this time, a power control adjustment might be sent. If the recipient does not support power control, it is allowed to send an LMP_max_power PDU in response to an LMP_incr_power_req PDU and an LMP_min_power PDU in response to an LMP_decr_power_req PDU. Another possibility is to send an LMP_not_accepted PDU with the error code *unsupported LMP feature*.

Upon receipt of an LMP_incr_power_req PDU or LMP_decr_power_req PDU, the output power shall be increased or decreased one step unless this would take power above the device's maximum power level or below its minimum power level. See Clause 7 for the definition of the step size. The TX power is a property of the physical link and affects all logical transports carried over the physical link. Power control requests carried over the default ACL-C logical link shall affect only the physical link associated with the default ACL-C logical link: they shall not affect the power level used on the physical links to other slaves. See Table 38 and Sequence 2.

Table 38—PDUs used for power control

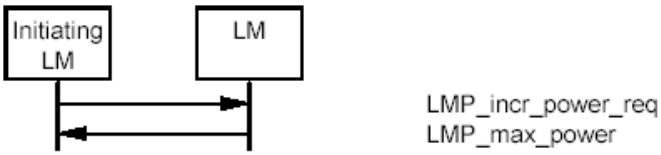
M/O	PDU	Contents
O(9)	LMP_incr_power_req	For future use (1 byte)
O(9)	LMP_decr_power_req	For future use (1 byte)
O(18)	LMP_max_power	—
O(18)	LMP_min_power	—



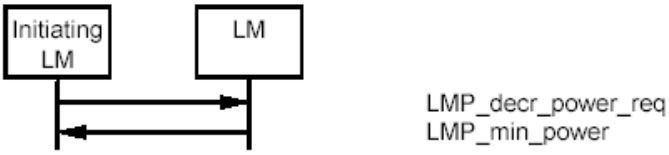
Sequence 2: A device requests a change of the other device's TX power.

If the receiver of an LMP_incr_power_req PDU is at maximum power, an LMP_max_power PDU shall be returned. The device shall request an increase again only after having requested a decrease at least once. If the receiver of an LMP_decr_power_req PDU is at minimum power, then an LMP_min_power PDU shall be returned, and the device shall request a decrease only after having requested an increase at least once. See Sequence 3 and Sequence 4.

One byte is reserved in an LMP_incr/decr_power_req PDU for future use. The parameter value shall be 0x00 and ignored upon receipt.



Sequence 3: The TX power cannot be increased.



Sequence 4: The TX power cannot be decreased.

9.3.1.4 Adaptive frequency hopping (AFH)

AFH is used to improve the performance of physical links in the presence of interference as well as reducing the interference caused by physical links on other devices in the ISM band. AFH shall be used only during the CONNECTION state. See Table 39.

Table 39—PDUs used for AFH

M/O	PDU	Contents
O(35) Rx O(43) Tx	LMP_set_AFH	AFH_Instant, AFH_Mode, AFH_Channel_Map

The LMP_set_AFH PDU contains three parameters: AFH_Instant, AFH_Mode, and AFH_Channel_Map. The AFH_Instant parameter specifies the instant at which the hopset switch will become effective. This is specified as the value of the master's clock (CLK), which is available to both devices. The AFH instant is chosen by the master and shall be an even value at least $6*T_{poll}$ or 96 slots (whichever is greater) in the future, where T_{poll} is at least the longest poll interval for all AFH-enabled physical links. The AFH instant shall be within 12 hr of the current clock value. The AFH_Mode parameter specifies whether AFH shall be enabled or disabled. The AFH_Channel_Map parameter specifies the set of channels that shall be used if AFH is enabled.

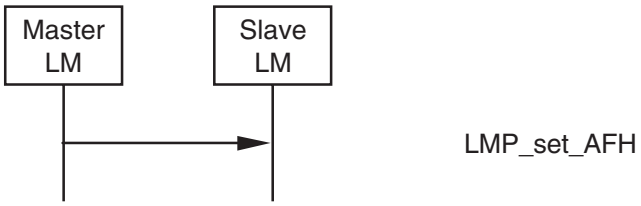
When the LMP_set_AFH PDU is received, the AFH instant shall be compared with the current CLK. If it is in the past, then the AFH instant has passed, and the slave shall immediately configure the hop selection kernel (see 8.2.6.3) with the new AFH_Mode and AFH_Channel_Map parameters specified in the LMP_set_AFH PDU. If it is in the future, then a timer shall be started to expire at the AFH instant. When this timer expires, it shall configure the hop selection kernel with the new AFH_Mode and AFH_Channel_Map parameters.

The master shall not send a new LMP_set_AFH PDU to a slave until it has received the BB acknowledgment for any previous LMP_set_AFH PDU addressed to that slave and the instant has passed.

Role switch while AFH is enabled shall follow the procedures define by 8.8.6.5.

9.3.1.4.1 Master enables AFH

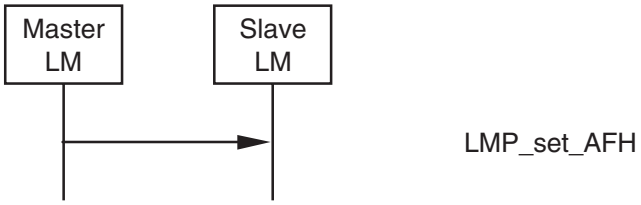
Prior to enabling AFH, the master LM shall pause traffic on the ACL-U logical link (see 8.5.3.1). The master shall then enable AFH on a physical link by sending the LMP_set_AFH PDU with the AFH_Mode parameter set to AFH_enabled, the AFH_channel_map parameter containing the set of used and unused channels, and the AFH_instant parameter set. The LM shall not calculate the AFH instant until after traffic on the ACL-U logical link has been stopped. The master considers the physical link to be AFH enabled once the BB acknowledgment has been received and the AFH instant has passed. Once the BB acknowledgment has been received, the master shall restart transmission on the ACL-U logical link. See Sequence 5.



Sequence 5: Master enables AFH.

9.3.1.4.2 Master disables AFH

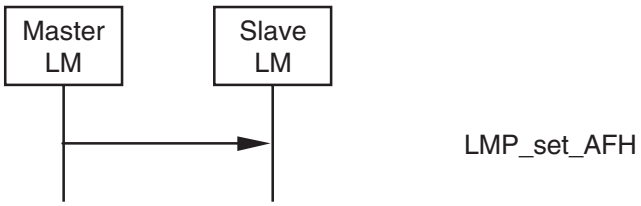
Prior to disabling AFH, the master LM shall pause traffic on the ACL-U logical link (8.5.3.1). The master shall then disable AFH operation on a physical link by sending the LMP_set_AFH PDU with the AFH_Mode parameter set to AFH_disabled and the AFH_Instant parameter set. The AFH_Channel_Map parameter is not valid when AFH mode is AFH disabled. The LM shall not calculate the AFH instant until after traffic on the ACL-U logical link has been stopped. The master considers the physical link to have entered AFH-disabled operation once the BB acknowledgment has been received and the AFH _instant has passed. Once the BB acknowledgment has been received, the master shall restart transmission on the ACL-U logical link. See Sequence 6.



Sequence 6: Master disables AFH.

9.3.1.4.3 Master updates AFH

A master shall update the AFH parameters on a physical link by sending the LMP_set_AFH PDU with the AFH_Mode parameter set to AFH_enabled, the AFH_Instant parameter set, and a new AFH_Channel_Map parameter set. The master shall consider the slave to have the updated AFH parameters once the BB acknowledgment has been received and the AFH instant has passed. See Sequence 7.



Sequence 7: Master updates AFH.

9.3.1.4.4 AFH operation in PARK, HOLD, and SNIFF

A slave in the PARK state, HOLD mode, or SNIFF mode shall retain the AFH_Mode and AFH_Channel_Map parameters prior to entering those modes. A master may change the AFH mode while a slave is in SNIFF mode.

A master that receives a request from an AFH-enabled slave to enter PARK state, HOLD mode, or SNIFF mode and decides to operate the slave using a different hop sequence shall respond with an LMP_set_AFH PDU specifying the new hop sequence.

The master continues with the LMP signalling, for park, hold, or sniff initiation, once the BB acknowledgment for the LMP_set_AFH PDU has been received. Optionally, the master may delay the continuation of this LMP signalling until after the instant. An AFH-capable slave device shall support both of these cases.

A master that receives a request from an AFH-enabled slave to enter PARK state, HOLD mode, or SNIFF mode and decides to not change the slave's hop sequence shall respond exactly as it would do without AFH. In this case, AFH operation has no effect on the LMP signalling.

9.3.1.5 Channel classification

A master may request channel classification information from a slave that is AFH enabled.

A slave that supports the AFH classification slave feature shall perform channel classification and reporting according to its AFH reporting mode. The master shall control the AFH reporting mode using the LMP_channel_classification_req PDU. The slave shall report its channel classification using the LMP_channel_classification PDU.

The slave shall report pairs of channels as good, bad, or unknown. See Table 68 for the detailed format of the AFH_Channel_Classification parameter. When one channel in the n^{th} channel pair is good and the other channel is unknown, the n^{th} channel pair shall be reported as good. When one channel in the n^{th} channel pair is bad and the other is unknown, the n^{th} channel pair shall be reported as bad. It is implementation dependent what to report when one channel in a channel pair is good and the other is bad. See Table 40.

Table 40—PDUs used for channel classification reporting

M/O	PDU	Contents
O(36) Rx O(44) Tx	LMP_channel_classification_req	AFH_Reporting_Mode, AFH_Min_Interval, AFH_Max_Interval
O(36) Tx O(44) Rx	LMP_channel_classification	AFH_Channel_Classification

The LMP_channel_classification_req PDU contains three parameters: AFH_Reporting_Mode, AFH_Min_Interval, and AFH_Max_Interval. The AFH_Min_Interval parameter defines the minimum amount of time from the last LMP_channel_classification command that was sent before the next LMP_channel_classification PDU may be sent. The AFH_Max_Interval parameter defines the maximum amount of time between the change in the radio environment being detected by a slave and its generation of an LMP_channel_classification PDU. The AFH maximum interval shall be equal to or larger than the AFH minimum interval.

The AFH_Reporting_Mode parameter shall determine if the slave is in the AFH_reporting_enabled or AFH_reporting_disabled state. The default state, prior to receipt of any LMP_channel_classification_req PDUs, shall be AFH_reporting_disabled. In the AFH_reporting_disabled state, the slave shall not generate any channel classification reports.

The AFH_reporting_mode parameter is implicitly set to the AFH_reporting_disabled state when any of the following occur:

— Establishment of a connection at the BB level
— Master-slave role switch
— Entry to PARK state operation
— Entry to HOLD mode

The AFH_reporting_mode parameter is implicitly restored to its former value when any of the following occur:

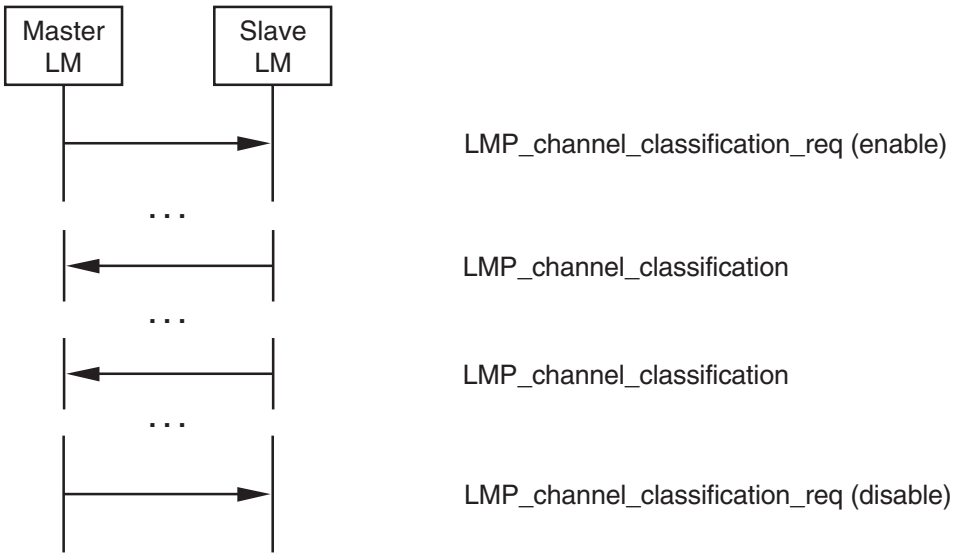
— Exit from PARK state operation
— Exit from HOLD mode
— Failure of master-slave role switch

9.3.1.5.1 Channel classification reporting enabling and disabling

A master enables slave channel classification reporting by sending the LMP_channel_classification_req PDU with the AFH_Reporting_Mode parameter set to AFH_reporting_enabled.

When a slave has had classification reporting enabled by the master, it shall send the LMP_channel_classification PDU according to the information in the latest LMP_channel_classification_req PDU. The LMP_channel_classification PDU shall not be sent if there has been no change in the slave's channel classification.

A master disables slave channel classification reporting by sending the LMP_channel_classification_req PDU with the AFH_Reporting_Mode parameter set to AFH_reporting_disabled. See Sequence 8.



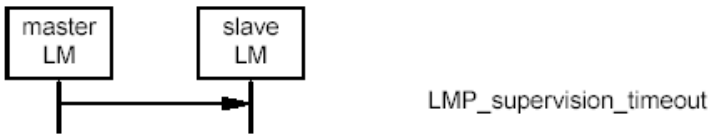
Sequence 8: Channel classification reporting.

9.3.1.6 Link supervision

Each physical link has a timer that is used for link supervision. This timer is used to detect physical link loss caused by devices moving out of range or being blocked by interference, a device's power-down, or other similar failure cases. Link supervision is specified in 8.3.1. See Table 41 and Sequence 9.

Table 41—PDU used to set the supervision timeout

M/O	PDU	Contents
M	LMP_supervision_timeout	Supervision timeout



Sequence 9: Setting the link supervision timeout.

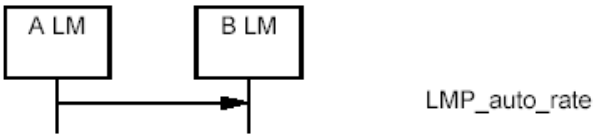
9.3.1.7 Channel quality-driven data rate (CQDDR) change

The data throughput for a given packet type depends on the quality of the RF channel. Quality measurements in the receiver of one device can be used to dynamically control the packet type transmitted from the remote device for optimization of the data throughput. Device A sends the LMP_auto_rate PDU once to notify device B to enable this feature. Once enabled, device B may change the packet type(s) that device A transmits by sending the LMP_preferred_rate PDU. This PDU has a parameter that determines the preferred coding (with or without 2/3FEC) and optionally the preferred size in slots of the packets. Device A is not required to change to the packet type specified by this parameter. Device A shall not send a packet that is larger than maximum slots (see 9.3.1.10) even if the preferred size is greater than this value.

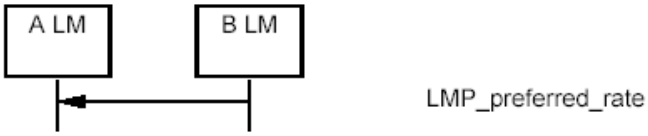
These PDUs may be sent at any time after connection setup is completed. See Table 42, Sequence 10, and Sequence 11.

Table 42—PDUs used for quality-driven change of data rate

M/O	PDU	Contents
O(10)	LMP_auto_rate	—
O(10)	LMP_preferred_rate	Data rate



Sequence 10: Device A notifies device B to enable CQDDR.



Sequence 11: Device B sends device A a preferred packet type.

9.3.1.8 Quality of service (QoS)

The LM provides QoS capabilities. A poll interval, T_{poll}, which is defined as the maximum time between transmissions from the master to a particular slave on the ACL logical transport, is used to support bandwidth allocation and latency control (see 8.8.6.1 for details). The poll interval is guaranteed in the active and SNIFF modes except when there are collisions with page, page scan, inquiry, and inquiry scan; during time-critical LMP sequences in the current piconet and any other piconets in which the device is a member; and during critical BB sequences (such as the page response, initial CONNECTION state until the first POLL, and master-slave switch). These PDUs maybe sent at anytime after connection setup is completed.

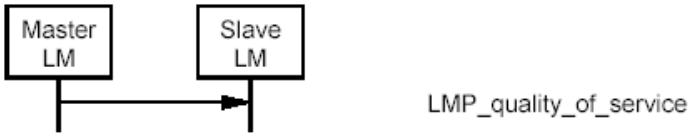
Master and slave negotiate the number of repetitions for broadcast packets (N_{BC}) (see 8.7.6.5). See Table 43.

Table 43— PDUs used for QoS

M/O	PDU	Contents
M	LMP_quality_of_service	Poll interval N_{BC}
M	LMP_quality_of_service_req	Poll interval N_{BC}

9.3.1.8.1 Master notifies slave of the QoS

The master notifies the slave of the new poll interval and N_{BC} by sending the LMP_quality_of_service PDU. The slave cannot reject the notification. See Sequence 12.

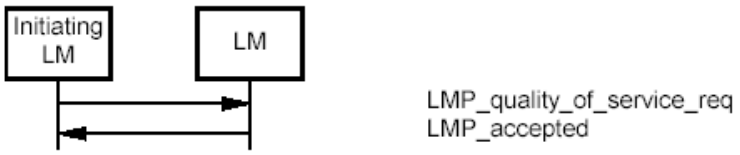


Sequence 12: Master notifies slave of QoS.

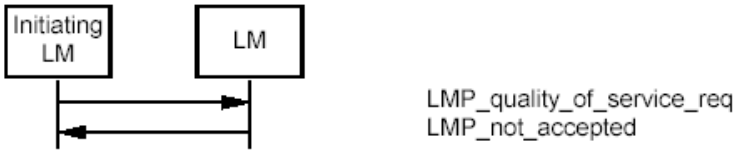
9.3.1.8.2 Device requests new QoS

Either the master or the slave may request a new poll interval and N_{BC} by sending an LMP_quality_of_service_req PDU. The parameter N_{BC} is meaningful only when it is sent by a master to a slave. For transmission of LMP_quality_of_service_req PDUs from a slave, this parameter shall be ignored by the master. The request can be accepted or rejected. This allows the master and slave to dynamically negotiate the QoS as needed.

The selected poll interval by the slave shall be less than or equal to the specified access latency for the outgoing traffic of the ACL link (see 14.5.3). See Sequence 13 and Sequence 14.



Sequence 13: Device accepts new QoS.



Sequence 14: Device rejects new QoS.

9.3.1.9 Paging scheme parameters

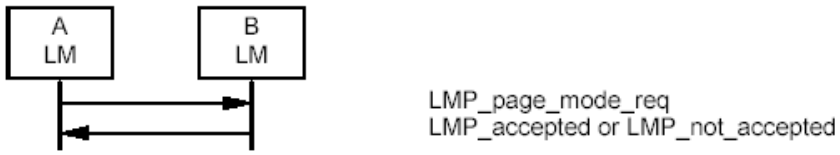
LMP provides a means to negotiate the paging scheme parameters that are used the next time a device is paged. See Table 44.

Table 44—PDUs used to request paging scheme

M/O	PDU	Contents
O(17)	LMP_page_mode_req	Paging scheme Paging scheme settings
O(17)	LMP_page_scan_mode_req	Paging scheme Paging scheme settings

9.3.1.9.1 Page mode

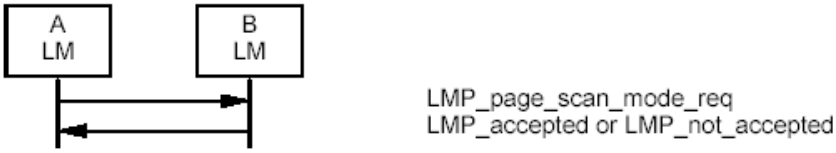
This procedure is initiated from device A and negotiates the paging scheme used when device A pages device B. Device A proposes a paging scheme, including the parameters for this scheme, and device B can accept or reject. On rejection, the old setting will not be changed. A request to switch to a reserved paging scheme shall be rejected. See Sequence 15.



Sequence 15: Negotiation for page mode.

9.3.1.9.2 Page scan mode

This procedure is initiated from device A and negotiates the paging scheme and paging scheme settings used when device B pages device A. Device A proposes a paging scheme and paging scheme settings, and device B may accept or reject. On rejection, the old setting is not changed. A request specifying the mandatory scheme shall be accepted. A request specifying a nonmandatory scheme shall be rejected. This procedure should be used when device A changes its paging scheme settings. A slave should also send this message to the master after connection establishment to inform the master of the slave's current paging scheme and paging scheme settings. See Sequence 16.



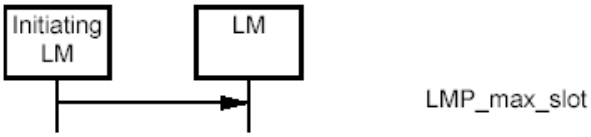
Sequence 16: Negotiation for page scan mode.

9.3.1.10 Control of multislot packets

The number of consecutive slots used by a device on an ACL-U logical link can be limited. It does not affect traffic on the eSCO links where the packet sizes are defined as part of link setup. A device allows the remote device to use a maximum number of slots by sending the LMP_max_slot PDU providing the maximum slots as a parameter. Each device can request to use a maximal number of slots by sending the LMP_max_slot_req PDU providing the maximum slots as a parameter. After a new connection, as a result of page, page scan, role switch, or unpark, the default value is 1 slot. These PDUs can be sent at any time after connection setup is completed. See Table 45, Sequence 17, Sequence 18, and Sequence 19.

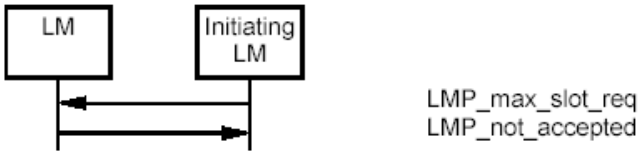
Table 45—PDUs used to control the use of multislot packets

M/O	PDU	Contents
M	LMP_max_slot	Maximum slots
M	LMP_max_slot_req	Maximum slots



Sequence 17: Device allows remote device to use a maximum number of slots.

Sequence 18: Device requests a maximum number of slots. Remote device accepts.



Sequence 19: Device requests a maximum number of slots. Remote device rejects.

9.3.2 Security

9.3.2.1 Authentication

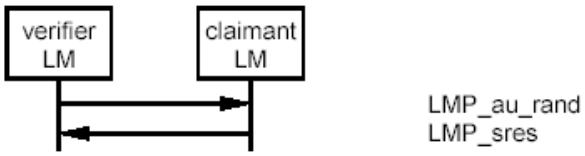
The authentication procedure is based on a challenge-response scheme as described in 13.5. The verifier sends an LMP_au_rand PDU that contains a random number (the challenge) to the claimant. The claimant calculates a response that is a function of this challenge, the claimant's BD_ADDR, and a secret key. The response is sent back to the verifier, which checks if the response was correct or not. The response shall be calculated as described in 13.6.1. A successful calculation of the authentication response requires that two devices share a secret key. This key is created as described in 9.3.2.2. Both the master and the slave can be verifiers. See Table 46.

Table 46—PDUs used for authentication

M/O	PDU	Contents
M	LMP_au_rand	Random number
M	LMP_sres	Authentication response

9.3.2.1.1 Claimant has link key

If the claimant has a link key associated with the verifier, it shall calculate the response and send it to the verifier with an LMP_sres PDU. The verifier checks the response. If the response is not correct, the verifier can end the connection by sending an LMP_detach PDU with the error code *authentication failure* (see 9.3.1.2). See Sequence 20.



Sequence 20: Authentication. Claimant has link key.

Upon reception of an LMP_au_rand PDU, an LM shall reply with an LMP_sres PDU before initiating its own authentication.

NOTE—There can be concurrent requests caused by the master and slave simultaneously initiating an authentication. The procedure in 9.1.5.1 assures that devices will not have different authenticated cyphering offset (ACO) (see 13.6.1) when they calculate the encryption key.

9.3.2.1.2 Claimant has no link key

If the claimant does not have a link key associated with the verifier, it shall send an LMP_not_accepted PDU with the error code *key missing* after receiving an LMP_au_rand PDU. See Sequence 21.

Sequence 21: Authentication fails. Claimant has no link key.

9.3.2.1.3 Repeated attempts

The scheme described in 13.5.1 shall be applied when an authentication fails. This will prevent an intruder from trying a large number of keys in a relatively short time.

9.3.2.2 Pairing

When two devices do not have a common link key, an initialization key (K_{init}) shall be created based on a personal identification number (PIN), a random number, and a BD_ADDR. K_{init} shall be created as specified in 13.6.3. When both devices have calculated K_{init}, the link key shall be created, and a mutual authentication is performed. The pairing procedure starts with a device sending an LMP_in_rand PDU; this device is referred to as the *initiating LM* or *initiator* in 9.3.2.2.1 and 9.3.2.2.5. The other device is referred to as the *responding LM* or *responder*. The PDUs used in the pairing procedure are listed in Table 47.

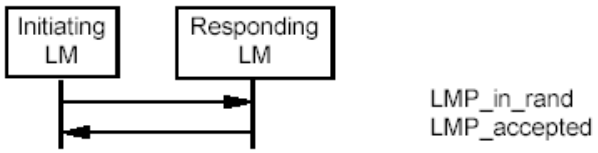
Table 47—PDUs used for pairing

M/O	PDU	Contents
M	LMP_in_rand	Random number
M	LMP_au_rand	Random number
M	LMP_sres	Authentication response
M	LMP_comb_key	Random number
M	LMP_unit_key	Key

All sequences described in 9.3, including the mutual authentication after the link key has been created, shall form a single transaction. The transaction ID from the first LMP_in_rand PDU shall be used for all subsequent sequences.

9.3.2.2.1 Responder accepts pairing

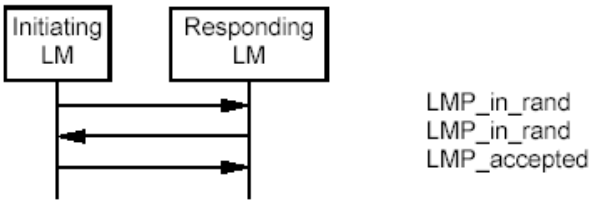
When the initiator sends an LMP_in_rand PDU, the responder shall reply with an LMP_accepted PDU. Both devices shall then calculate K_{init} based on the BD_ADDR of the responder, and the procedure continues with creation of the link key (see 9.3.2.2.4). See Sequence 22.



Sequence 22: Pairing accepted. Responder has a variable PIN, and initiator has a variable or fixed PIN.

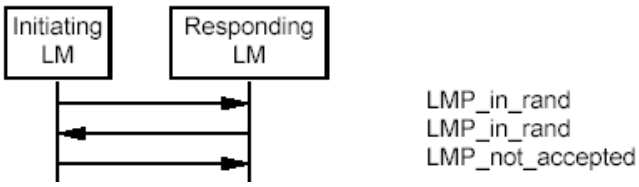
9.3.2.2.2 Responder has a fixed PIN

If the responder has a fixed PIN and accepts pairing, it shall generate a new random number and send it back in an LMP_in_rand PDU. If the initiator has a variable PIN, it shall accept the LMP_in_rand PDU and shall respond with an LMP_accepted PDU. Both sides shall then calculate K_{init} based on the last IN_RAND and the BD_ADDR of the initiator. The procedure continues with creation of the link key (see 9.3.2.2.4). See Sequence 23.



Sequence 23: Responder has a fixed PIN, and initiator has a variable PIN.

If the responder has a fixed PIN and the initiator also has a fixed PIN, the second LMP_in_rand PDU shall be rejected by the initiator sending an LMP_not_accepted PDU with the error code *pairing not allowed*. See Sequence 24.



Sequence 24: Both devices have a fixed PIN.

9.3.2.2.3 Responder rejects pairing

If the responder rejects pairing, it shall send an LMP_not_accepted PDU with the error code *pairing not allowed* after receiving an LMP_in_rand PDU. See Sequence 25.

Sequence 25: Responder rejects pairing.

9.3.2.2.4 Creation of the link key

When K_{init} is calculated in both devices, the link key shall be created. This link key will be used in the authentication between the two devices for all subsequent connections until it is changed (see 9.3.2.3 and 9.3.2.4). The link key created in the pairing procedure will be either a combination key or one of the device's unit keys. The following rules shall apply to the selection of the link key:

— If one device sends an LMP_unit_key PDU and the other device sends LMP_comb_key PDU, the unit key will be the link key.
— If both devices send an LMP_unit_key PDU, the master's unit key will be the link key.
— If both devices send an LMP_comb_key PDU, the link key shall be calculated as described in 13.3.2.

The content of the LMP_unit_key PDU is the unit key bitwise XORed with K_{init}. The content of the LMP_comb_key PDU is LK_RAND bitwise XORed with K_{init}. Any device configured to use a combination key shall store the link key.

The use of unit keys is deprecated since it is implicitly insecure.

When the link key (i.e., combination or unit key) has been created, mutual authentication shall be performed to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication is performed with the initiator as the verifier. When finalized, an authentication in the reverse direction is performed. See Sequence 26.

Sequence 26: Creation of the link key.

9.3.2.2.5 Repeated attempts

When the authentication after creation of the link key fails because of an incorrect authentication response, the same scheme as in 9.3.2.1.3 shall be used. This prevents an intruder from trying a large number of different PINs in a relatively short time.

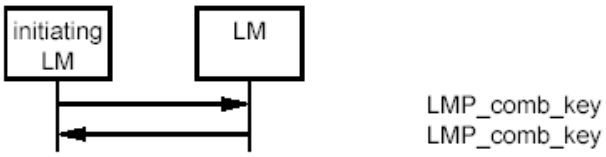
9.3.2.3 Change link key

If the link key is derived from combination keys and the current link key is the semi-permanent link key, the link key can be changed. If the link key is a unit key, the devices shall go through the pairing procedure in order to change the link key. The contents of the LMP_comb_key PDU is protected by a bitwise XOR with the current link key.

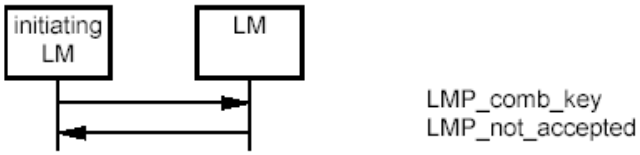
All sequences described in 9.3.2.3, including the mutual authentication after the link key has been changed, shall form a single transaction. The transaction ID from the first LMP_comb_key PDU shall be used for all subsequent sequences. See Table 48, Sequence 27, and Sequence 28.

Table 48—PDUs used to change link key

M/O	PDU	Contents
M	LMP_comb_key	Random number



Sequence 27: Successful change of the link key.



Sequence 28: Change of the link key not possible since the other device uses a unit key.

If the change of link key is successful, the new link key shall be stored, and the old link key shall be discarded. The new link key shall be used as link key for all the following connections between the two devices until the link key is changed again. The new link key also becomes the current link key. It will remain the current link key until the link key is changed again or until a temporary link key is created (see 9.3.2.4).

When the new link key has been created, mutual authentication shall be performed to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication is performed with the device that initiated the link key change as verifier. When finalized, an authentication in the reverse direction is performed.

9.3.2.4 Change current link key type

The current link key can be a semi-permanent link key or a temporary link key. It may be changed temporarily, but the change shall be valid only for the current connection (see 13.3.1). Changing to a temporary link key is necessary if the piconet is to support encrypted broadcast. The current link key may not be changed before the connection establishment procedure has completed. This feature is supported only if broadcast encryption is supported as indicated by the LMP features mask. See Table 49.

Table 49—PDUs used to change current link key

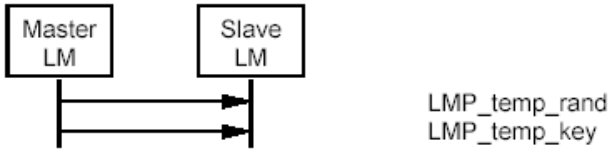
M/O	PDU	Contents
O(23)	LMP_temp_rand	Random number
O(23)	LMP_temp_key	Key
O(23)	LMP_use_semi_permanent_key	—

9.3.2.4.1 Change to a temporary link key

The master starts by creating the master key K_{master} as specified in Equation (23). Then the master shall generate a random number, RAND, and shall send it to the slave in an LMP_temp_rand PDU. Both sides then calculate an overlay denoted OVL as OVL = E_{22}(current link key, RAND, 16). The master shall then send K_{master} protected by a modulo-2 addition with OVL to the slave in an LMP_temp_key PDU. The slave

calculates K_{master}, based on OVL, which becomes the current link key. It shall be the current link key until the devices fall back to the semi-permanent link key (see 9.3.2.4.2). See Sequence 29.

NOTE—The terminology in this subclause is the same as used in 13.3.2.8.



Sequence 29: Change to a temporary link key.

All sequences described in 9.3.2.4.1, including the mutual authentication after K_{master} has been created, shall form a single transaction. The transaction ID shall be set to 0.

When the devices have changed to the temporary key, a mutual authentication shall be made to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication shall be performed with the master as verifier. When finalized, an authentication in the reverse direction is performed.

Should the mutual authentication fail at either side, the LM of the verifier should start the detach procedure. Because authentication is mutual, having the verifier initiate a detach will ensure the detach occurs if one of the devices is erroneous

9.3.2.4.2 Make the semi-permanent link key the current link key

After the current link key has been changed to K_{master}, this change can be undone, and the semi-permanent link key becomes the current link key again. If encryption is used on the link, the procedure to go back to the semi-permanent link key shall be immediately followed by the procedure where the master stops encryption (see 9.3.2.5.4). Encryption may be restarted by the master according to the procedures in 9.3.2.5.1. This is to assure that encryption with encryption parameters known by other devices in the piconet is not used when the semi-permanent link key is the current link key. See Sequence 30.

Sequence 30: Link key changed to the semi-permanent link key.

9.3.2.5 Encryption

If at least one authentication has been performed, encryption may be used. If the master intends to broadcast encrypted data, then it must use the same encryption parameters for all slaves in the piconet. In this case, it shall issue a temporary key, K_{master}, and shall make this key the current link key for all slaves in the piconet before encryption is started (see 9.3.2.4). See Table 50.

All sequences described in 9.3.2.5 shall form a single transaction. The transaction ID from the LMP_encryption_mode_req PDU shall be used for all subsequent sequences.

Table 50—PDUs used for handling encryption

M/O	PDU	Contents
O	LMP_encryption_mode_req	Encryption mode
O	LMP_encryption_key_size_req	Key size
O	LMP_start_encryption_req	Random number
O	LMP_stop_encryption_req	—

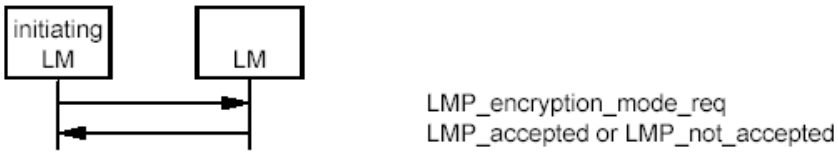
9.3.2.5.1 Encryption mode

The master and the slave must agree upon whether to use encryption (encryption mode = 1 in the LMP_encryption_mode_req PDU) or not (encryption mode = 0). If the semi-permanent key is used (Key_Flag = 0x00), encryption shall apply only to point-to-point packets. If the master link key is used (Key_Flag = 0x01), encryption shall apply to both point-to-point packets and broadcast packets. If master and slave agree on the encryption mode, the master continues to give more detailed information about the encryption.

Devices should never send an LMP_encryption_mode_req PDU with an encryption mode value of 2; however, for backwards compatibility, if the LMP_encryption_mode_req PDU is received with an encryption mode value of 2, then it should be treated the same as an encryption mode value of 1.

The initiating LM shall pause traffic on the ACL-U logical link (see 8.5.3.1). The initiating device shall then send the LMP_encryption_mode_req PDU. If the responding device accepts the change in encryption mode, then it shall complete the transmission of the current packet on the ACL logical transport and shall then suspend transmission on the ACL-U logical link. The responding device shall then send the LMP_accepted PDU.

ACL-U logical link traffic shall be resumed only after the attempt to encrypt or decrypt the logical transport is completed, i.e., at the end of Sequence 31, Sequence 32, or Sequence 33.



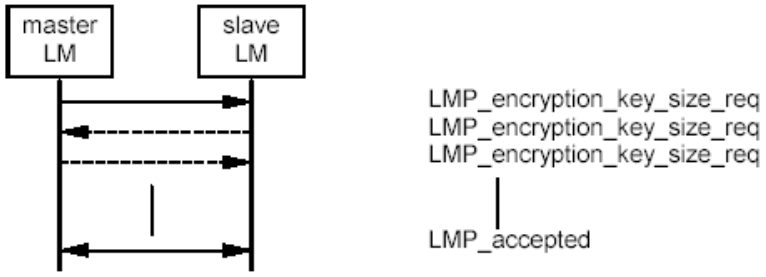
Sequence 31: Negotiation for encryption mode.

After a device has sent an LMP_encryption_mode_req PDU, it shall not send an LMP_au_rand PDU before encryption is started. After a device has received an LMP_encryption_mode_req PDU and sent an LMP_accepted PDU, it shall not send an LMP_au_rand PDU before encryption is started. If an LMP_au_rand PDU is sent violating these rules, the claimant shall respond with an LMP_not_accepted PDU with the error code *PDU not allowed*. This assures that devices will not have different ACOs when they calculate the encryption key. If the encryption mode is not accepted or the encryption key size negotiation results in disagreement, the devices may send an LMP_au_rand PDU again.

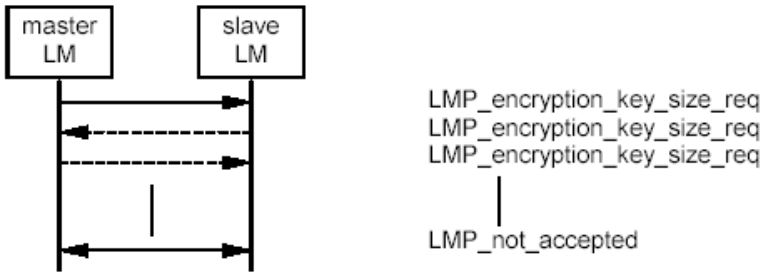
9.3.2.5.2 Encryption key size

NOTE—This subclause uses the same terms as in 13.4.1.

The master sends an LMP_encryption_key_size_req PDU including the suggested key size $L_{sug,\ m}$, that is initially equal to $L_{max,\ m}$. If $L_{min,\ s} \leq L_{sug,\ m}$ and the slave supports $L_{sug,\ m}$, it shall respond with an LMP_accepted PDU, and $L_{sug,\ m}$ shall be used as the key size. If both conditions are not fulfilled, the slave sends back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug,\ s}$. This value shall be the slave's largest supported key size that is less than $L_{sug,\ m}$. Then the master performs the corresponding test on the slave's suggestion. This procedure is repeated until a key size agreement is reached or it becomes clear that no such agreement can be reached. If an agreement is reached, a device sends an LMP_accepted PDU, and the key size in the last LMP_encryption_key_size_req PDU shall be used. After this, encryption is started (see 9.3.2.5.3). If an agreement is not reached, a device sends an LMP_not_accepted PDU with the error code *unsupported parameter value*, and the devices shall not communicate using encryption. See Sequence 32 and Sequence 33.



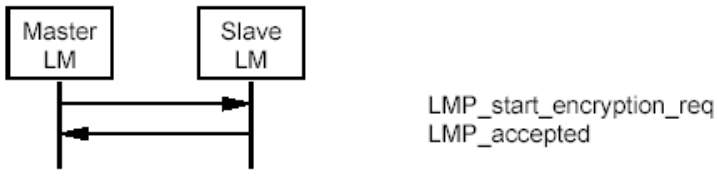
Sequence 32: Encryption key size negotiation successful.



Sequence 33: Encryption key size negotiation failed.

9.3.2.5.3 Start encryption

To start encryption, the master issues the random number EN_RAND and calculates the encryption key. See 13.3.2.5. The random number shall be the same for all slaves in the piconet when broadcast encryption is used. The master issues EN_RAND by sending an LMP_start_encryption_req PDU including EN_RAND. The slave shall calculate the encryption key when this message is received and shall acknowledge with an LMP_accepted PDU. See Sequence 34.



Sequence 34: Start of encryption.

Starting encryption shall be performed in three steps:

a) Master is configured to transmit unencrypted packets and to receive encrypted packets.
b) Slave is configured to transmit and receive encrypted packets.
c) Master is configured to transmit and receive encrypted packets.

Between step a and step b, master-to-slave transmission is possible. This is when an LMP_start_ encryption_req PDU is transmitted. Step b is triggered when the slave receives this message. Between step b and step c, slave-to-master transmission is possible. This is when an LMP_accepted PDU is transmitted. Step c is triggered when the master receives this message.

9.3.2.5.4 Stop encryption

To stop encryption, a device shall send an LMP_encryption_mode_req PDU with the parameter encryption mode equal to 0 (no encryption). The other device responds with an LMP_accepted PDU or an LMP_not_accepted PDU (the procedure is described in Sequence 31 in 9.3.2.5.1). If accepted, encryption shall be stopped when the master sends an LMP_stop_encryption_req PDU, and the slave shall respond with an LMP_accepted PDU according to Sequence 35.

Sequence 35: Stop of encryption.

Stopping encryption shall be performed in three steps, similar to the procedure for starting encryption.

a) Master is configured to transmit encrypted packets and to receive unencrypted packets.
b) Slave is configured to transmit and receive unencrypted packets.
c) Master is configured to transmit and receive unencrypted packets.

Between step a and step b, master-to-slave transmission is possible. This is when an LMP_stop_ encryption_req PDU is transmitted. Step b is triggered when the slave receives this message. Between step b and step c, slave-to-master transmission is possible. This is when an LMP_accepted PDU is transmitted. Step c is triggered when the master receives this message.

9.3.2.5.5 Change encryption mode, key, or random number

If the encryption key or encryption random number need to be changed or if the current link key needs to be changed according to the procedures in 9.3.2.4, encryption shall be stopped and restarted after completion, using the procedures in 9.3.2.5.3 and 9.3.2.5.4, for the new parameters to take effect.

NOTE—Because the ACL-C channel has priority over the ACL-U channel, it is possible for data to be queued up in the protocol stack at the point when encryption is stopped. Such data could then be sent in the clear between encryption being stopped and restarted.

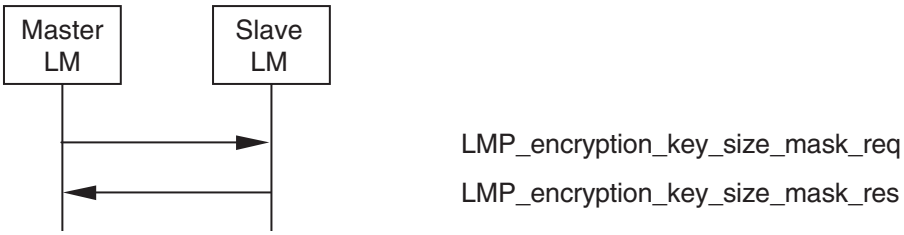
When broadcast encryption is supported via the LMP features mask, it is possible for the master to request a slave's supported encryption key sizes. See Table 51.

The master shall send an LMP_key_size_req PDU to the slave to obtain the slaves supported encryption key sizes.

Table 51—PDUs used for encryption key size request

M/O	PDU	Contents
O(23)	LMP_encryption_key_size_mask_req	
O(23)	LMP_encryption_key_size_mask_res	Key size mask

The slave shall return a bit mask indicating all broadcast encryption key sizes supported. The LSB shall indicate support for a key size of 1, the next MSB shall indicate support for a key size of 2 ,and so on up to a key size of 16. In all cases, a bit set to 1 shall indicate support for a key size; a bit set to 0 shall indicate that the key size is not supported. See Sequence 36.



Sequence 36: Request for supported encryption key sizes.

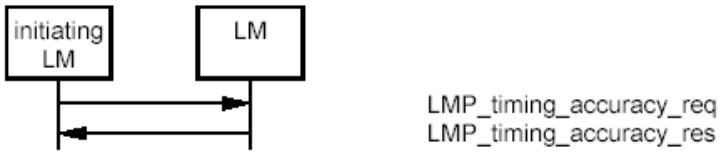
9.3.3 Informational requests

9.3.3.1 Timing accuracy

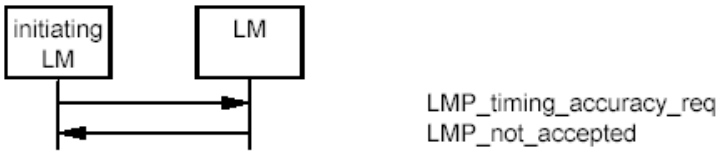
LMP supports requests for the timing accuracy. This information can be used to minimize the scan window during piconet physical channel resynchronization (see 8.2.2.5.2). The timing accuracy parameters returned are the long-term drift measured in parts per million and the long-term jitter measured in microseconds of the worst-case clock used. These parameters are fixed for a certain device and shall be identical when requested several times. Otherwise, the requesting device shall assume worst-case values (drift = 250 ppm and jitter = 10 μs). See Table 52, Sequence 37, and Sequence 38.

Table 52—PDUs used for requesting timing accuracy information

M/O	PDU	Contents
O(4)	LMP_timing_accuracy_req	—
O(4)	LMP_timing_accuracy_res	Drift Jitter



Sequence 37: The requested device supports timing accuracy information.



Sequence 38: The requested device does not support timing accuracy information.

9.3.3.2 Clock offset

The clock offset can be used to speed up the paging time the next time the same device is paged. The master can request the clock offset at anytime following a successful BB paging procedure (i.e., before, during, or after connection setup). The clock offset shall be defined by the following equation:

$(CLKN_{16-2\ slave} - CLKN_{16-2\ master})$ mod 2**15.

See Table 53 and Sequence 39.

Table 53—PDUs used for clock offset request

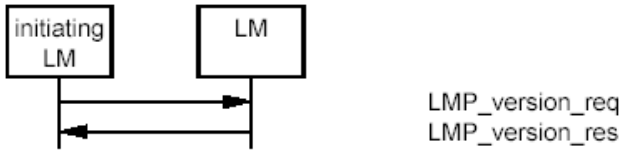
M/O	PDU	Contents
M	LMP_clkoffset_req	—
M	LMP_clkoffset_res	Clock offset

Sequence 39: Clock offset requested.

9.3.3.3 LMP version

LMP supports requests for the version of the LMP. The LMP_version_req and LMP_version_res PDUs contain three parameters: VersNr, CompId, and SubVersNr. VersNr specifies the version of the IEEE 802.15.1-2005 LMP specification that the device supports. CompId is used to track possible problems with the lower layers. All companies that create a unique implementation of the LM shall have their own

CompId. The same company is also responsible for the administration and maintenance of the SubVersNr. It is recommended that each company have a unique SubVersNr for each RF/BB/LM implementation. For a given VersNr and CompId, the values of the SubVersNr shall increase each time a new implementation is released. For both CompId and SubVersNr, the value 0xFFFF means that no valid number applies. There is no ability to negotiate the version of the LMP. Sequence 40 is used only to exchange the parameters. LMP version can be requested at any time following a successful BB paging procedure. See Table 54.



Sequence 40: Request for LMP version.

Table 54—PDUs used for LMP version request

M/O	PDU	Contents
M	LMP_version_req	VersNr CompId SubVersNr
M	LMP_version_res	VersNr CompId SubVersNr

9.3.3.4 Supported features

The supported features may be requested at any time following a successful BB paging procedure by sending the LMP_features_req PDU. Upon reception of an LMP_features_req PDU, the receiving device shall return an LMP_features_res PDU.

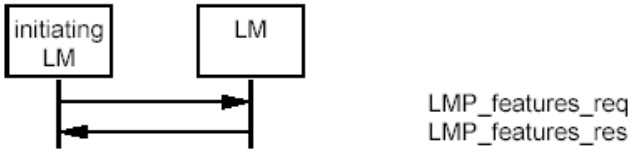
The number of features bits required may in the future exceed the size of a single page of features. An extended features mask is, therefore, provided to allow support for more than 64 features. Support for the extended features mask is indicated by the presence of the appropriate bit in the LMP features mask. The LMP_features_req_ext and LMP_features_res_ext PDUs operate in precisely the same way as the LMP_features_req and LMP_features_res PDUs except that they allow the various pages of the extended features mask to be requested. The LMP_features_req_ext PDU may be sent at any time following the exchange of the LMP_features_req and LMP_features_rsp PDUs.

The LMP_features_req_ext PDU contains a feature page index that specifies which page is requested and the contents of that page for the requesting device. Pages are numbered from 0–255 with page 0 corresponding to the normal features mask. Each page consists of 64 bits. If a device supports the LMP_features_res_ext PDU, but does not support any page number, it shall return a mask with every bit set to 0. If a device does not support the LMP_features_res_ext PDU, it shall return an LMP_not_accepted PDU. It also contains the maximum features page number containing any nonzero bit for this device. The recipient of an LMP_features_req_ext PDU shall respond with an LMP_features_res_ext PDU containing the same page number and the appropriate features page along with its own maximum features page number.

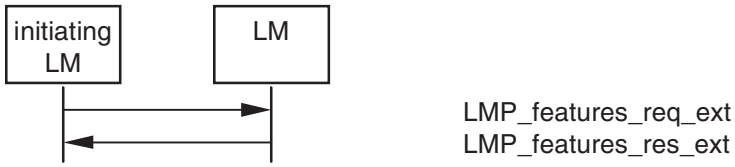
If the extended features request is not supported, then all bits in all extended features pages for that device shall be assumed to be zero. See Table 55, Sequence 41, and Sequence 42.

Table 55—PDUs used for features request

M/O	PDU	Contents
M	LMP_features_req	Features
M	LMP_features_res	Features
O(63)	LMP_features_req_ext	Features page Maximum supported page Extended features
O(63)	LMP_features_res_ext	Features page Maximum supported page Extended features



Sequence 41: Request for supported features.



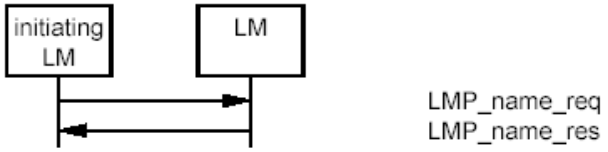
Sequence 42: Request for extended features.

9.3.3.5 Name request

LMP supports name request to another device. The name is a user-friendly name associated with the device and consists of a maximum of 248 bytes coded according to the UTF-8 standard. The name is fragmented over one or more **DM1** packets. When an LMP_name_req PDU is sent, a name offset indicates which fragment is expected. The corresponding LMP_name_res PDU carries the same name offset, the name length indicating the total number of bytes in the name of the device, and the name fragment, where

— Name fragment(N) = name(N + name offset), if (N + name offset) < name length
— Name fragment(N) = 0, otherwise.

Here $0 \le N \le 13$. In the first sent LMP_name_req PDU, name offset = 0. Sequence 43 is then repeated until the initiator has collected all fragments of the name. The name request may be made at any time following a successful BB paging procedure. See Table 56.



Sequence 43: Device's name requested and responses.

Table 56—PDUs used for name request

M/O	PDU	Contents
M	LMP_name_req	Name offset
M	LMP_name_res	Name offset Name length Name fragment

9.3.4 Role switch

This subclause describes the LMP sequences used for role switch including the slot offset command used to retrieve timing information and the role switch commands used to reverse roles.

9.3.4.1 Slot offset

With the LMP_slot_offset PDU, the information about the difference between the slot boundaries in different piconets is transmitted. The LMP_slot_offset PDU may be sent anytime after the BB paging procedure has completed. This PDU carries the parameters slot offset and BD_ADDR. The slot offset shall be the time in microseconds between the start of a master transmission in the current piconet to the start of the next following master transmission in the piconet where the BD_ADDR device (normally the slave) is master at the time that the request is interpreted by the BD_ADDR device. See Figure 90.

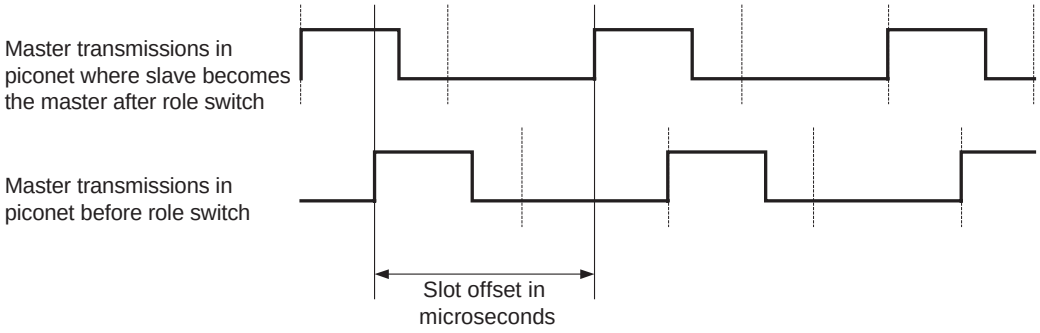


Figure 90—Slot offset for role switch

See 9.3.4 for the use of the LMP_slot_offset PDU in the context of the role switch. In the case of role switch, the BD_ADDR is that of the slave device. See Table 57 and Sequence 44.

Table 57— PDU used for slot offset information

M/O	PDU	Contents
O(3)	LMP_slot_offset	Slot offset BD_ADDR

Sequence 44: Slot offset information is sent.

9.3.4.2 Role switch

Since the paging device always becomes the master of the piconet, a switch of the master-slave role is sometimes needed (see 8.8.6.5). The LMP_switch_req PDU may be sent anytime after the BB paging procedure has completed.

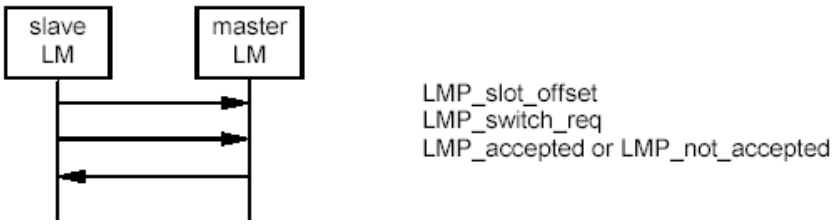
Support for the LMP_slot_offset PDU is mandatory if the LMP_switch_req PDU is supported.

The LMP_slot_offset PDU shall be sent only if the ACL logical transport is in active mode. The LMP_switch_req PDU shall be sent only if the ACL logical transport is in active mode, when encryption is disabled, and all synchronous logical transports on the same physical link are disabled. Additionally, the LMP_slot_offset or LMP_switch_req PDU shall not be initiated or accepted while a synchronous logical transport is being negotiated by the LM. See Table 58.

Table 58—PDUs used for role switch

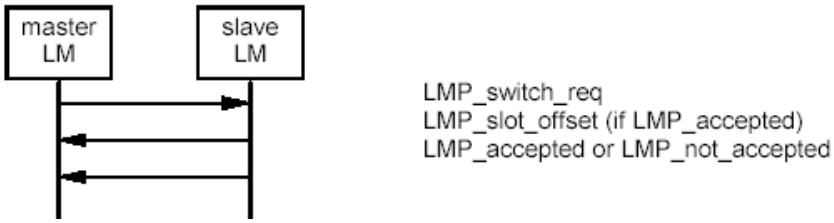
M/O	PDU	Contents
O(5)	LMP_switch_req	Switch instant
O(5)	LMP_slot_offset	Slot offset BD_ADDR

The initiating LM shall pause traffic on the ACL-U logical link (see 8.5.3.1). It shall then send an LMP_slot_offset PDU immediately followed by an LMP_switch_req PDU. If the master accepts the role switch, it shall pause traffic on the ACL-U logical link (see 8.5.3.1) and respond with an LMP_accepted PDU. When the role switch has been completed at the BB level (successfully or not), both devices reenable transmission on the ACL-U logical link. If the master rejects the role switch, it responds with an LMP_not_accepted PDU, and the slave reenables transmission on the ACL-U logical link. The transaction ID for all PDUs in the sequence shall be set to 1. See Sequence 45.



Sequence 45: Role switch (slave initiated).

If the master initiates the role switch, it shall pause traffic on the ACL-U logical link (see 8.5.3.1) and send an LMP_switch_req PDU. If the slave accepts the role switch, it shall pause traffic on the ACL-U logical link (see 8.5.3.1) and responds with an LMP_slot_offset PDU immediately followed by an LMP_accepted PDU. When the role switch has been completed at the BB (successfully or not), both devices reenable transmission on the ACL-U logical link. If the slave rejects the role switch, it responds with an LMP_not_accepted PDU, and the master reenables transmission on the ACL-U logical link. The transaction ID for all PDUs in the sequence shall be set to 0. See Sequence 46.



Sequence 46: Role switch (master initiated).

The LMP_switch_req PDU contains a parameter, switch instant, that specifies the instant at which the TDD switch is performed. This is specified as CLK, which is available to both devices. This instant is chosen by the sender of the message and shall be at least $2*T_{poll}$ or 32 (whichever is greater) slots in the future. The switch instant shall be within 12 hr of the current clock value to avoid clock wrap.

The sender of the LMP_switch_req PDU selects the switch instant, queues the LMP_switch_req PDU to link control for transmission, and starts a timer to expire at the switch instant. When the timer expires, it initiates the mode switch. In the case of a master-initiated switch, if the LMP_slot_offset PDU has not been received by the switch instant, the role switch is carried out without an estimate of the slave's slot offset. If an LMP_not_accepted PDU is received before the timer expires, then the timer is stopped, and the role switch shall not be initiated.

When the LMP_switch_req is received, the switch instant is compared with the CLK. If it is in the past, then the instant has been passed, and an LMP_not_accepted PDU with the error code *instant passed* shall be returned. If it is in the future and the role switch is allowed, then an LMP_accepted PDU shall be returned, and a timer is started to expire at the switch instant. When this timer expires, the role switch shall be initiated.

After a successful role switch, the supervision timeout and poll interval T_{poll} shall be set to their default values. The authentication state and the ACO shall remain unchanged. AFH shall follow the procedures described in 8.8.6.5. The default value for the max_slots parameter shall be used.

9.3.5 Modes of operation

This subclause describes the LMP sequences required to put an active link into HOLD mode, PARK state, or SNIFF mode, along with the LMP sequences used to communicate with slaves in PARK state.

9.3.5.1 HOLD mode

The ACL logical transport of a connection between two devices can be placed in HOLD mode for a specified hold time. See 8.8.8 for details. See also Table 59.

Table 59—PDUs used for HOLD mode

M/O	PDU	Contents
O(6)	LMP_hold	Hold time, hold instant
O(6)	LMP_hold_req	Hold time, hold instant

The LMP_hold and LMP_hold_req PDUs both contain a parameter, hold instant, that specifies the instant at which the HOLD mode becomes effective. This is specified as CLK, which is available to both devices. The hold instant is chosen by the sender of the message and should be at least $6*T_{poll}$ slots in the future. The hold instant shall be within 12 hr of the current clock value to avoid clock wrap.

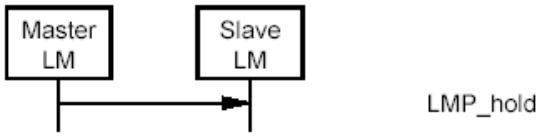
9.3.5.1.1 Master forces HOLD mode

The master may force HOLD mode if there has previously been a request for HOLD mode that has been accepted. The hold time included in the PDU when the master forces HOLD mode shall not be longer than any hold time the slave has previously accepted when there was a request for HOLD mode. See Sequence 47.

The master LM shall first pause traffic on the ACL-U logical link (see 8.5.3.1). It shall select the hold instant and queue the LMP_hold PDU to its link control for transmission. It shall then start a timer to wait until the hold instant occurs. When this timer expires, then the connection shall enter HOLD mode. If the BB acknowledgment for the LMP_hold PDU is not received, then the master may enter HOLD mode, but it shall not use its low accuracy clock during the HOLD mode.

When the slave LM receives an LMP_hold PDU, it compares the hold instant with the current CLK value. If it is in the future, then it starts a timer to expire at this instant and enters HOLD mode when it expires.

When the master LM exits from HOLD mode, it reenables transmission on the ACL-U logical link.



Sequence 47: Master forces slave into HOLD mode.

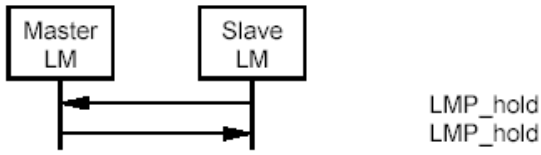
9.3.5.1.2 Slave forces HOLD mode

The slave may force HOLD mode if there has previously been a request for HOLD mode that has been accepted. The hold time included in the PDU when the slave forces HOLD mode shall not be longer than any hold time the master has previously accepted when there was a request for HOLD mode. See Sequence 48.

The slave LM shall first complete the transmission of the current packet on the ACL logical transport and then shall suspend transmission on the ACL-U logical link. It shall select the hold instant and queue the LMP_hold PDU to its link control for transmission. It shall then wait for an LMP_hold PDU from the master acting according to the procedure described in 9.3.5.1.1.

When the master LM receives an LMP_hold PDU, it shall pause traffic on the ACL-U logical link (see 8.5.3.1). It shall then inspect the hold instant. If this is less than $6*T_{poll}$ slots in the future, it shall modify the instant so that it is at least $6*T_{poll}$ slots in the future. It shall then send an LMP_hold PDU using the mechanism described in 9.3.5.1.1.

When the master and slave LMs exit from HOLD mode, they shall reenable transmission on the ACL-U logical link.



Sequence 48: Slave forces master into HOLD mode.

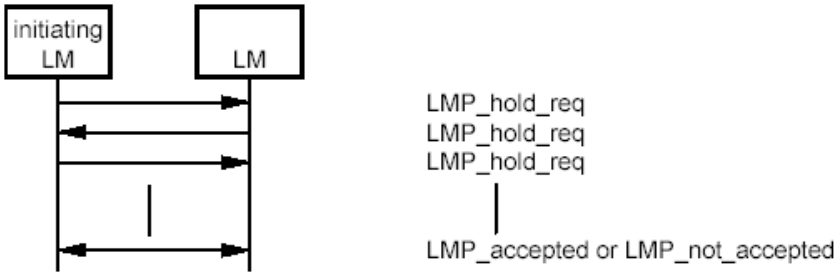
9.3.5.1.3 Master or slave requests HOLD mode

The master or the slave can request to enter HOLD mode. Upon receipt of the request, the same request with modified parameters can be returned, or the negotiation can be terminated. If an agreement is seen, an LMP_accepted PDU terminates the negotiation, and the ACL link is placed in HOLD mode. If no agreement is seen, an LMP_not_accepted PDU with the error code *unsupported parameter value* terminates the negotiation, and HOLD mode is not entered.

The initiating LM shall pause traffic on the ACL-U logical link (see 8.5.3.1). On receiving an LMP_hold_req PDU, the receiving LM shall complete the transmission of the current packet on the ACL logical transport and then shall suspend transmission on the ACL-U logical link.

The LM sending the LMP_hold_req PDU selects the hold instant that shall be at least $9*T_{poll}$ slots in the future. If this is a response to a previous LMP_hold_req PDU and the hold instant contained is at least $9*T_{poll}$ slots in the future, then this shall be used. LMP_hold_req PDU shall then be queued to its link control for transmission, and it shall start a timer to expire at the hold instant. When the timer expires, the connection shall enter HOLD mode unless an LMP_not_accepted or LMP_hold_req PDU is received before expiry. If the LM receiving LMP_hold_req PDU agrees to enter HOLD mode, it shall return an LMP_accepted PDU and shall start a timer to expire at the hold instant. When this timer expires, it enters HOLD mode.

When each LM exits from HOLD mode, it shall reenable transmission on the ACL-U logical link. See Sequence 49.



Sequence 49: Negotiation for HOLD mode.

9.3.5.2 PARK state

If a slave does not need to participate in the channel, but should still remain synchronized to the master, it may be placed in PARK state. See 8.8.9 for details.

To keep a parked slave connected, the master shall periodically unpark and repark the slave if the supervision timeout is not set to zero (see 8.3.1).

All PDUs sent from the master to parked slaves are carried on the control logical channel using the PSB logical transport (LMP link of PSB logical transport). These PDUs (i.e., LMP_set_broadcast_scan_window, LMP_modify_beacon, LMP_unpark_BD_addr_req and LMP_unpark_ PM_addr_req) are the only PDUs that shall be sent to a slave in PARK state and the only LMP PDUs that shall be broadcast. To increase reliability for broadcast, the packets are as short as possible. Therefore, the format for these LMP PDUs are somewhat different. The parameters are not always byte-aligned, and the length of the PDUs is variable.

The messages for controlling PARK state include parameters, defined in 8.8.9. When a slave is placed in PARK state, it is assigned a unique PM_ADDR, which can be used by the master to unpark that slave. The all-zero PM_ADDR has a special meaning; it is not a valid PM_ADDR. If a device is assigned this PM_ADDR, it shall be identified with its BD_ADDR when it is unparked by the master. See Table 60.

Table 60—PDUs used for PARK state

M/O	PDU	Contents
O(8)	LMP_park_req	Timing control flags D_B T_B N_B Δ_B PM_ADDR AR_ADDR N_{Bsleep} D_{Bsleep} D_{access} T_{access} $N_{acc\text{-}slots}$ N_{poll} M_{access} Access scheme

Table 60—PDUs used for PARK state *(continued)*

M/O	PDU	Contents
O(8)	LMP_set_broadcast_scan_window	Timing control flags D_B (optional) Broadcast scan window
O(8)	LMP_modify_beacon	Timing control flags D_B (optional) T_B N_B Δ_B D_{access} T_{access} $N_{acc\text{-}slots}$ N_{poll} M_{access} access scheme
O(8)	LMP_unpark_PM_ADDR_req	Timing control flags D_B (optional) LT_ADDR 1st unpark LT_ADDR 2nd unpark(optional) PM_ADDR 1st unpark PM_ADDR 2nd unpark(optional) LT_ADDR 3rd unpark(optional) LT_ADDR 4th unpark(optional) PM_ADDR 3rd unpark(optional) PM_ADDR 4th unpark(optional) LT_ADDR 5th unpark(optional) LT_ADDR 6th unpark(optional) PM_ADDR 5th unpark(optional) PM_ADDR 6th unpark(optional) LT_ADDR 7th unpark(optional) PM_ADDR 7th unpark(optional)
O(8)	LMP_unpark_BD_ADDR _req	Timing control flags D_B (optional) LT_ADDR LT_ADDR (optional) BD_ADDR BD_ADDR (optional)

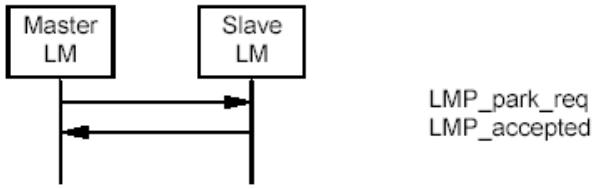
9.3.5.2.1 Master requests slave to enter PARK state

The master can request PARK state. The master LM shall pause traffic on the ACL-U logical link (see 8.5.3.1) and then send an LMP_park_req PDU. If the slave agrees to enter PARK state, it shall pause traffic on the ACL-U logical link (see 8.5.3.1) and then respond with an LMP_accepted PDU.

When the slave queues an LMP_accepted PDU, it shall start a timer for $6*T_{poll}$ slots. If the BB acknowledgment is received before this timer expires, the slave shall enter PARK state immediately; otherwise, it shall enter PARK state when the timer expires.

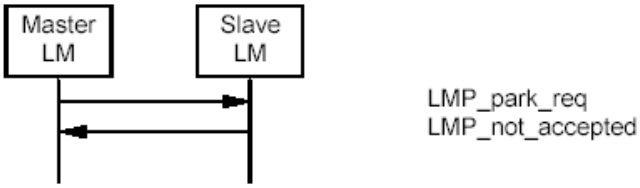
When the master receives an LMP_accepted PDU, it shall start a timer for $6*T_{poll}$ slots. When this timer expires, the slave is in PARK state, and the LT_ADDR may be reused.

If the master never receives an LMP_accepted PDU, then a link supervision timeout will occur. See Sequence 50.



Sequence 50: Slave accepts to enter PARK state.

If the slave rejects the attempt to enter PARK state, it shall respond with an LMP_not_accepted PDU, and the master shall reenable transmission on the ACL-U logical link. See Sequence 51.



Sequence 51: Slave rejects to enter into PARK state.

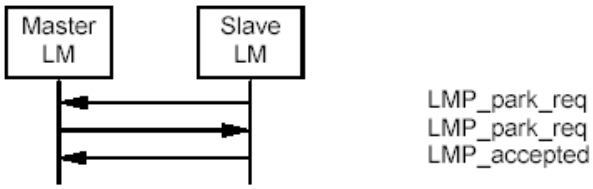
9.3.5.2.2 Slave requests to enter PARK state

The slave can request PARK state. The slave LM shall pause traffic on the ACL-U logical link (see 8.5.3.1) and then send an LMP_park_req PDU. When sent by the slave, the parameters PM_ADDR and AR_ADDR are not valid and the other parameters represent suggested values. If the master accepts the slave's request to enter PARK state, it shall pause traffic on the ACL-U logical link (see 8.5.3.1) and then send an LMP_park_req PDU, where the parameter values may be different from the values in the PDU sent from the slave. If the slave can accept these parameters, it shall respond with an LMP_accepted PDU.

When the slave queues an LMP_accepted PDU for transmission, it shall start a timer for $6*T_{poll}$ slots. If the BB acknowledgment is received before this timer expires, it shall enter PARK state immediately; otherwise, it shall enter PARK state when the timer expires.

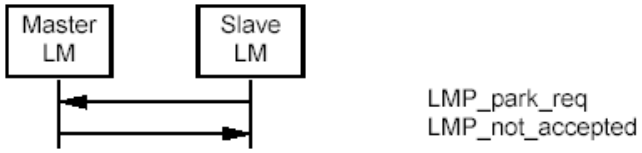
When the master receives an LMP_accepted PDU, it shall start a timer for $6*T_{poll}$ slots. When this timer expires, the slave is in PARK state, and the LT_ADDR may be reused.

If the master never receives the LMP_accepted PDU, then a link supervision timeout will occur. See Sequence 52.



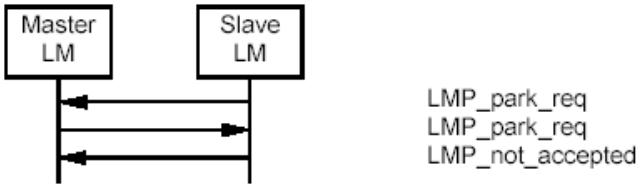
Sequence 52: Slave requests to enter PARK state and accepts master's beacon parameters.

If the master does not accept the slave's request to enter PARK state, it shall send an LMP_not_accepted PDU. The slave shall then reenable transmission on the ACL-U logical link. See Sequence 53.



Sequence 53: Master rejects slave's request to enter PARK state.

If the slave does not accept the parameters in the LMP_park_req PDU sent from the master, it shall respond with an LMP_not_accepted PDU, and both devices shall reenable transmission on the ACL-U logical link. See Sequence 54.



Sequence 54: Slave requests to enter PARK state, but rejects master's beacon parameters.

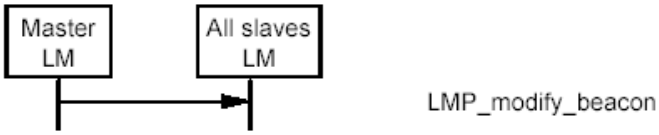
9.3.5.2.3 Master sets up broadcast scan window

If more broadcast capacity is needed than the beacon train, the master may indicate to the slaves that more broadcast information will follow the beacon train by sending an LMP_set_broadcast_scan_window PDU. This message shall be sent in a broadcast packet at the beacon slot(s). The scan window shall start in the beacon instant and shall be valid only for the current beacon. See Sequence 55.

Sequence 55: Master notifies all slaves of increase in broadcast capacity.

9.3.5.2.4 Master modifies beacon parameters

When the beacon parameters change, the master notifies the parked slaves of this by sending an LMP_modify_beacon PDU. This PDU shall be sent in a broadcast packet. See Sequence 56.



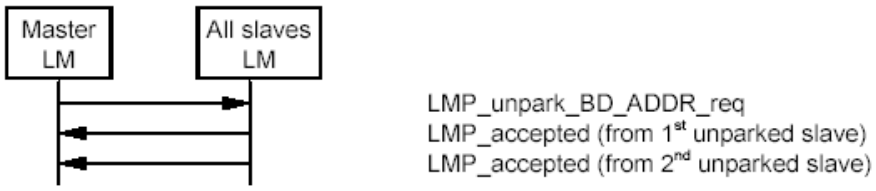
Sequence 56: Master modifies beacon parameters.

9.3.5.2.5 Unparking slaves

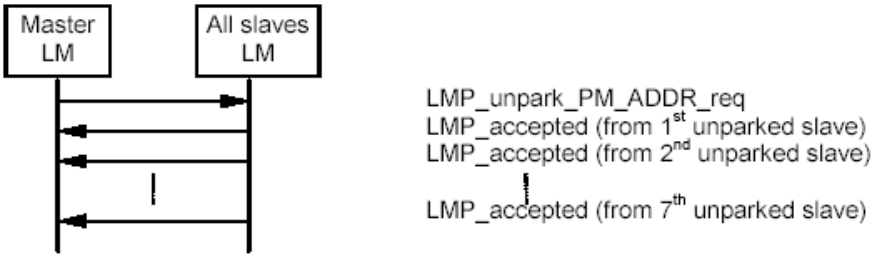
The master can unpark one or many slaves by sending a broadcast LMP message including the PM_ADDR or the BD_ADDR of the device(s) to be unparked. Broadcast LMP messages are carried on the control logical channel using the PSB logical transport. See 8.8.9.5 for further details. This message also includes the LT_ADDR that the master assigns to the slave(s). After sending this message, the master shall check the success of the unpark by polling each unparked slave by sending POLL packets, so that the slave is granted access to the channel. The unparked slave shall then send a response with an LMP_accepted PDU. If this message is not received from the slave within a *newconnectionTO* after the master sent the unpark message, the unpark failed, and the master shall consider the slave as still being in PARK state.

One PDU is used where the parked device is identified with the PM_ADDR, and another PDU is used where it is identified with the BD_ADDR. Both messages have variable length depending on the number of slaves the master unparks. For each slave the master wishes to unpark, an LT_ADDR, followed by the PM_ADDR or BD_ADDR of the device that is assigned this LT_ADDR, is included in the payload. If the slaves are identified with the PM_ADDR, a maximum of seven slaves can be unparked with the same message. If they are identified with the BD_ADDR, a maximum of two slaves can be unparked with the same message.

After a successful unparking, both master and slave reenable transmission on the ACL-U logical link. See Sequence 57 and Sequence 58.



Sequence 57: Master unparks slaves addressed with their BD_ADDR.



Sequence 58: Master unparks slaves addressed with their PM_ADDR.

9.3.5.3 SNIFF mode

To enter SNIFF mode, master and slave negotiate a sniff interval, T_{sniff}, and a sniff offset, D_{sniff}, that specifies the timing of the sniff slots. The offset determines the time of the first sniff slot; after that, the sniff slots follow periodically with the sniff interval T_{sniff}. To avoid clock wraparound during the initialization, one of two options is chosen for the calculation of the first sniff slot. A timing control flag in the message from the master indicates this. Only bit 1 of the timing control flag is valid.

When the ACL logical transport is in SNIFF mode, the master shall start a transmission only in the sniff slots. Two parameters control the listening activity in the slave: the sniff attempt and the sniff timeout. The sniff attempt parameter determines for how many slots the slave shall listen when the slave is not treating this as a scatternet link, beginning at the sniff slot, even if it does not receive a packet with its own LT_ADDR. The sniff timeout parameter determines for how many additional slots the slave shall listen

when the slave is not treating this as a scatternet link if it continues to receive only packets with its own LT_ADDR. It is not possible to modify the sniff parameters while the device is in SNIFF mode. See Table 61. (See 8.8.7 for more details of SNIFF mode behavior.)

Table 61—PDUs used for SNIFF mode

M/O	PDU	Contents
O(7)	LMP_sniff_req	Timing control flags D_{sniff} T_{sniff} Sniff attempt Sniff timeout
O(7)	LMP_unsniff_req	—

9.3.5.3.1 Master or slave requests SNIFF mode

Either the master or the slave may request entry to SNIFF mode.

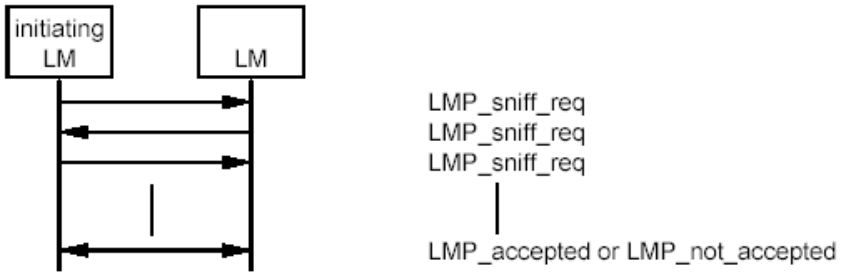
The process is initiated by sending an LMP_sniff_req PDU containing a set of parameters. The receiving LM shall then decide whether to reject the attempt by sending an LMP_not_accepted PDU, to suggest different parameters by replying with an LMP_sniff_req PDU, or to accept the request.

Before the first time that the master sends an LMP_sniff_req PDU, it shall enter SNIFF TRANSITION mode. If the master receives or sends an LMP_not_accepted PDU, it shall exit from SNIFF TRANSITION mode. If the master receives an LMP_sniff_req PDU, it shall enter SNIFF TRANSITION mode.

If the master decides to accept the request, it shall send an LMP_accepted PDU. When the master receives the BB acknowledgment for this PDU, it shall exit SNIFF TRANSITION mode and enter SNIFF mode.

If the master receives an LMP_accepted PDU, the master shall exit from SNIFF TRANSITION mode and enter SNIFF mode.

If the slave receives an LMP_sniff_req PDU, it must decide whether to accept the request. If the slave rejects SNIFF mode, then it replies with an LMP_not_accepted PDU. If the slave accepts SNIFF mode, but requires a different set of parameters, it shall respond with an LMP_sniff_req PDU containing the new parameters. If the slave decides that the parameters are acceptable, then it shall send an LMP_accepted PDU and enter SNIFF mode. If the slave receives an LMP_not_accepted PDU, it shall terminate the attempt to enter SNIFF mode. See Sequence 59.



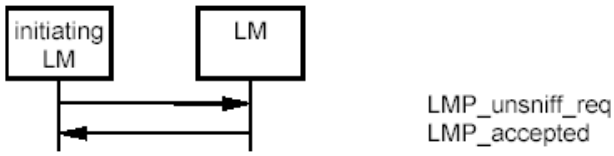
Sequence 59: Negotiation for SNIFF mode.

9.3.5.3.2 Moving a slave from SNIFF mode to active mode

SNIFF mode may be exited by either the master or the slave sending an LMP_unsniff_req PDU. The requested device must reply with an LMP_accepted PDU.

If the master requests an exit from SNIFF mode, it shall enter SNIFF TRANSITION mode and then send an LMP_unsniff_req PDU. When the slave receives the LMP_unsniff_req, it shall exit from SNIFF mode and reply with an LMP_accepted PDU. When the master receives the LMP_accepted PDU, it shall exit from SNIFF TRANSITION mode and enter active mode.

If the slave requests an exit from SNIFF mode, it shall send an LMP_unsniff_req PDU. When the master receives the LMP_unsniff_req PDU, it shall enter SNIFF TRANSITION mode and then send an LMP_accepted PDU. When the slave receives the LMP_accepted PDU, it shall exit from SNIFF mode and enter active mode. When the master receives the BB acknowledgment for the LMP_accepted PDU, it shall leave SNIFF TRANSITION mode and enter active mode. See Sequence 60.



Sequence 60: Slave moved from SNIFF mode to active mode.

9.3.6 Logical transports

When a connection is first established between two devices, the connection consists of the default ACL logical transport carrying the control logical link (for LMP messages) and the user logical link (for L2CAP data). One or more synchronous logical transports (SCO or eSCO) may then be added. A new logical transport shall not be created if it would cause all slots to be allocated to reserved slots on secondary LT_ADDRs.

9.3.6.1 SCO logical transport

The SCO logical transport reserves slots separated by the SCO interval T_{sco}. The first slot reserved for the SCO logical transport is defined by T_{sco} and the SCO offset D_{sco}. See 8.8.6.2 for details. A device shall initiate a request for **HV2** or **HV3** packet type only if the other device supports it (bit 12, bit 13) in its features mask. A device shall initiate CVSD, μ-law, or A-law coding or uncoded (transparent) data only if the other device supports the corresponding feature. To avoid problems with a wraparound of the clock during initialization of the SCO logical transport, the timing control flags parameter is used to indicate how the first SCO slot shall be calculated. Only bit 1 of the timing control flags parameter is valid. The SCO link is distinguished from all other SCO links by an SCO handle. The SCO handle zero shall not be used. See Table 62.

Table 62—PDUs used for managing the SCO links

M/O	PDU	Contents
O(11)	LMP_SCO_link_req	SCO handle Timing control flags D_{sco} T_{sco} SCO packet Air mode
O(11)	LMP_remove_SCO_link_req	SCO handle Error

9.3.6.1.1 Master initiates an SCO link

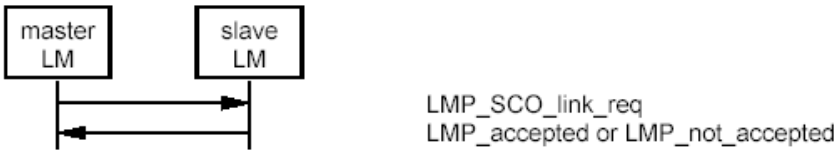
When establishing an SCO link, the master sends a request, a LMP_SCO_link_req PDU, with parameters that specify the timing, packet type, and coding that will be used on the SCO link. Each of the SCO packet types supports three different voice coding formats on the air-interface: μ-law log PCM, A-law log PCM, and CVSD. The air coding by log PCM or CVSD may be deactivated to achieve a transparent synchronous data link at 64 kb/s.

The slots used for the SCO links are determined by three parameters controlled by the master: T_{sco}, D_{sco}, and a flag indicating how the first SCO slot is calculated. After the first slot, the SCO slots follow periodically at an interval of T_{sco}.

If the slave does not accept the SCO link, but is willing to consider another possible set of SCO parameters, it can indicate what it does not accept in the error code field of LMP_not_accepted PDU. The master may then issue a new request with modified parameters.

The SCO handle in the message shall be different from existing SCO link(s).

If the SCO packet type is **HV1**, the LMP_accepted PDU shall be sent using the **DM1** packet. See Sequence 61.

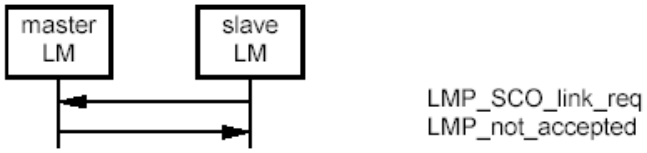


Sequence 61: Master requests an SCO link.

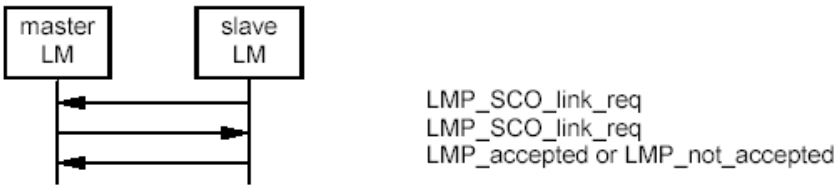
9.3.6.1.2 Slave initiates an SCO link

The slave may initiate the establishment of an SCO link. The slave sends an LMP_SCO_link_req PDU, but the parameters timing control flags and D_{sco} are invalid as well as the SCO handle, which shall be zero. If the master is not capable of establishing an SCO link, it replies with an LMP_not_accepted PDU. Otherwise, it sends back an LMP_SCO_link_req PDU. This message includes the assigned SCO handle, D_{sco}, and the timing control flags. The master should try to use the same parameters as in the slave request; if the master cannot meet that request, it is allowed to use other values. The slave shall then reply with LMP_accepted or LMP_not_accepted PDU.

If the SCO packet type is **HV1**, the LMP_accepted shall be sent using the **DM1** packet. See Sequence 62 and Sequence 63.



Sequence 62: Master rejects slave's request for an SCO link.



Sequence 63: Master accepts slave's request for an SCO link.

9.3.6.1.3 Master requests change of SCO parameters

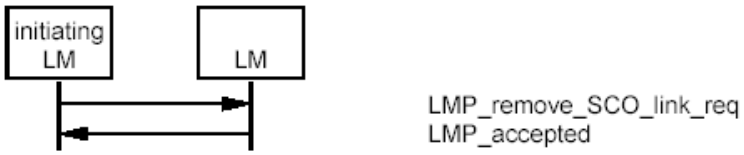
The master sends an LMP_SCO_link_req PDU where the SCO handle is the handle of the SCO link for which the master wishes to change parameters. If the slave accepts the new parameters, it replies with an LMP_accepted PDU, and the SCO link will change to the new parameters. If the slave does not accept the new parameters, it shall reply with an LMP_not_accepted PDU, and the SCO link is left unchanged. When the slave replies with an LMP_not_accepted PDU, it shall indicate in the error code parameter what it does not accept. The master may then try to change the SCO link again with modified parameters. The sequence is the same as in 9.3.6.1.1.

9.3.6.1.4 Slave requests change of SCO parameters

The slave sends an LMP_SCO_link_req PDU where the SCO handle is the handle of the SCO link to be changed. The parameters timing control flags and D_{sco} are not valid in this PDU. If the master does not accept the new parameters, it shall reply with an LMP_not_accepted PDU, and the SCO link is left unchanged. If the master accepts the new parameters, it shall reply with an LMP_SCO_link_req PDU containing the same parameters as in the slave request. When receiving this message, the slave replies with an LMP_not_accepted PDU if it does not accept the new parameters. The SCO link is then left unchanged. If the slave accepts the new parameters, it replies with an LMP_accepted PDU, and the SCO link will change to the new parameters. The sequence is the same as in 9.3.6.1.2.

9.3.6.1.5 Remove an SCO link

Master or slave can remove the SCO link by sending a request including the SCO handle of the SCO link to be removed and an error code indicating why the SCO link is removed. The receiving side shall respond with an LMP_accepted PDU. See Sequence 64.



Sequence 64: SCO link removed.

9.3.6.2 eSCO logical transport

After an ACL link has been established, one or more eSCO links can be set up to the remote device. The eSCO links are similar to SCO links using timing control flags, an interval T_{eSCO}, and an offset D_{eSCO}. Only bit 1 of the timing control flags parameter is valid. As opposed to SCO links, eSCO links have a configurable data rate that may be asymmetric and can be set up to provide limited retransmissions of lost or damaged packets inside a retransmission window of size W_{eSCO}. The D_{eSCO} shall be based on CLK. See Table 63.

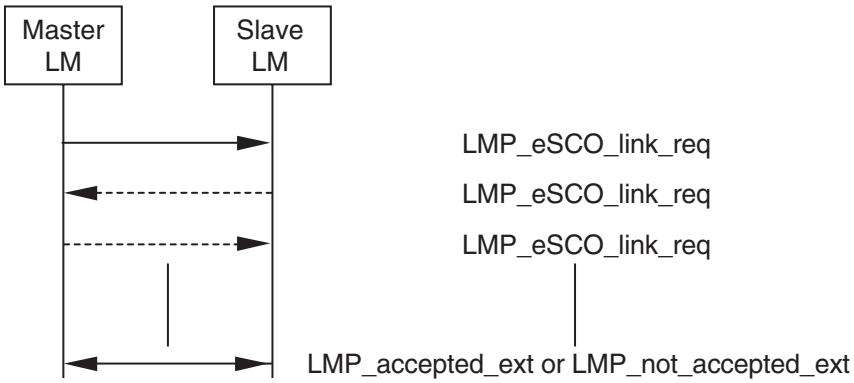
Table 63—PDUs used for managing the eSCO links

M/O	PDU	Contents
O(31)	LMP_eSCO_link_req	eSCO handle eSCO LT_ADDR Timing control flags D_{eSCO} T_{eSCO} W_{eSCO} eSCO packet type M→S eSCO packet type S→M Packet length M→S Packet length S→M Air mode Negotiation state
O(31)	LMP_remove_eSCO_link_req	eSCO handle Error

The parameters D_{eSCO}, T_{eSCO}, W_{eSCO}, eSCO packet type M→S, eSCO packet type S→M, packet length M→S, packet length S→M are henceforth referred to as the *negotiable parameters*.

9.3.6.2.1 Master initiates an eSCO link

When establishing an eSCO link, the master sends an LMP_eSCO_link_req PDU specifying all parameters. The slave may accept this with an LMP_accepted_ext PDU, reject it with an LMP_not_accepted_ext PDU, or respond with its own LMP_eSCO_link_req PDU specifying alternatives for some or all parameters. The slave shall not negotiate the eSCO handle or eSCO LT_ADDR parameters. The negotiation of parameters continues until the master or slave either accepts the latest parameters with an LMP_accepted_ext PDU or terminates the negotiation with an LMP_not_accepted_ext PDU. The negotiation shall use the procedures defined in 9.3.6.2.5. See Sequence 65.

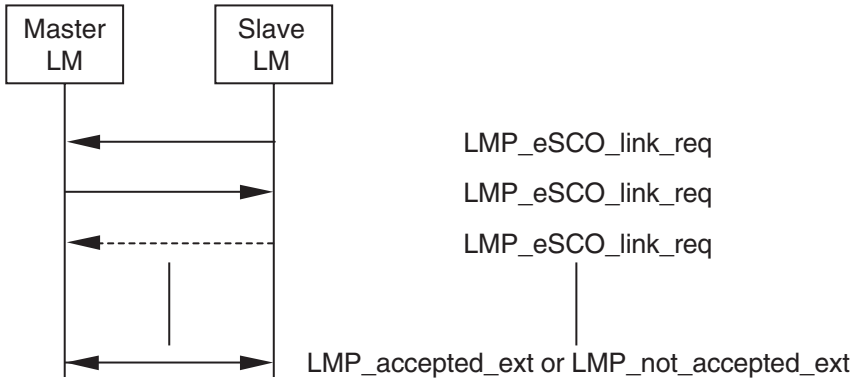


Sequence 65: Master requests an eSCO link.

9.3.6.2.2 Slave initiates an eSCO link

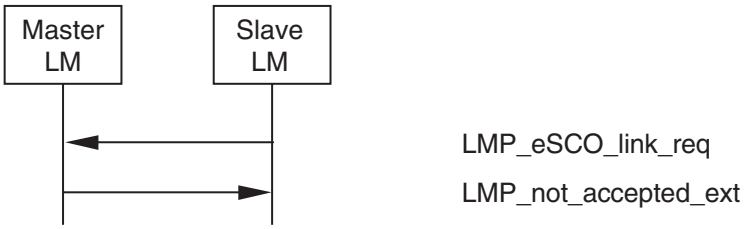
When attempting to establish an eSCO link, the slave shall send an LMP_eSCO_link_req PDU specifying all parameters, with the exception of eSCO LT_ADDR and eSCO handle, which are invalid. The latter shall be set to zero. The master may respond to this with an LMP_eSCO_link_req PDU, filling in these missing parameters and potentially changing the other requested parameters. The slave may accept this with an

LMP_accepted_ext PDU or respond with a further LMP_eSCO_link_req PDU specifying alternatives for some or all of the parameters. The negotiation of parameters continues until the master or slave either accepts the latest parameters with an LMP_accepted_ext PDU or terminates the negotiation with an LMP_not_accepted_ext PDU. See Sequence 66.



Sequence 66: Slave requests an eSCO link.

The master may reject the request immediately with an LMP_not_accepted_ext PDU. The negotiation shall use the procedures defined in 9.3.6.2.5. See Sequence 67.



Sequence 67: Master rejects slave's request for an eSCO link.

9.3.6.2.3 Master or slave requests change of eSCO parameters

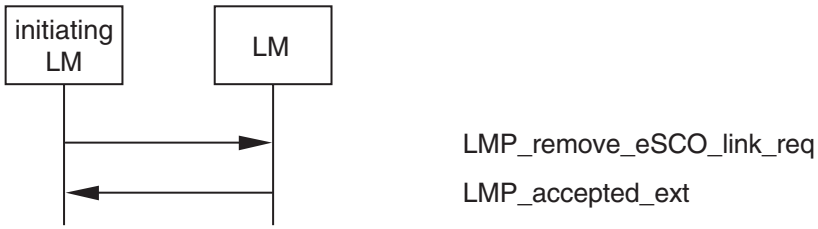
The master or slave may request a renegotiation of the eSCO parameters. The master or slave shall send an LMP_eSCO_link_req PDU with the eSCO handle of the eSCO link the device wishes to renegotiate. The remote device may accept the changed parameters immediately with LMP_accepted_ext PDU, or the negotiation may be continued with further LMP_eSCO_link_req PDUs until the master or slave accepts the latest parameters with an LMP_accepted_ext PDU or terminates the negotation with an LMP_not_accepted_ext PDU. In the case of termination with an LMP_not_accepted_ext PDU, the eSCO link continues on the previously negotiated parameters.

The sequence is the same as in 9.3.6.2.2.

During renegotiation, the eSCO LT_ADDR and eSCO handle shall not be renegotiated and shall be set to the originally negotiated values. The negotiation shall use the procedures defined in 9.3.6.2.5.

9.3.6.2.4 Remove an eSCO link

Either the master or slave may remove the eSCO link by sending a request including the eSCO handle of the eSCO link to be removed and a error code indicating why the eSCO link is removed. The receiving side shall respond with an LMP_accepted_ext PDU. See Sequence 68.



Sequence 68: eSCO link removed.

9.3.6.2.5 Rules for the LMP negotiation and renegotiation

Rule 1: The negotiation state shall be set to 0 by the initiating LM. After the initial LMP_eSCO_link_req PDU is sent, the negotiation state shall not be set to 0.

Rule 2: If the bandwidth (defined as 1600 times the packet length in bytes divided by T_{eSCO} in slots) for either RX or TX or the air mode cannot be accepted, the device shall send an LMP_not_accepted_ext PDU with the appropriate error code.

Rule 3: Bandwidth and air mode are not negotiable and shall not be changed for the duration of the negotiation. Once one side has rejected the negotiation (with an LMP_not_accepted_ext PDU), a new negotiation may be started with different bandwidth and air mode parameters.

Rule 4: If the parameters would cause a latency violation (T_{eSCO} + W_{eSCO} + reserved synchronous slots > allowed local latency), the device should propose new parameters that shall not cause a reserved slot violation or latency violation for the device that is sending the parameters. In this case, the negotiation state shall be set to 3. Otherwise, the device shall send an LMP_not_accepted_ext PDU.

Rule 5: Once a device has received an LMP_eSCO_link_req PDU with the negotiation state set to 3 (latency violation), the device shall not propose any combination of packet type, T_{eSCO}, and W_{eSCO} that will give an equal or larger latency than the combination that caused the latency violation for the other device.

Rule 6: If the parameters would cause both a reserved slot violation and a latency violation, the device shall set the negotiation state to 3 (latency violation).

Rule 7: If the parameters would cause a reserved slot violation, the device should propose new parameters that shall not cause a reserved slot violation. In this case, the negotiation state shall be set to 2. Otherwise, the device shall send an LMP_not_accepted_ext PDU.

Rule 8: If the requested parameters are not supported, the device should propose a setting that is supported and set the negotiation state to 4. If it is not possible to find such a parameter set, the device shall send an LMP_not_accepted_ext PDU.

Rule 9: When proposing new parameters for reasons other than a latency violation, reserved slot violation, or configuration not supported, the negotiation state shall be set to 1.

9.3.6.2.6 Negotiation state definitions

The following terms are defined with regard to the negotiation state:

reserved slot violation: The receiving LM cannot set up the requested eSCO logical transport because the eSCO reserved slots would overlap with other regularly scheduled slots (e.g., other synchronous reserved slots, sniff instants, or park beacons).

latency violation: The receiving LM cannot set up the requested eSCO logical transport because the latency (W_{eSCO} + T_{eSCO} + reserved synchronous slots) is greater than the maximum allowed latency.

configuration not supported: The combination of parameters requested is not inside the supported range for the device.

9.3.7 Test mode

This subclause describes the LMP procedures used to activate control and exit test mode. Throughout this subclause, the device that is placed in test mode is known as the *device under test (DUT)*.

9.3.7.1 Activation and deactivation of test mode

The activation may be carried out locally (via a hardware or software interface) or using the air interface.

— For activation over the air interface, entering the test mode shall be locally enabled for security and type approval reasons. The implementation of this local enabling is not subject to standardization.

  The tester sends an LMP command that shall force the DUT to enter test mode. The DUT shall terminate all normal operation before entering the test mode.

  If test mode is locally enabled, the DUT shall return an LMP_accepted PDU on reception of an activation command. An LMP_not_accepted PDU with the error code *PDU not allowed* shall be returned if the DUT is not locally enabled.

— If the activation is performed locally using a hardware or software interface, the DUT shall terminate all normal operation before entering the test mode.

  Until a connection to the tester exists, the device shall perform page scan and inquiry scan. Extended scan activity is recommended.

The DUT is always the slave.

See Sequence 69 and Sequence 70.

Sequence 69: Activation of test mode successful.

Sequence 70: Activation of test mode fails. Slave is not allowed to enter test mode.

The test mode can be deactivated in two ways. Sending an LMP_test_control PDU with the test scenario set to “exit test mode” exits the test mode, and the slave returns to normal operation still connected to the master. Sending an LMP_detach PDU to the DUT ends the test mode and the connection.

9.3.7.2 Control of test mode

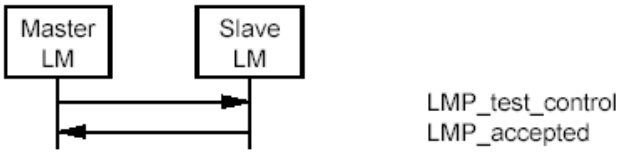
Control and configuration are performed using special LMP commands (see 9.3.7.3). These commands shall be rejected if the device is not in test mode. In this case, an LMP_not_accepted PDU shall be returned. The DUT shall return an LMP_accepted PDU on reception of a control command when in test mode.

An IEEE 802.15.1-2005 device in test mode shall ignore all LMP commands not related to control of the test mode. LMP commands dealing with power control and the request for LMP features (LMP_features_ req), and AFH (LMP_set_AFH, LMP_channel_classification_req, and LMP_channel_ classification) are allowed in test mode; the normal procedures are also used to test the adaptive power control.

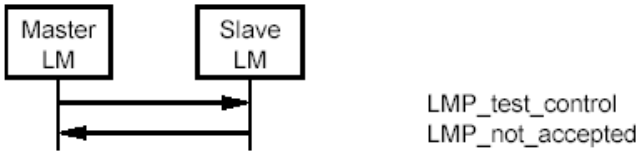
The DUT shall leave the test mode when an LMP_detach command is received or an LMP_test_control command is received with test scenario set to “exit test mode.”

When the DUT has entered test mode, the LMP_test_control PDU can be sent to the DUT to start a specific test. This PDU is acknowledged with an LMP_accepted PDU. If a device that is not in test mode receives an LMP_test_control PDU, it responds with an LMP_not_accepted PDU, where the error code shall be *PDU not allowed*.

See Sequence 71 and Sequence 72.



Sequence 71: Control of test mode successful.



Sequence 72: Control of test mode rejected since slave is not in test mode.

9.3.7.3 Summary of test mode PDUs

Table 64 lists all LMP messages used for test mode. To ensure that the contents of an LMP_test_control PDU are suitably whitened (important when sent in transmitter mode), each byte of all parameters listed in Table 65 are XORed with 0x55 before being sent.

The control PDU is used for both transmitter and loop back tests. The restrictions in Table 66 apply for the parameter settings.

Table 64—LMP messages used for test mode

LMP PDU	PDU number	Possible direction	Contents	Position in payload
LMP_test_activate	56	M → S		
LMP_test_control	57	M → S	Test scenario Hopping mode TX frequency RX frequency Power control mode Poll period Packet type Length of test data	2 3 4 5 6 7 8 9–10
LMP_detach	7	M → S		
LMP_accepted	3	M ← S		
LMP_not_accepted	4	M ← S		

Table 65—Parameters used in LMP_test_control PDU

Name	Length (bytes)	Type	Unit	Detailed
Test scenario	1	u_int8		0: Pause test mode 1: Transmitter test – 0 pattern 2: Transmitter test – 1 pattern 3: Transmitter test – 1010 pattern 4: Pseudorandom bit sequence 5: Closed loop back – ACL packets 6: Closed loop back – Synchronous packets 7: ACL packets without whitening 8: Synchronous packets without whitening 9: Transmitter test – 1111 0000 pattern 10–254: Reserved 255: Exit test mode The value is XORed with 0x55.
Hopping mode	1	u_int8		0: RX/TX on single frequency 1: Normal hopping 2: Reserved 3: Reserved 4: Reserved 5–255: Reserved The value is XORed with 0x55.
TX frequency (for DUT)	1	u_int8		f = [2402 + k] MHz The value is XORed with 0x55.
RX frequency (for DUT)	1	u_int8		f = [2402 + k] MHz The value is XORed with 0x55.
Power control mode	1	u_int8		0: Fixed TX output power 1: Adaptive power control The value is XORed with 0x55.
Poll period	1	u_int8	1.25 ms	The value is XORed with 0x55.

Table 65—Parameters used in LMP_test_control PDU *(continued)*

Name	Length (bytes)	Type	Unit	Detailed
Packet type	1	u_int8		Bits 3:0 Numbering as in packet header (see Clause 8) Bits 7:4 0: ACL/SCO 1: eSCO 2–15: Reserved The value is XORed with 0x55.
Length of test sequence (= length of user data in Clause 8)	2	u_int16	1 byte	Unsigned binary number The value is XORed with 0x55.

Table 66—Restrictions for parameters used in LMP_test_control PDU

Parameter	Restrictions transmitter test	Restrictions loopback test
TX frequency	$0 \le k \le 93$	$0 \le k \le 78$
RX frequency	Same as TX frequency	$0 \le k \le 78$
Poll period		Not applicable (set to 0)
Length of test sequence	Depends on packet type: **DH1**: ≤ 27 bytes **DH3**: ≤ 183 bytes **DH5**: ≤ 339 bytes **AUX1**: ≤ 29 bytes **HV3**: = 30 bytes **EV3**: ≤ 30 bytes **EV5**: ≤ 180 bytes	For ACL and SCO packets: not applicable (set to 0) For eSCO packets: **EV3**: ≤ 1–30 bytes **EV4**: ≤ 1–120 bytes **EV5**: ≤ 1–180 bytes

9.4 Summary

9.4.1 PDU summary

Table 67 shows the coding of the different LM PDUs.

Table 67—Coding of the different LM PDUs

LMP PDU	Length (bytes)	Op-code	Packet type	Possible direction	Contents	Position in payload
Escape 1	Variable	124	**DM1**	M ↔ S	Extended opcode	2
					Variable	3–?
Escape 2	Variable	125	**DM1**	M ↔ S	Extended opcode	2
					Variable	3–?

Table 67—Coding of the different LM PDUs *(continued)*

LMP PDU	Length (bytes)	Op-code	Packet type	Possible direction	Contents	Position in payload
Escape 3	Variable	126	**DM1**	M ↔ S	Extended opcode	2
					Variable	3–?
Escape 4	Variable	127	**DM1**	M ↔ S	Extended opcode	2
					Variable	3–?
LMP_accepted	2	3	**DM1**/ **DV**	M ↔ S	Opcode	2
LMP_accepted_ext	4	127/ 01	**DM1**	M ↔ S	Escape opcode	3
					Extended opcode	4
LMP_au_rand	17	11	**DM1**	M ↔ S	Random number	2–17
LMP_auto_rate	1	35	**DM1**/ **DV**	M ↔ S	—	
LMP_channel_ classification_req	7	127/ 16	**DM1**	M → S	AFH_reporting_mode	3
					AFH_min_interval	4–5
					AFH_max_interval	6–7
LMP_channel_ classification	12	127/ 17	**DM1**	M ← S	AFH_channel_ classification	3–12
LMP_clkoffset_req	1	5	**DM1**/ **DV**	M → S	—	
LMP_clkoffset_res	3	6	**DM1**/ **DV**	M ← S	Clock offset	2–3
LMP_comb_key	17	9	**DM1**	M ↔ S	Random number	2–17
LMP_decr_power_req	2	32	**DM1**/ **DV**	M ↔ S	For future use	2
LMP_detach	2	7	**DM1**/ **DV**	M ↔ S	Error code	2
LMP_encryption_key_ size_mask_req	1	58	**DM1**	M → S		
LMP_encryption_key_ size_mask_res	3	59	**DM1**	M ← S	Key size mask	2–3
LMP_encryption_key_ size_req	2	16	**DM1**/ **DV**	M ↔ S	Key size	2
LMP_encryption_mode_ req	2	15	**DM1**/ **DV**	M ↔ S	Encryption mode	2

Table 67—Coding of the different LM PDUs *(continued)*

LMP PDU	Length (bytes)	Op-code	Packet type	Possible direction	Contents	Position in payload
LMP_eSCO_link_req	16	127/ 12	**DM1**	M ↔ S	eSCO handle	3
					eSCO LT_ADDR	4
					Timing control flags	5
					D_{eSCO}	6
					T_{eSCO}	7
					W_{eSCO}	8
					SCO packet type M→S	9
					SCO packet type S→M	10
					Packet length M→S	11–12
					Packet length S→M	13–14
					Air mode	15
					Negotiation state	16
LMP_features_req	9	39	**DM1**/ **DV**	M ↔ S	Features	2–9
LMP_features_req_ext	12	127/ 03	**DM1**	M ↔ S	Features page	3
					Maximum supported page	4
					Extended features	5–12
LMP_features_res	9	40	**DM1**/ **DV**	M ↔ S	Features	2–9
LMP_features_res_ext	12	127/ 04	**DM1**	M ↔ S	Features page	3
					Maximum supported page	4
					Extended features	5–12
LMP_host_connection_req	1	51	**DM1**/ **DV**	M ↔ S	—	
LMP_hold	7	20	**DM1**/ **DV**	M ↔ S	Hold time	2–3
					Hold instant	4–7
LMP_hold_req	7	21	**DM1**/ **DV**	M ↔ S	Hold time	2–3
					Hold instant	4–7
LMP_incr_power_req	2	31	**DM1**/ **DV**	M ↔ S	For future use	2
LMP_in_rand	17	8	**DM1**	M ↔ S	Random number	2–17
LMP_max_power	1	33	**DM1**/ **DV**	M ↔ S	—	

Table 67—Coding of the different LM PDUs *(continued)*

LMP PDU	Length (bytes)	Op-code	Packet type	Possible direction	Contents	Position in payload
LMP_max_slot	2	45	**DM1**/ **DV**	M ↔ S	Maximum slots	2
LMP_max_slot_req	2	46	**DM1**/ **DV**	M ↔ S	Maximum slots	2
LMP_min_power	1	34	**DM1**/ **DV**	M ↔ S	—	
LMP_modify_beacon	11 or 13	28	**DM1**	M → S	Timing control flags	2
					D_B	3–4
					T_B	5–6
					N_B	7
					Δ_B	8
					D_{access}	9
					T_{access}	10
					$N_{acc\text{-}slots}$	11
					N_{poll}	12
					M_{access}	13:0–3
					Access scheme	13:4–7
LMP_name_req	2	1	**DM1**/ **DV**	M ↔ S	Name offset	2
LMP_name_res	17	2	**DM1**	M ↔ S	Name offset	2
					Name length	3
					Name fragment	4–17
LMP_not_accepted	3	4	**DM1**/ **DV**	M ↔ S	Opcode	2
					Error code	3
LMP_not_accepted_ext	5	127/ 02	**DM1**	M ↔ S	Escape opcode	3
					Extended opcode	4
					Error code	5
LMP_page_mode_req	3	53	**DM1**/ **DV**	M ↔ S	Paging scheme	2
					Paging scheme settings	3
LMP_page_scan_mode_ req	3	54	**DM1**/ **DV**	M ↔ S	Paging scheme	2
					Paging scheme settings	3

Table 67—Coding of the different LM PDUs *(continued)*

LMP PDU	Length (bytes)	Op-code	Packet type	Possible direction	Contents	Position in payload
LMP_park_req	17	25	**DM1**	M ↔ S	Timing control flags	2
					D_B	3–4
					T_B	5–6
					N_B	7
					Δ_B	8
					PM_ADDR	9
					AR_ADDR	10
					N_{Bsleep}	11
					D_{Bsleep}	12
					D_{access}	13
					T_{access}	14
					$N_{acc\text{-}slots}$	15
					N_{poll}	16
					M_{access}	17:0–3
					Access scheme	17:4–7
LMP_preferred_rate	2	36	**DM1**/ **DV**	M ↔ S	Data rate	2
LMP_quality_of_service	4	41	**DM1**/ **DV**	M → S	Poll interval	2–3
					N_{BC}	4
LMP_quality_of_service_req	4	42	**DM1**/ **DV**	M ↔ S	Poll interval	2–3
					N_{BC}	4
LMP_remove_eSCO_link_ req (see Note 4)	4	127/ 13	**DM1**	M ↔ S	eSCO handle	3
					Error code	4
LMP_remove_SCO_link_ req	3	44	**DM1**/ **DV**	M ↔ S	SCO handle	2
					Error code	3
LMP_SCO_link_req	7	43	**DM1**/ **DV**	M ↔ S	SCO handle	2
					Timing control flags	3
					D_{sco}	4
					T_{sco}	5
					SCO packet	6
					Air mode	7

Table 67—Coding of the different LM PDUs *(continued)*

LMP PDU	Length (bytes)	Op-code	Packet type	Possible direction	Contents	Position in payload
LMP_set_AFH	16	60	**DM1**	M → S	AFH instant	2–5
					AFH mode	6
					AFH channel map	7–16
LMP_set_broadcast_ scan_window	4 or 6	27	**DM1**	M → S	Timing control flags	2
					D_B	3–4
					Broadcast scan window	5–6
LMP_setup_complete	1	49	**DM1**	M ↔ S	—	
LMP_slot_offset	9	52	**DM1**/ **DV**	M ↔ S	Slot offset	2–3
					BD_ADDR	4–9
LMP_sniff_req	10	23	**DM1**	M ↔ S	Timing control flags	2
					D_{sniff}	3–4
					T_{sniff}	5–6
					Sniff attempt	7–8
					Sniff timeout	9–10
LMP_sres	5	12	**DM1**/ **DV**	M ↔ S	Authentication response	2–5
LMP_start_encryption_req	17	17	**DM1**	M → S	Random number	2–17
LMP_stop_encryption_req	1	18	**DM1**/ **DV**	M → S	—	
LMP_supervision_timeout	3	55	**DM1**/ **DV**	M →s	Supervision timeout	2–3
LMP_switch_req	5	19	**DM1**/ **DV**	M ↔ S	Switch instant	2–5
LMP_temp_rand	17	13	**DM1**	M → S	Random number	2–17
LMP_temp_key	17	14	**DM1**	M → S	Key	2–17
LMP_test_activate	1	56	**DM1**/ **DV**	M → S	—	

Table 67—Coding of the different LM PDUs *(continued)*

LMP PDU	Length (bytes)	Op-code	Packet type	Possible direction	Contents	Position in payload
LMP_test_control	10	57	**DM1**	M → S	Test scenario	2
					Hopping mode	3
					TX frequency	4
					RX frequency	5
					Power control mode	6
					Poll period	7
					Packet type	8
					Length of test data	9–10
LMP_timing_accuracy_req	1	47	**DM1**/ **DV**	M ↔ S	—	
LMP_timing_accuracy_res	3	48	**DM1**/ **DV**	M ↔ S	Drift	2
					Jitter	3
LMP_unit_key	17	10	**DM1**	M ↔ S	Key	2–17
LMP_unpark_BD_ADDR_ req	variable	29	**DM1**	M → S	Timing control flags	2
					D_B	3–4
					LT_ADDR 1st unpark	5:0–2
					LT_ADDR 2nd unpark	5:4–6
					BD_ADDR 1st unpark	6–11
					BD_ADDR 2nd unpark	12–17
LMP_unpark_PM_ADDR_ req	variable	30	**DM1**	M → S	Timing control flags	2
					D_B	3–4
					LT_ADDR 1st unpark	5:0–3
					LT_ADDR 2nd unpark	5:4–7
					PM_ADDR 1st unpark	6
					PM_ADDR 2nd unpark	7
					LT_ADDR 3rd unpark	8:0–3
					LT_ADDR 4th unpark	8:4–7
					PM_ADDR 3rd unpark	9
					PM_ADDR 4th unpark	10
					LT_ADDR 5th unpark	11:0–3
					LT_ADDR 6th unpark	11:4–7
					PM_ADDR 5th unpark	12

Table 67—Coding of the different LM PDUs *(continued)*

LMP PDU	Length (bytes)	Op-code	Packet type	Possible direction	Contents	Position in payload
LMP_unpark_PM_ADDR_ req *(continued)*					PM_ADDR 6th unpark	13
					LT_ADDR 7th unpark	14:0–3
					PM_ADDR 7th unpark	15
LMP_unsniff_req	1	24	**DM1**/ **DV**	M ↔ S	—	
LMP_use_semi_ permanent_key	1	50	**DM1**/ **DV**	M → S	—	
LMP_version_req	6	37	**DM1**/ **DV**	M ↔ S	VersNr	2
					CompId	3–4
					SubVersNr	5–6
LMP_version_res	6	38	**DM1**/ **DV**	M ↔ S	VersNr	2
					CompId	3–4
					SubVersNr	5–6

NOTES

1—For LMP_set_broadcast_scan_window, LMP_modify_beacon, LMP_unpark_BD_ADDR_req, and LMP_ unpark_PM_ADDR_req PDUs, the parameter D_B is optional. This parameter is present only if bit 0 of timing control flags is 1. If the parameter is not included, the position in payload for all parameters following D_B are decreased by 2.

2—For the LMP_unpark_BD_ADDR PDU, the LT_ADDR and the BD_ADDR of the 2nd unparked slave are optional. If only one slave is unparked, LT_ADDR 2nd unpark shall be zero, and BD_ADDR 2nd unpark is left out.

3—For the LMP_unpark_PM_ADDR PDU, the LT_ADDR and the PM_ADDR of the 2nd – 7th unparked slaves are optional. If N slaves are unparked, the fields up to and including the N^{th} unparked slave are present. If N is odd, the LT_ADDR $(N + 1)^{th}$ unpark shall be zero. The length of the message is $x + 3N/2$ if N is even and $x + 3(N + 1)/2 – 1$ if N is odd, where $x = 2$ or 4 depending on if the D_B is included or not (see Note 1).

4—Parameters coincide with their namesakes in LMP_<remove>_SCO_link_req PDUs apart from the following:

a) eSCO_LT_ADDR: The eSCO connection will be active on an additional LT_ADDR that needs to be defined. The master is allowed to reassign an active eSCO link to a different LT_ADDR.
b) D_{eSCO}, T_{eSCO}: As per LMP_SCO_link_req, but with a greater flexibility in values (e.g., no longer fixed with respect to **HV1**, **HV2**, and **HV3** packet choice).
c) W_{eSCO}: The eSCO retransmission window size (in slots).
d) Packet type and packet length may be set differently in master-to-slave or slave-to-master directions for asynchronous eSCO links.
e) Packet length (in bytes): eSCO packet types no longer have fixed length
f) Negotiation state: This is used to better enable the negotiation of the negotiable parameters: D_{eSCO}, T_{eSCO}, W_{eSCO}, eSCO packet type M→S, eSCO packet type S→M, packet length M→S, packet length S→M. When responding to an eSCO link request with a new suggestion for these parameters, this flag may be set to 1 to indicate that the last received negotiable parameters are possible, but the new parameters specified in the response eSCO link request would be preferable, to 2 to indicate that the last received negotiable parameters are not possible as they cause a reserved slot violation, or to 3 to indicate that the last received negotiable parameters would cause a latency violation. The flag shall be set to zero in the initiating LMP_eSCO_ link_req PDU.

9.4.2 Parameter definitions

The parameter definitions are listed in Table 68.

Table 68—Parameters in LM PDUs

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
Access scheme	1	u_int4		0: Polling technique 1–15: Reserved	
AFH_channel_ classification	10	multiple bytes	—	This parameter contains 40 two-bit fields. The n^{th} (numbering from 0) such field defines the classification of channels $2n$ and $2n + 1$, other than the 39^{th} field, which just contains the classification of channel 78. Each field interpreted as an integer whose values indicate: 0 = unknown 1 = good 2 = reserved 3 = bad	
AFH_channel_map	10	multiple bytes	—	If AFH_mode is AFH_enabled, this parameter contains 79 one-bit fields; otherwise, the contents are reserved. The n^{th} (numbering 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). The 1-bit field is interpreted as follows: 0: channel n is unused 1: channel n is used	
AFH_instant	4	u_int32	slots	Bits 27:1 of the CLK value at the time of switching hop sequences. Must be even.	
AFH_max_interval	2	u_int16	slots	Range is 0x0640 to 0xBB80 slots (1 to 30 s)	
AFH_min_interval	2	u_int16	slots	Range is 0x0640 to 0xBB80 slots (1 to 30 s)	
AFH_mode	1	u_int8	—	0: AFH disabled 1: AFH enabled 2–255: Reserved	
AFH_reporting_ mode	1	u_int8	—	0: AFH reporting disabled 1: AFH reporting enabled 2–255: Reserved	

Table 68—Parameters in LM PDUs *(continued)*

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
Air mode	1	u_int8		0: μ-law log 1: A-law log 2: CVSD 3: Transparent data 4–255: Reserved	See Table 69
AR_ADDR	1	u_int8			
Authentication response	4	multiple bytes			
BD_ADDR	6	multiple bytes		BD_ADDR of the sending device	
Broadcast scan window	2	u_int16	slots		
Clock offset	2	u_int16	1.25 ms	($CLKN_{16-2}$ slave – $CLKN_{16-2}$ master) mod 2^{15} MSB of second byte not used.	
CompId	2	u_int16		See Bluetooth Assigned Numbers [B1]	
D_{access}	1	u_int8	slots		
Data rate	1	u_int8		Bit 0 = 0: Use FEC Bit 0 = 1: Do not use FEC Bit 1–2 = 0: No packet-size preference available Bit 1–2 = 1: Use 1-slot packets Bit 1–2 = 2: Use 3-slot packets Bit 1–2 = 3: Use 5-slot packets Bit 3–7: Reserved	
D_B	2	u_int16	slots		
Δ_B	1	u_int8	slots		
D_{Bsleep}	1	u_int8			
D_{eSCO}	1	u_int8	slots	Valid range is 0–254 slots	See Table 69.
Drift	1	u_int8	ppm		
D_{sco}	1	u_int8	slots	Only even values are valid[a]	0 to (T_{sco} – 2)
D_{sniff}	2	u_int16	slots	Only even values are valid[a]	0 to (T_{sniff} – 2)
Encryption mode	1	u_int8		0: No encryption 1: Encryption 2: Encryption 3–255: Reserved	
Error code	1	u_int8		See 10.3	
Escape opcode	1	u_int8		Identifies which escape opcode is being acknowledged: range 124–127	
eSCO handle	1	u_int8			

Table 68—Parameters in LM PDUs *(continued)*

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
eSCO LT_ADDR	1	u_int8		LT_ADDR for the eSCO logical transport. The range is extended to 8 bits compared with the normal LT_ADDR field: range 0–7.	0–7
eSCO packet type	1	u_int8		0x00: NULL/POLL 0x07: **EV3** 0x0C: **EV4** 0x0D: **EV5** Other values are reserved	If the value is 0x00, the POLL packet shall be used by the master, and the NULL packet shall be used by the slave. See Table 69.
Extended features	8	multiple bytes		One page of extended features	
Extended opcode	1	u_int8		Which extended opcode is being acknowledged	
Features	8	multiple bytes		See Table 35.	
Features page	1	u_int8		Identifies which page of extended features is being requested. 0 means standard features 1–255 other feature pages	
Hold instant	4	u_int32	slots	Bits 27:1 of the CLK value	
Hold time	2	u_int16	slots	Only even values are valid[1]	0x0014–0x8000; shall not exceed (*supervisionTO* * 0.999)
Jitter	1	u_int8	µs		
Key	16	multiple bytes			
Key size	1	u_int8	byte		
Key size mask	2	u_int16		Bit mask of supported broadcast encryption key sizes: LSB is support for length 1, and so on. The bit shall be one if the key size is supported.	
LT_ADDR	1	u_int4			
M_{access}	1	u_int4		Number of access windows	
Max slots	1	u_int8	slots		
Max supported page	1	u_int8		Highest page of extended features that contains a nonzero bit for the originating device. Range 0–255	

Table 68—Parameters in LM PDUs *(continued)*

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
$N_{acc\text{-}slots}$	1	u_int8	slots		
Name fragment	14	multiple bytes		UTF-8 characters.	
Name length	1	u_int8	bytes		
Name offset	1	u_int8	bytes		
N_B	1	u_int8			
N_{BC}	1	u_int8			
N_{Bsleep}	1	u_int8			
Negotiation state	1	u_int8		0: Initiate negotiation 1: The latest received set of negotiable parameters were possible, but these parameters are preferred. 2: The latest received set of negotiable parameters would cause a reserved slot violation. 3: The latest received set of negotiable parameters would cause a latency violation. 4: The latest received set of egotiable parameters are not supported. Other values are reserved.	
N_{poll}	1	u_int8			
Opcode	1	u_int8			
Packet length	2	u_int16	bytes	Length of the eSCO payload 0 for POLL/NULL 1–30 for **EV3** 1–120 for **EV4** 1–180 for **EV5** Other values are invalid	See Table 69.
Paging scheme	1	u_int8		0: Mandatory scheme 1-255: Reserved	
Paging scheme settings	1	u_int8		For mandatory scheme: 0: R0 1: R1 2: R2 3–255: Reserved	
PM_ADDR	1	u_int8			
Poll interval	2	u_int16	slots	Only even values are valid[a]	0x0006–0x1000
Random number	16	multiple bytes			

Table 68—Parameters in LM PDUs *(continued)*

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
Reserved(*n*)	*n*	u_int8		Reserved for future use – must be 0 when transmitted, ignore value when received	
SCO handle	1	u_int8			
SCO packet	1	u_int8		0: **HV1** 1: **HV2** 2: **HV3** 3–255: Reserved	
Slot offset	2	u_int16	μs	0 ≤ slot offset < 1250	
Sniff attempt	2	u_int16	received slots	Number of receive slots	1 to $T_{sniff}/2$
Sniff timeout	2	u_int16	received slots	Number of receive slots	0–0x0028
SubVersNr	2	u_int16		Defined by each company	
Supervision timeout	2	u_int16	slots	0 means an infinite timeout	0 and 0x0190–0xFFFF
Switch instant	4	u_int32	slots	Bits 27:1 of the CLK value	
T_{access}	1	u_int8	slots		
T_B	2	u_int16	slots		
T_{eSCO}	1	u_int8	slots	Valid range is 4–254 slots	See Table 69
Timing control flags	1	u_int8		Bit 0 = 0: No timing change Bit 0 = 1: Timing change Bit 1 = 0: Use initialization 1 Bit 1 = 1: Use initialization 2 Bit 2 = 0: Access window Bit 2 = 1: No access window Bit 3–7: Reserved	
T_{sco}	1	u_int8	slots	Only even values are valid[a]	2–6
T_{sniff}	2	u_int16	slots	Only even values are valid[a]	0x0006–0x0540; shall not exceed (*supervisionTO* * 0.999)
VersNr	1	u_int8		See Bluetooth Assigned Numbers [B1]	
W_{eSCO}	1	u_int8	slots	Number of slots in the retransmission window. Valid range is 0–254 slots.	See Table 69.

[a]If a device receives an LMP PDU with an odd value in this parameter field, the PDU should be rejected with an error code of *invalid LMP parameters*.

9.4.3 Default values

Devices shall use the values in Table 69 before anything else has been negotiated.

Table 69—Mandatory parameter ranges for eSCO packet types

Parameter	EV3	EV4	EV5
D_{eSCO}	0–4 (even)	0–14 (even)	0–14 (even)
T_{eSCO}	6	16 (even)	16 (even)
W_{eSCO}	0–4 (even)	0–6 (even)	0–6 (even)
eSCO packet type M→S	**EV3**	**EV3**, **EV4**	**EV3**, **EV5**
eSCO packet type S→M	**EV3**	**EV3**, **EV4**	**EV3**, **EV5**
Packet length M→S	30	1–120	1–180
Packet length S→M	30	1–120	1–180
Air mode	At least one of A-law, μ-law, CVSD, transparent	Transparent	Transparent