IEEE Standards

802.15.2™

IEEE Recommended Practice for Information technology— 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 Bands

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE

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

28 August 2003

Print: SH95135
PDF: SS95135

IEEE Recommended Practice for Information technology— 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 Bands

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Reaffirmed 13 May 2009
Approved 12 June 2003

IEEE-SA Standards Board

Approved 15 October 2003

American National Standards Institute

Abstract: This recommended practice addresses the issue of coexistence of wireless local area networks and wireless personal area networks. These wireless networks often operate in the same unlicensed band. This recommended practice describes coexistence mechanisms that can be used to facilitate coexistence of wireless local area networks (i.e., IEEE Std 802.11b™-1999)and wireless personal area networks (i.e., IEEE Std 802.15.1™-2002).
Keywords: coexistence, collaborative, collocated, interference, mechanisms, non-collaborative, WLAN, WPAN

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

Print: ISBN 0-7381-3702-2 SH95135
PDF: ISBN 0-7381-3703-0 SS95135

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.

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
P.O. Box 1331
Piscataway, NJ 08855-1331
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. A patent holder 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 all applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreements offered by patent holders. Further information may be obtained from the IEEE Standards Department.

Introduction

This introduction is not part of IEEE Std 802.15.2-2003, IEEE Recommended Practice for Information Technology—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 Bands

Interpretations and errata

Interpretations and errata associated with this standard may be found at one of the following Internet locations:

— http://standards.ieee.org/reading/ieee/interp/

— http://standards.ieee.org/reading/ieee/updates/errata

Participants

At the time this recommended practice was completed, two (2) working groups participated that had the following membership:

Working group 802.15
Robert F. Heile, *Chair*
James D. Allen, *Vice-Chair*
Ian C. Gifford, *Co-Vice-Chair*
Patrick Kinney, *Secretary*

Stephen J. Shellhammer, *Chair, 802.15.2*
Nada Golmie, *Vice-Chair, 802.15.2 and Chair, 802.15.2 MAC*
David E. Cypher, *Editor-in-Chief, 802.15.2*
Arun Arunachalam, *Secretary, 802.15.2*
Jim Lansford, *Chair, 802.15.2 PHY*

Roberto Aiello
Masaaki Akahane
Richard Alfvin
Arun Arunachalam
Naiel Askar
Venkat Bahl
Daniel Bailey
Jay Bain
John Barr
Anuj Batra
Timothy J. Blaney
Stan Bottoms
Monique Bourgeois
Chuck Brabenac
Ed Callaway
Soo-Young Chang
Hung Kun Chen
Aik Chindapol
Michael Derby
Mary DuVal
Michael Dydyk
Jason Ellis
Jeff Foerster
Pierre Gandolfo
James Gilb
Paul Gorday
Jose Gutierrez
Yasuo Harada
Allen Heberling
Barry Herold
Bob Huang
Laura L. Huckabee
Eran Igler
Katsumi Ishii
Phil Jamieson
Park Jong-Hun
Jeyhan Karaoguz
Joy H. Kelly
Stuart J. Kerry
Yongsuk Kim
Gunter Kleindl
Bruce P. Kraemer
David G. Leeper
Liang Li
Jie Liang
Shawn T. Liu
Yeong-Chang Maa
Ralph Mason
Michael D. McInnis
Jim Meyer
Leonard Miller
Akira Miura
Tony Morelli
Said Moridi
Marco Naeve
Chiu Y. Ngo
Erwin R. Noble
Knut Odman
Jack Pardee
Marcus Pendergrass
Robert D. Poor
Gregg Rasor
Ivan Reede
Jim Richards
William Roberts
Richard Roberts
Chris Rogers
Philippe Rouzet
Chandos Rypinski
John Santhoff
Mark Schrader
Tom Schuster
Erik Schylander
Michael Seals
Nick Shepherd
Gadi Shor
Bill Shvodian
Thomas Siep
Kazimierz Siwiak
Carl Stevenson
Rene Struik
Shigeru Sugaya
Kazuhisa Takamura
Katsumi Takaoka
Teik-Kheong Tan
Larry Taylor
Wim van Houtum
Hans van Leeuwen
Ritesh Vishwakarma
Thierry Walrant
Fujio Watanabe
Matthew Welborn
Richard Wilson
Stephen Wood
Edward G. Woodrow
Hirohisa Yamaguchi
Song-Lin Young

Working group 802.11
Stuart J. Kerry, *Chair*
Al Petrick, *Vice-Chair*
Harry Worstell, *Vice-Chair*
Tim Godfrey, *Secretary*
Brian Mathews, *Publicity Standing Committee*
Teik-Kheong Tan, *Wireless Next-Generation Standing Committee*

John Fakatselis, *Chair, Task Group e*
Duncan Kitchin, *Vice-Chair, Task Group e*
David Bagby, *Chair, Task Group f*
Matthew B. Shoemake, *Chair, Task Group g*
Mika Kasslin, *Chair, Task Group h*
David Halasz, *Chair, Task Group*

Bernard Aboba
L. Enrique Aguado
Masaaki Akahane
Areg Alimian
Richard Allen
Baruch Altman
Keith Amann
Merwyn Andrade
Carl F. Andren
David C. Andrus
Butch Anton
Mitch Aramaki
Takashi Aramaki
Larry Arnett
Geert A. Awater
David Bagby
Jay Bain
Bala Balachander
Simon Barber
Steve Bard
Michael Barkway
Gil Bar-Noy
Kevin M. Barry
Anuj Batra
Bob Beach
Randolph Beltz
Mathilde Benveniste
Stuart Biddulph
Simon Black
Simon Blake-Wilson
Timothy Blaney
Jan Boer
Jim Brennan
Ronald Brockmann
Robert Brummer
Richard Bulman, Jr.
Kevin Burak
Alistair G. Buttar
Dominick Cafarelli
Colum Caldwell
Nancy Cam-Winget
Bill Carney
Michael Carrafiello
Pat Carson
Joan Ceuterick
Hung-Kun Chen
James C. Chen
Kwang-Cheng Chen
Yi-Ming Chen
Brian Cheng
Greg Chesson
Harshal S. Chhaya
Alan Chickinsky
Aik Chindapol
Leigh M. Chinitz
Bong-Rak Choi
Sunghyun Choi
Patrick Chokron
Frank Ciotti
Ken Clements
John T. Coffey
Terry Cole
Anthony Collins
Craig Conkling
Dennis Connors
Todor Cooklev
Thomas P. Costas
Wm. Caldwell Crosswy
Russell J. Cyr
Peter Dahl
Barry Davis
Rolf De Vegt
Peter de Wit
Michael Derby
Georg Dickmann
Wim Diepstraten
Haoran Duan
Jeffrey Dunnihoo
Roger Durand
Eryk Dutkiewicz
Mary DuVal
Donald E. Eastlake III
Dennis Eaton
Peter Ecclesine
Jon Edney
Darwin Engwer
Javier Espinoza
Christoph Euscher
John Fakatselis
Lars Falk
Augustin J. Farrugia
Weishi Feng
Niels T. Ferguson
Matthew James Fischer
Michael Fischer
Jason Flaks
Aharon Friedman
Kenji Fujisawa
Shinya Fukuoka
Marcus Gahler
Zvi Ganz
James Gardner
Atul Garg
Vafa Ghazi
Amar Ghori
James Gilb
Tim Godfrey
Wataru Gohda
Peter Goidas
Andrew J. Gowans
Rik Graulus
Evan Green
Larry Green
Patrick Green
Kerry Greer
Daqing Gu
Rajugopal Gubbi
Srikanth Gummadi
Fred Haisch
David Halasz
Steve D. Halford
Neil Hamady
Mark Hamilton
Christopher J. Hansen
Yasuo Harada
Amer A. Hassan
Kevin Hayes
Victor Hayes
Chris Heegard
Robert Heile
Juha Heiskala
Jerry Heller
Bent Hessen-Schmidt
Garth Hillman
Christopher Hinsz
Jun Hirano
Jin-Meng Ho
Maarten Hoeben

Michael Hoghooghi
Russell Housley
Frank P. Howley, Jr.
Dave Hudak
John Hughes
David Hunter
David Hytha
Hiroshi Ide
Masataka Iizuka
Yasuhiko Inoue
Katsumi Ishii
Marc Jalfon
Hemaprabhu Jayanna
Jung Je Son
Ho-In Jeon
Peter Johansson
Sherry Johnson
V. K. Jones
Bobby Jose
Mark F. Kahn
Srinivas Kandala
Jeyhan Karaoguz
Kevin Karcz
Mika Kasslin
Patrick Kelly
Richard Kennedy
Stuart J. Kerry
Jamshid Khun-Jush
Ryoji Kido
Je Woo Kim
Joonsuk Kim
Ziv Kimhi
Ken Kimura
Duncan Kitchin
Günter Kleindl
Roger Knobbe
John M. Kowalski
Bruce P. Kraemer
Thomas Kuehnel
Geng-Sheng Kuo
Denis Kuwahara
Paul A. Lambert
David S. Landeta
Jim Lansford
Colin Lanzl
Kim Laraqui
Peter Larsson
David J. Leach, Jr.
Martin Lefkowitz
Onno Letanche
Sheung Li
William Li
Yunxin Li
Jie Liang
Isaac Lim Wei Lih
Shawn Liu
Jay Livingston
Titus Lo
Peter Loc
Ralph Lombardo, Jr.
Luke Ludeman
Yeong-Chang Maa
Akira Maeki
Douglas Makishima
Mahalingam Mani
Roger Marks
Leslie A. Martin
Brian Mathews
Jo-Ellen F. Mathews
Mark Mathews
Conrad Maxwell
Ron McCallister
Justin McCann
Kelly McClellan
Gary McCoy
Bill McFarland
Gary McGarr
Bill McIntosh
Jorge Medina
Pratik Mehta
Robert C. Meier
Robert Miller
Khashayar Mirfakhraei
Sanjay Moghe
Tim Moore
Paul Moose
Mike Moreton
Robert Moskowitz
Oliver Muelhens
Peter Murphy
Peter Murray
Andrew Myles
Marco Naeve
Ravi Narasimhan
Kevin Negus
David B. Nelson
Dan Nemits
Chiu Ngo
Henry Nielsen
Toshi Nishida
Gunnar Nitsche
Erwin R. Noble
Tzvetan D. Novkov
Ivan Oakes
Timothy O'Farrell
Bob O'Hara
Yoshihiro Ohtani
Lior Ophir
Dirk Ostermiller
Richard H. Paine
Mike Paljug
Gregory Parks
Gavin Parnaby
Lizy Paul
Sebastien Perrot
Al Petrick
Anselmo Pilla
Victoria M. Poncini
James Portaro
Al Potter
Mike Press
Ron Provencio
Henry Ptasinski
Ali Raissinia
Murali Ramadoss
Noman Rangwala
Javad Razavilar
David Reed
Ivan Reede
Stanley A. Reible
Danny Rettig
Edward Reuss
Bill Rhyne
Jim Richards
David Richkas
Maximilian Riegel
Carlos A. Rios
Benno Ritter
Kent G. Rollins
Stefan Rommer
Jon Rosdahl
Rob Roy
Gunnar Rydnell
Kenichi Sakusabe
Anil K. Sanwalka
Edward Schell
Sid Schrum
Joe Sensendorf
Rick Shaw
Yangmin Shen
Matthew Sherman
Matthew B. Shoemake
William Shvodian
Aman Singla
David Skellern
Donald I. Sloan
Kevin Smart
Dave Smith
H. Keith Smith
V. Srinivasa Somayazulu
Wei-Jei Song
Amjad Soomro
Gary Spiess
Geetha Srikantan
Dorothy V. Stanley
Adrian Stephens
Spencer Stephens
William M. Stevens
Carl R. Stevenson
Susan Storma
Michael Su
Barani Subbiah
Minoru Takemoto
Pek-Yew Tan
Teik-Kheong Tan
Takuma Tanimoto
Roger Teague
Carl Temme
John Terry
Yossi Texerman
Jerry A. Thrasher
James D. Tomcik
Walt Trzaskus

Allen Tsai
Chih C. Tsien
Tom Tsoulogiannis
Khaled Turki
Mike Tzamaloukas
Toru Ueda
Niels Van Erven
Wim J. van Houtum
Patrick Vandenameele
Dmitri Varsanofiev
Jagannatha L. Venkatesha
Mahesh Venkatraman
Madan Venugopal
Alex Vereshchak
Ritesh Vishwakarma
John Vollbrecht
Toan X. Vu
Tim Wakeley
Jesse R. Walker
Thierry Walrant
Christopher Ware
Fujio Watanabe
Mark Webster
Mathew Welborn
Menzo Wentink
Doug Whiting
Peter K. Williams
Richard G. C. Williams
Steven D. Williams
Harry Worstell
Charles R. Wright
Liwen Wu
Yang Xiao
Shugong Xu
Hidehiro Yamashita
Wen-Ping Ying
Kit Yong
Albert Young
Heejung Yu
Patrick Yu
Chris Zegelin
Glen Zorn
Arnoud Zwemmer
Jim Zyren

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

James Allen
Gholamreza Arefi-Anbarani
Eladio Arvelo
Morris Balamut
John Barnett
Vilas Bhade
Benjamin Chen
Todor Cooklev
Todd Cooper
Kenneth D. Cornett
Vern Dubendorf
Dr. Sourav Dutta
Avraham Freedman
Theodore Georgantas
Ian C. Gifford
James Gilb
Robert Grow
Rajugopal Gubbi
Jose Gutierrez
Chris Guv
Zion Hadad
Karen Halford
Steve Halford
Simon Harrison
Robert F. Heile
Downing Hopkins
Srinivas Kandala
James Kemerling
Stuart J. Kerry
Pat Kinney
Joe Kubler
Jim Lansford
Pi-Cheng Law
Daniel Levesque
Jie Liang
Randolph Little
Gregory Luri
Roger Marks
Peter Martini
George Miao
Andrew Myles
Paul Nikolich
Erwin Noble
Timothy O’Farrell
Chris Osterloh
Richard Paine
John Pardee
Subbu Ponnuswamy
Hugo Pues
Vikram Punj
Mike Rudnick
Osman Sakr
John Sarallo
John Sargent
Michael Seals
Stephen J. Shellhammer
Neil Shipp
Gil Shultz
Tom Siep
Kevin Smart
Carl Stevenson
Larry Telle
William Watte
Hung-yu Wei
Forrest Wright
Oren Yuen

When the IEEE-SA Standards Board approved this recommended practice on 12 June 2003, it had the following membership:

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

Andy Ickowicz
IEEE Standards Project Editor

Contents

IEEE Recommended Practice for Information technology— 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 Bands

1. Overview

This recommended practice addresses the issue of coexistence of wireless personal area networks (WPAN) and wireless local area networks (WLAN). These wireless networks often operate in the same unlicensed frequency band. This recommended practice describes coexistence mechanisms that can be used to facilitate coexistence of WPANs (i.e., IEEE Std 802.15.1™-2002[1]) and WLANs (i.e., IEEE Std 802.11b™-1999). The unlicensed frequency bands used by each wireless technology are specified within its respective standard. This recommended practice also describes a computer model of the mutual interference between IEEE Std 802.15.1-2002 and IEEE Std 802.11b-1999 for information.

1.1 Scope

The scope is to develop a recommended practice for an IEEE 802.15™ WPAN that coexists with other selected wireless devices operating in unlicensed frequency bands, to suggest modifications to other IEEE 802.15 standards to enhance coexistence with other selected wireless devices operating in unlicensed frequency bands, and to suggest recommended practices for IEEE Std 802.11™, 1999 Edition devices to facilitate coexistence with IEEE 802.15 devices operating in unlicensed frequency bands.

The scope of this recommended practice is limited to coexistence of IEEE Std 802.15.1-2002 WPANs and IEEE Std 802.11b-1999 WLANs. This recommended practice will cover the IEEE Std 802.11b-1999 direct sequence spread spectrum standard at data rates of 1, 2, 5.5, and 11 Mbit/s. Both IEEE 802.11™ and IEEE 802.15 are continuing to work on additional standards.

[1]Information on references can be found in Clause 2.

1.2 Purpose

Usage models exist that presume coexistence of IEEE 802.15 devices with other wireless devices operating in unlicensed frequency bands. The purpose of this recommended practice is to facilitate coexistence of IEEE 802.15 WPAN devices with selected other wireless devices[2] operating in unlicensed frequency bands. The intended users of this recommended practice include IEEE 802 WLAN developers, as well as designers and consumers of wireless products being developed to operate in unlicensed frequency bands.

This recommended practice includes a computer model of the mutual interference of an IEEE 802.11b WLAN and IEEE 802.15.1 WPAN. This model can be used to predict the impact of the mutual interference between these wireless systems. The model includes many parameters that can be modified to fit various user scenarios.

This recommended practice defines several coexistence mechanisms that can be used to facilitate coexistence of WLAN and WPAN networks. The several coexistence mechanisms defined in this recommended practice are divided into two classes: collaborative and non-collaborative. A collaborative coexistence mechanism can be used when there is a communication link between the WLAN and WPAN networks. This is best implemented when both a WLAN and WPAN device are embedded into the same piece of equipment (e.g., an IEEE 802.11b card and an IEEE 802.15.1 module embedded in the same laptop computer). A non-collaborative coexistence mechanism does not require any communication link between the WLAN and WPAN.

2. References

This recommended practice shall be used in conjunction with the following publications. If the following publications are superseded by an approved revision, the revision shall apply.

IEEE Std 802.11, 1999 Edition (R2003) (ISO/IEC 8802-11: 1999), IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Network—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.[3, 4]

IEEE Std 802.11b-1999 (Supplement to ANSI/IEEE Std 802.11, 1999 Edition), Supplement to IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band.

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™).

[2]The term “selected wireless devices” includes the following: a) Other 802 devices, and b) other wireless devices in the international marketplace operating in the same frequency band as an IEEE 802.15 WPAN. We will limit our scope to dealing with devices that have usage scenarios that assume IEEE 802.15 devices will coexist with these selected and that we are able to obtain technical specification on these selected devices.

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

[4]The IEEE standards referred to in Clause 2 are trademarks belonging to the Institute of Electrical and Electronics Engineers, Inc.

3. Definitions, terms, acronyms, abbreviations, terminology, and variables

For the purposes of this recommended practice, the following subclauses contain the applicable definitions and terms; acronyms and abbreviations; and terminology and variables. *The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition* [B12] should be referenced for terms not defined in this clause.

3.1 Definitions and terms

3.1.1 backward compatible: The ability of one "new" system to interwork with another "old" system. In this case the different set of rules implies that the new set of rules is a modification of the old set of rules. A subset of interworking.

3.1.2 coexistence: The ability of one system to perform a task in a given shared environment where other systems have an ability to perform their tasks and may or may not be using the same set of rules.

3.1.3 coexistence mechanism: A method for reducing the interference of one system, which is performing a task, on another different wireless system, that is performing its task.

3.1.4 collaborative coexistence mechanism: A coexistence mechanism in which the two systems shall exchange information.

3.1.5 collocation: When two devices' antennas are positioned less than 0.5 meters apart.

3.1.6 conformance: The ability of a system to follow a single set of rules.

3.1.7 connection-oriented: Data transmission in which the information-transfer phase is preceded by a call-establishment phase and followed by a call-termination phase. (See Weik [B17].)

3.1.8 frequency-hopping: A technique in which the instantaneous carrier frequency of a signal is periodically changed, according to a predetermined code, to other positions within a frequency spectrum that is much wider than that required for normal message transmission. (See Weik [B17].)

3.1.9 interference: In a communication system, extraneous power entering or induced in a channel from natural or man-made sources that might interfere with reception of desired signals or the disturbance caused by the undesired power. (See Weik [B17].)

3.1.10 interoperable: The ability of two systems to perform a given task using a single set of rules.

3.1.11 interworking: The ability of two systems to perform a task given that each system implements a different set of rules.

3.1.12 medium sharing element: Defines how IEEE 802.11 traffic and non-IEEE 802.11 traffic share access to the medium.

3.1.13 multipath fading: Fading due to the propagation of an electromagnetic wave over many different paths, dissipating energy and causing distortion, particularly by signal cancellation at the destination because of differences in arrival time due to the different paths. (See Weik [B17].)

3.1.14 non-collaborative coexistence mechanism: A coexistence mechanism in which the two systems shall not exchange information.

3.1.15 operable: The ability of a system to perform the functions as expected.

3.1.16 period of stationarity: The time period over which the parameters defining the transmissions of the devices being modeled do not change.

3.1.17 propagation: The movement or transmission of a wave in a medium or in free space, usually described in terms of phase or group velocity. (See Weik [B17].)

3.1.18 spread spectrum: A communication technique in which the information-modulated signal is transmitted in a bandwidth that is considerably greater than the frequency content of the original information. (See Weik [B17].)

3.1.19 synchronous: Pertaining to events that occur at the same time or at the same rate. (See Weik [B17].)

3.1.20 synchronous connection-oriented link: A point-to-point link between a master and a single slave in the piconet.

3.2 Acronyms and abbreviations

ACL	asynchronous connectionless
ACK	acknowledgement packet
AFH	adaptive frequency-hopping
AP	access point
ARQ	automatic repeat request
AWGN	additive white Gaussian noise
AWMA	alternating wireless medium access
BER	bit error rate
BPF	bandpass filter
BPSK	binary phase shift keying
CCA	clear channel assessment
CCK	complementary code keying
CRC	cyclic redundancy check
CSMA/CA	carrier sense multiple access with collision avoidance
CW	contention window
DBPSK	differential binary phase shift keying
DCF	distributed coordination function
DIFS	distributed (coordination function) interframe space
DQPSK	differential quadrature phase shift keying
DSSS	direct sequence spread spectrum
FCS	frame check sequence
FEC	forward error correction
FH	frequency-hopping
FHSS	frequency-hopping spread spectrum
GFSK	Gaussian frequency shift keying
GLRT	generalized likelihood-ratio test
HEC	header error check
ICR	interference collision ratio
I&D	integrate and dump
LAP	lower address parts
LD	limiter-discriminator
LDI	limiter-discriminator with integrate and dump
LMP	link manager protocol
L2CAP	logical link control and adaptation protocol
MAC	medium access control
MIB	management information base

MLME	MAC sublayer management entity
MPDU	MAC protocol data unit
MSE	medium sharing element
PCF	point coordination function
PER	packet error rate
PHY	physical
PLCP	physical layer convergence protocol
PN	pseudorandom noise (e.g., PN code sequence)
PPDU	physical protocol data unit
PSDU	physical service data unit
PTA	packet traffic arbitration
QoS	quality of service
QPSK	quadrature phase shift keying
RF	radio frequency
RLSL	recursive least-squares lattice
RSSI	received signal strength indication
RX	receive/receiver/receiving
SCO	synchronous connection-oriented
SER	symbol error rate
SINR	signal to interference plus noise ratio (s/(i+n))
SIFS	short interframe space
SIR	signal to interference ratio (s/i)
SNR	signal to noise ratio (s/n)
STA	station
TBTT	target beacon transmit time
TDMA	time-division multiple access
TU	time unit (as defined in IEEE Std 802.11, 1999 Edition)
TX	transmit/transmitter/transmission
UAP	upper address parts
WLAN	wireless local area network
WPAN	wireless personal area network

3.3 Terminology and variables[5]

Packet: Is used consistently through this recommendation to mean “medium access control (MAC) frame” in the context of IEEE 802.11 and “baseband packet” in the context of IEEE 802.15.1.

Packet error rate: The probability of a packet being received with one or more uncorrected bit errors.

[5]The terminology and variables listed in this subclause are only applicable within this recommended practice. Application of these outside of this recommended practice is not applicable. This is why they have their own subclause within this clause.

f_{adp} next adapted hop-frequency, f_{adp} is an element of S_G or f_{adp} is an element of S_{BK}
f_{hop} next hop-frequency from the IEEE 802.15.1 hop kernel, f_{hop} is indexed by an element of [0, ..., 78]
k_{hop} index that points to the next hop-frequency
N_B number of “bad” channels ($N_B = | S_B |$)
N_{BK} number of “bad” channels kept in the adapted hopping sequence ($N_{BK} = | S_{BK} |$)
N_{BR} number of “bad” channels removed from the adapted hopping sequence ($N_{BR} = | S_{BR} |$)
N_G number of “good” channels ($N_G = | S_G |$)
N_{min} minimum number of hop channels (typically set by regulatory constraints)
$p(k)$ partition sequence
S set of all channels = S_G union S_{BK} union S_{BR} = S_G union S_B
S_B set of “bad” channels (or indices pointing to the "bad"channels)
S_{BK} set of “bad” channels (or indices) kept in the adapted hopping sequence
$S_{BK}(i)$ i-th channel of S_{BK} , i is an element of [0, ..., N_{BK} -1]
S_{BR} set of “bad” channels (or indices) removed from the adapted hopping sequence
S_G set of “good” channels (or indices pointing to the “good” channels)
$S_G(i)$ i-th channel of S_G , i is an element of [0, ..., N_G -1]
T_d time-out delay
T_S slot time (i.e., 625µs)

4. General descriptions

This clause describes in general terms 1) the issue that this recommended practice attempts to address; 2) the coexistence mechanisms being recommended to reduce the problem and when to use each coexistence mechanism; 3) the models used to evaluate the effects; and 4) an overview to the structure of this recommended practice.

4.1 Description of the interference problem

Because both IEEE Std 802.11b-1999 and IEEE 802.15.1-2002 specify operations in the same 2.4 GHz unlicensed frequency band, there is mutual interference between the two wireless systems that may result in severe performance degradation. There are many factors that effect the level of interference, namely, the separation between the WLAN and WPAN devices, the amount of data traffic flowing over each of the two wireless networks, the power levels of the various devices, and the data rate of the WLAN. Also, different types of information being sent over the wireless networks have different levels of sensitivity to the interference. For example, a voice link may be more sensitive to interference than a data link being used to transfer a data file. This subclause gives an overview of the mutual interference problem. Subsequent subclauses describe the modeling of the mutual interference and give illustrations of the impact of this mutual interference on both the WLAN and WPAN networks.

There are several versions of IEEE 802.11 physical (PHY) layer. All versions of IEEE 802.11 use a common MAC sublayer. When implementing distributed coordination function (DCF) the 802.11 MAC uses carrier sense multiple access with collision avoidance (CSMA/CA) for medium access control. The scope of this recommended practice is limited to DCF implementations of IEEE 802.11, and does not include point coordination function (PCF) implementations. Initially, 802.11 included both a 1- and 2-Mbit/s frequency-hopping spread spectrum (FHSS) PHY layer, as well as a 1- and 2-Mbit/s direct sequence spread spectrum (DSSS) PHY layer. The FHSS PHY layer uses a 1-MHz channel separation and hops pseudo-randomly over 79 channels. The DSSS PHY layer uses a 22 MHz channel and may support up to three non-overlapping channels in the unlicensed band.

Subsequently, the IEEE 802.11 DSSS PHY layer was extended to include both 5.5 and 11 Mbit/s data rates using complementary code keying (CCK). This high-rate PHY layer is standardized to be named IEEE

802.11b. This high-rate version includes four data rates: 1, 2, 5.5, and 11 Mbit/s. The channel bandwidth of the IEEE 802.11b PHY layer is 22 MHz.

The WPAN covered in this recommended practice is IEEE Std 802.15.1-2002, which is a 1-Mbit/s FHSS system. The IEEE 802.15.1 PHY layer uses the same 79, 1 MHz-wide channels that are used by the FHSS version of IEEE 802.11. IEEE 802.15.1 hops pseudo-randomly at a nominal rate of 1600 hops/second. The IEEE 802.15.1 MAC sublayer supports a master/slave topology referred to as a piconet. The master controls medium access by polling the slaves for data and using scheduled periodic transmission for voice packets.

The following is a brief description of the interference problem for each of the three systems: IEEE 802.11 frequency-hopping (FH), IEEE 802.11b, and IEEE 802.15.1.

4.1.1 IEEE 802.11 FH WLAN in the presence of IEEE 802.15.1 interference

The IEEE 802.11 FH WLAN has the same hopping channels as the IEEE 802.15.1 WPAN. However, the two systems operate at very different hopping rates. IEEE 802.11 FH specifies a hopping rate of greater than 2.5 hops/second, with typical systems operating at 10 hops/second. IEEE 802.15.1 specifies a maximum hopping rate of 1600 hops/second for data transfer. So while IEEE 802.11 FH dwells on a given frequency for approximately 100 ms, IEEE 802.15.1 will have hopped 160 times. So the odds are that IEEE 802.15.1 will hop into the frequency used by IEEE 802.11 FH several times while IEEE 802.11 FH is dwelling on a given channel. IEEE 802.11 FH packets will be corrupted by the IEEE 802.15.1 interference whenever IEEE 802.15.1 hops into the channel used by IEEE 802.11 FH, assuming the IEEE 802.15.1 power level is high enough to corrupt the IEEE 802.11 FH packet at the IEEE 802.11 FH receiver. It is also possible for the IEEE 802.11 FH WLAN packet to be corrupted by the IEEE 802.15.1 interference if the IEEE 802.15.1 packet is sent in an adjacent channel to the IEEE 802.11 FH data. For example, if currently IEEE 802.11 FH is using the 2440 MHz channel then the two adjacent channels are at 2439 and 2441 MHz. Usually, there is only limited attenuation in adjacent channels. It is likely that there will be limited interference if the IEEE 802.15.1 WPAN is greater than one channel away from the current IEEE 802.11 FH channel. Whether an IEEE 802.11 packet is corrupted or not depends on how close the IEEE 802.15.1 unit is to the IEEE 802.11 FH unit, because that effects the interference power level.

The IEEE 802.11 MAC sublayer incorporates automatic repeat request (ARQ) to insure reliable delivery of data across the wireless link. So there is little chance that the data will be lost. The impact of interference on the WLAN is that the delivered data throughput decreases and the network latency increases. The application's requirements determine if these degradations are tolerable.

4.1.2 IEEE 802.11b WLAN in the presence of IEEE 802.15.1 interference

The high-rate IEEE Std 802.11b-1999 defines a frequency-static WLAN that supports four data rates: 1, 2, 5.5, and 11 Mbit/s. Most implementations allow manual or automatic modification of the data rate. The higher rates are desirable for many applications but the distance of transmission using the higher rates is less than that of the lower rates. Many implementations automatically scale the data rate to the highest data rate that is sustainable to each WLAN mobile unit.

The bandwidth of IEEE 802.11b is up to 22 MHz. There is a potential packet collision between a WLAN packet and an IEEE 802.15.1 packet when the WPAN hops into the WLAN passband. Since the bandwidth of the IEEE 802.11b WLAN is 22 MHz, as the IEEE 802.15.1 WPAN hops around the unlicensed band, 22 of the 79 IEEE 802.15.1 channels fall within the WLAN passband.

Because there are four data rates defined within IEEE 802.11b, the temporal duration of the WLAN packets may vary significantly for packets carrying the exact same data. The longer the duration of the WLAN packet, the more likely that it may collide with an interfering WPAN packet.

One of the important issues that effects the level of interference is the WLAN automatic data rate scaling. If it is implemented and enabled, it is possible for the WPAN interference to cause the WLAN to scale to a lower data rate. At a lower data rate the temporal duration of the WLAN packets is increased. This increase in packet duration may lead to an increase in packet collisions with the interfering WPAN packets. In some implementations, this may lead to yet a further decrease in the WLAN data rate. This may result in the WLAN scaling down its data rate to 1 Mbit/s.

The IEEE 802.11 MAC sublayer incorporates ARQ to insure reliable delivery of data across the wireless link. So there is little chance that the data will be lost. The effect this has on the WLAN is that the delivered data throughput decreases and the network latency increases. The application’s requirements determine if these degradations are tolerable.

4.1.3 IEEE 802.15.1 in the presence of an IEEE 802.11 FH interferer

Both IEEE 802.15.1 and IEEE 802.11 FH use FHSS by using the same 79 channels. Both FH systems are susceptible to interference on the channel in use and the two adjacent channels. Also, because IEEE 802.15.1 uses short packets the packet error rate (PER) in IEEE 802.15.1 in the presence of IEEE 802.11 FH is not very significant.

IEEE 802.15.1 uses two types of links between the piconet master and the piconet slave. For data transfer IEEE 802.15.1 uses an asynchronous connectionless (ACL) link. The ACL link incorporates ARQ to ensure reliable delivery of data. IEEE 802.15.1 voice communications use a synchronous connection-oriented (SCO) link. On account of the SCO link does not support ARQ, there will be some perceivable degradation in voice quality during periods of IEEE 802.11 FH interference. The detailed model described later quantifies the level of PER. The network throughput would decrease and the network latency would increase for IEEE 802.11 FH interference. A large number of errors on a SCO link can cause voice quality degradation.

4.1.4 IEEE 802.15.1 in the presence of an IEEE 802.11b interferer

IEEE 802.15.1 uses FHSS, while IEEE 802.11b uses DSSS and CCK. The bandwidth of IEEE 802.11b is 22 MHz. 22 of the 79 hopping channels available to IEEE 802.15.1 hops are subject to interference. A FH system is susceptible to interference from the adjacent channels as well. This increases the total number of interference channels from 22 to 24. The detailed model, which is described later, quantifies the level of PER based on these assumptions. The IEEE 802.11b is used because it represents a worse interferer than the IEEE 802.11 FH. The results from this scenario for data transfers are that the network throughput would decrease and the network latency would increase, in the presence of IEEE 802.11b interference. The PER for a SCO link may cause voice quality degradation.

4.2 Overview of the coexistence mechanisms

There are two categories of coexistence mechanisms: collaborative and non-collaborative. Collaborative coexistence mechanisms exchange information between two wireless networks. That is in this case a collaborative coexistence mechanism requires communication between the IEEE 802.11 WLAN and the IEEE 802.15 WPAN. Non-collaborative mechanisms do not exchange information between two wireless networks. These coexistence mechanisms are only applicable after a WLAN or WPAN are established and user data is to be sent. These coexistence mechanisms will not help in the process for establishing a WLAN or WPAN.

Both types of coexistence mechanisms are designed to mitigate interference resulting from the operation of IEEE 802.15.1 devices in the presence of frequency static or slow-hopping WLAN devices (for example IEEE 802.11b). Note that interference due to multiple IEEE 802.15.1 devices is mitigated by frequency-hopping. All collaborative coexistence mechanism described in this recommended practice are intended to be used when at least one WLAN station and WPAN device are collocated within the same physical unit.

When collocated, there needs to be a communication link between the WLAN and WPAN devices within this physical unit, which could be a wired connection between these devices or an integrated solution. The exact implementation of this communication link is outside the scope of this recommended practice.

Non-collaborative coexistence mechanisms are intended to be used when there is no communication link between the WLAN and WPAN.

Table 1 shows the coexistence mechanisms listed in this recommend practice. The “Name” column assigns the name of the coexistence mechanism. The “Type” column lists whether it is collaborative or non-collaborative. The “Clause/Annex” column gives the location within this recommended practice where the description of this mechanism may be found.

Table 1—Listing of the coexistence mechanisms

Name	Type	Clause/Annex
Alternating wireless medium access	collaborative	Clause 5
Packet traffic arbitration	collaborative	Clause 6
Deterministic interference suppression	collaborative	Clause 7
Adaptive interference suppression	non-collaborative	Clause 8
Adaptive packet selection	non-collaborative	Clause 9
Packet scheduling for ACL links	non-collaborative	Clause 10
Packet scheduling for SCO links	non-collaborative	Annex A
Adaptive frequency-hopping	non-collaborative	Annex B

4.2.1 Collaborative coexistence mechanisms

The three collaborative coexistence mechanisms defined in this recommended practice consist of two MAC sublayer techniques (see Clause 5 and Clause 6) and one PHY layer technique (see Clause 7). Both MAC sublayer techniques involve coordinated scheduling of packet transmission between the two wireless (WLAN and WPAN) networks. The PHY layer technique is a programmable notch filter in the IEEE 802.11b receiver to notch out the narrow-band IEEE 802.15.1 interferer. These collaborative mechanisms may be used separately or combined with others to provide a better coexistence mechanism.

The collaborative coexistence mechanism provides coexistence of a WLAN (in particular IEEE 802.11b) and a WPAN (in particular IEEE 802.15.1[6]) by sharing information between collocated IEEE 802.11b and IEEE 802.15.1 radios and locally controlling transmissions to avoid interference. These mechanisms are interoperable with legacy devices that do not include these features.

There are two modes of operation and the mode is chosen depending on the network topology and supported traffic. In the first mode, both IEEE 802.15.1 SCO and ACL traffic are supported where SCO traffic is given higher priority than the ACL traffic in scheduling. The second mode is based on time-division multiple

[6]Although this recommended practice consistently references IEEE 802.15.1, and not Bluetooth®, the mechanism is equally applicable to both IEEE 802.15.1 and Bluetooth®.

access and is used when there is ACL traffic in high piconet density areas. In time-division multiple access (TDMA) mode, the IEEE 802.11b beacon-to-beacon interval is subdivided into two subintervals: one subinterval for IEEE 802.11b and other subinterval for IEEE 802.15.1. Since each radio has its own subinterval, both radios will operate properly, due to total orthogonality. This technique does require an additional feature to restrict when the IEEE 802.15.1 master transmits. The mode to be used is chosen under the command of the access point (AP) management software. Frequency nulling may be used in conjunction with these modes to further reduce interference.

Both alternating wireless medium access (AWMA) and packet traffic arbitration (PTA) may be combined to produce a better coexistence mechanism. This is not described in detail, but in Figure 1 the overall structure of the combined collaborative coexistence mechanisms is shown.

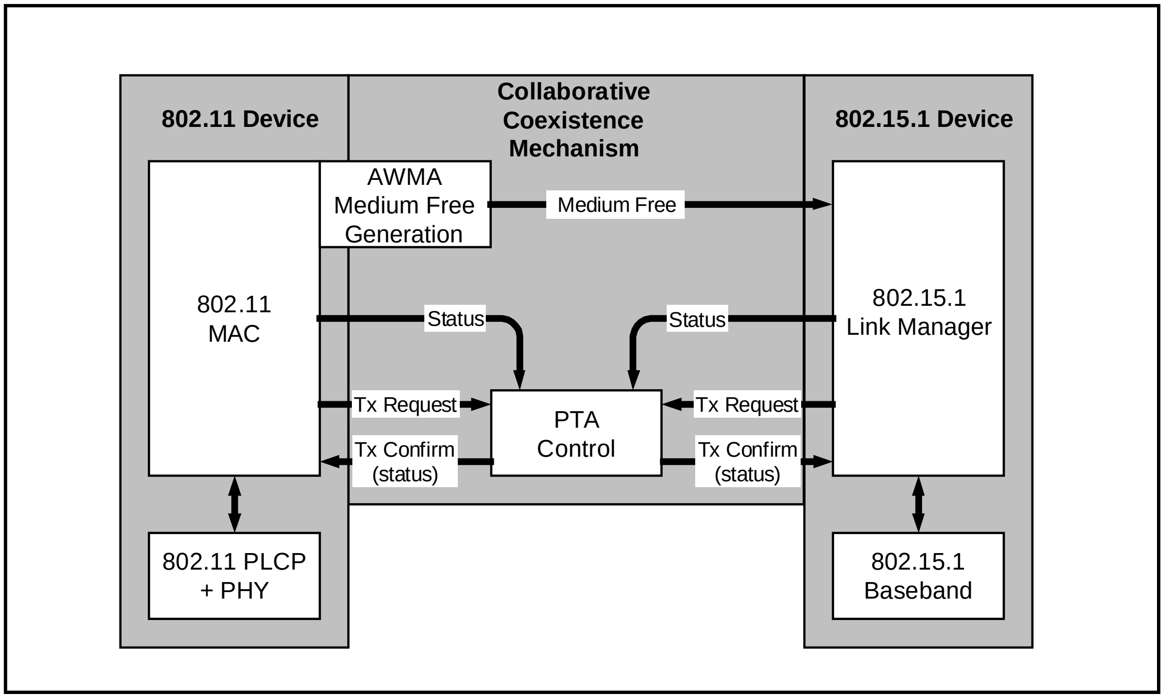

Figure 1—Overall structure of 802.11b / 802.15.1 combined AWMA and PTA collaborative coexistence mechanism

4.2.2 Recommendations on the utilization of collaborative coexistence mechanisms

It is recommended that when it is possible, or necessary, to collocate a WLAN and WPAN device within the same physical unit (e.g., laptop computer), that either the AWMA collaborative coexistence mechanism or the PTA collaborative coexistence mechanism be used. If the PTA mechanism is used it is also recommended that the deterministic interference suppression mechanism be used in concert with the PTA mechanism. While PTA can be used without deterministic interference suppression, the combination of the two mechanisms leads to increased WLAN/WPAN coexistence.

If there is a high density of physical units incorporating both a WLAN and WPAN device in a common area (greater than or equal to three units in a circle of radius 10 meters) and WPAN SCO link (voice link) is not being utilized, then it is recommended that the AWMA mechanism be used. If the density of units incorporating both the WLAN and WPAN devices is low (less than three units in a circle with a radius of 10 meters), or the WPAN SCO link is used, then it is recommended that the PTA mechanism be used in concert with the deterministic interference suppression mechanism.

4.2.3 Non-collaborative coexistence mechanisms

This recommended practice describes several methods (See Clause 8, Clause 9, and Clause 10) that enhance the performance of the IEEE 802.15.1 and IEEE 802.11 networks through the use of adaptive interference suppression of IEEE 802.11b devices, adaptive packet selection, and packet scheduling for ACL links. These methods do not require the collaboration between the IEEE 802.11 devices and the IEEE 802.15.1 devices. Therefore, they belong to the general category of non-collaborative coexistence mechanisms.

Two other methods [packet scheduling for SCO links and adaptive frequency-hopping (AFH) for the IEEE 802.15.1 devices] are provided as information in Annex A and Annex B, respectively.

The key idea for adaptive packet selection and scheduling methods is to adapt the transmission according to channel conditions. For instance, if the channel is dominated by interference from an IEEE 802.11b network, the PER will be mainly due to collisions between IEEE 802.15.1 and IEEE 802.11 systems, instead of bit errors resulting from noise. Packet types that do not include forward error correction (FEC) protection could provide better throughput if combined with intelligent packet scheduling. The foundation for the effectiveness of these types of methods is to be able to figure out the current channel conditions accurately and in a timely manner. Channel estimation may be done in a variety of ways: received signal strength indication (RSSI), header error check (HEC) decoding profile, bit error rate (BER) and PER profile, and an intelligent combination of all of the above (see Clause 11).

There are five non-collaborative mechanisms described in this recommended practice. At least two of these share a common function called channel classification, which is contained in a separate clause under that heading. Three mechanisms are covered under the second item (b) in the following list:

a) *adaptive interference suppression*. A mechanism based solely on signal processing in the physical layer of the WLAN.
b) *adaptive packet selection and scheduling*. IEEE 802.15.1 systems utilize various packet types with varying configurations such as packet length and degree of error protection used. By selecting the best packet type according to the channel condition of the upcoming frequency hop, better data throughput and network performance may be obtained. In addition, by carefully scheduling packet transmission so that the IEEE 802.15.1 devices transmit during hops that are outside the WLAN frequencies and refrain from transmitting while in-band, interference to WLAN systems could be avoided/minimized and at the same time increase the throughput of the IEEE 802.15.1 systems.
c) *adaptive frequency-hopping (AFH)*. IEEE 802.15.1 systems frequency hop over 79 channels (in the U.S.) at a nominal rate of 1600 hops/second in connection state, and 3200 hops/second in inquiry and page states. By identifying the channels with interference, it is possible to change the sequence of hops such that those channels with interference (“bad” channels) are avoided. From traffic type and channel condition, a partition sequence is generated as input to the frequency re-mapper, which modifies hopping frequencies to avoid or minimize interference effects.

4.2.4 Recommendations on the utilization of non-collaborative coexistence mechanisms

When it is not possible, or necessary, to collocate a WLAN and WPAN device within the same physical unit, then a non-collaborative coexistence mechanism may be the only practical method. There are possible range limitations under which a non-collaborative mechanism may not be sufficient, however. For example, when an IEEE 802.11b system and an IEEE 802.15.1 system (Class 3) are operated 30 centimeters apart, the IEEE 802.15.1 signal will be considerably above the detection threshold of the WLAN system, even when out of band; thus, non-collaboration schemes relying on channel estimation and interference detection will be unable to prevent interference in these short range situations.[7]

[7]Class 3 IEEE 802.15.1 is 0dBm. Free space path loss at 30 centimeters is 30dB. The IEEE 802.11b specification requires 35dB of attenuation outside of the desired passband, and a minimum detection sensitivity of –76dBm at 11Mbit/s. Even when out of band, the IEEE 802.15.1 signal will be at least 11dB above the detection threshold, which will significantly degrade IEEE 802.11b reception.

The non-collaborative mechanisms considered range from adaptive frequency-hopping to packet scheduling and traffic control. They all use similar techniques for detecting the presence of other devices in the band such as measuring the packet or frame error rate, the signal strength or the signal to interference ratio (often implemented as the RSSI).

For example, each device can maintain a frame error rate measurement per frequency used. FH devices can then infer which frequencies are occupied by other users of the band and thus modify their frequency hopping pattern. They can even choose not to transmit on a certain frequency if that frequency is inferred to be occupied.

MAC sublayer packet selection mechanisms consider encapsulation rules and use the variety of IEEE 802.15.1 packet lengths to avoid overlap in frequency between IEEE 802.11 and IEEE 802.15.1. In other words, the IEEE 802.15.1 scheduler knows to use the packet length of proper duration (1, 3, or 5 slots) in order to skip the so-called “bad” frequency.

It is recommended that AFH be used when appropriate changes to the IEEE Std 802.15.1-2002 hopping sequence have been implemented.

Furthermore, it is recommended that interference aware packet scheduling and traffic control mechanisms be implemented. These mechanisms can be implemented either separately or in combination with other coexistence schemes such as AWMA, PTA, or AFH for additional performance improvements.

It is recommended that adaptive interference suppression be used with all of the above-mentioned mechanisms because it operates at the physical layer; it can also be used by itself. It is recommended that the adaptive interference suppression filter be used when there is sufficient IEEE 802.15.1 interference to noticeably degrade performance and delaying the IEEE 802.11 traffic is not sufficient. Specifically, delay sensitive traffic such as streaming media will benefit from the use of this mechanism.

4.3 Interference model

The coexistence modeling approach used is based on detailed simulation models for the radio frequency (RF) channel and the MAC sublayer that were developed using OPNET Modeler[8] and the PHY layers that were developed in ANSI C[9].

The PHY layer models for the IEEE 802.15.1 and IEEE 802.11 transceivers are based on models developed in ANSI C. The MAC sublayer models interface with these PHY layer models, and the integrated MAC and PHY layer simulation models constitute an evaluation framework that is critical to studying the various intricate effects between the MAC sublayer and PHY layer. Although interference is typically associated with the RF channel modeling and measured at the PHY layer, it may significantly impact the performance of higher layers. Changes in the behavior of the MAC sublayer protocol and the associated data traffic distribution impact the interference scenario and the overall system performance.

The physical layer models, source code for the physical layer analytical model, MAC sublayer models, data traffic models, performance metrics, and the coexistence modeling results, are all contained in separate informative annexes (See Annex C, Annex D, Annex E, Annex F, Annex G, and Annex H).

[8]The OPNET Modeler ®, a network technology development environment, is a software application provided by OPNET Technologies, Inc.™. More info: http://www.opnet.com/products/modeler/home.html. The use of OPNET Technologies product to prepare this recommended practice does not constitute an endorsement of OPNET Modeler ® by the IEEE LAN/MAN Standards Committee or by the IEEE.
[9]ANSI X3.159-1989 Standard C (ISO/IEC 9899:1990).

4.4 Overview of the recommended practice

The layout of the recommended practice consists of an individual clause or informative annex for each coexistence mechanism. The collaborative coexistence mechanisms are described first followed by the non-collaborative coexistence mechanisms. A clause devoted to channel classification ends the normative clauses. Finally numerous informative annexes are included that provide other coexistence mechanisms, performance or simulation results supporting a particular coexistence mechanism, and some other background information.

The numerous informative annexes contain: a non-collaborative coexistence mechanism for packet scheduling for SCO links, AFH, performance results for AWMA, PTA, deterministic- and adaptive- interference suppression, the theoretical coexistence models; experimental validation of models; the PHY layer models between IEEE Std 802.11, 1999 Edition and IEEE Std 802.15.1-2002 and their related RF channel models under various characteristics; the MAC sublayer models for IEEE Std 802.11, 1999 Edition and IEEE Std 802.15.1-2002; their related various data traffic models; the performance metrics used to evaluate the results of simulations; the results of the coexistence modeling; and the bibliography.

5. Alternating wireless medium access

AWMA utilizes a portion of the wireless IEEE 802.11 beacon interval for wireless IEEE 802.15 operations. From a timing perspective, the medium assignment alternates between usage following IEEE 802.11 procedures and usage following IEEE 802.15 procedures. Each wireless network restricts their transmissions to the appropriate time segment, which prevents interference between the two wireless networks.

In AWMA, a WLAN radio and a WPAN radio are collocated in the same physical unit. This allows for a wired connection between the WLAN radio and the WPAN radio. This wired communication link is used by the collaborative coexistence mechanism to coordinate access to the wireless medium, between the WLAN and WPAN.

The AWMA mechanism uses the shared clock within all the WLAN-enable devices and thus all WLAN devices connected to the same WLAN AP share common WLAN and WPAN time intervals. Therefore, all devices connected to the same AP restrict their WLAN traffic and WPAN traffic to non-overlapping time intervals. As such, there will be no WLAN/WPAN interference for any devices connected to the same WLAN AP. In the case of multiple APs, typically the APs are not synchronized. In that case there will be some residual interference between WPAN devices synchronized with on WLAN AP and WLAN devices synchronized with another AP. If the WLAN APs are synchronized then this residual interference can also be eliminated. Additional description of this synchronization issue is given in 5.1.

The IEEE 802.11 WLAN AP sends out a beacon at a periodic interval. The beacon period is T_B. AWMA subdivides this interval into two subintervals: one for WLAN traffic and one for WPAN traffic. Figure 2 illustrates the separation of the WLAN beacon interval into two subintervals. The WLAN interval begins at the WLAN target beacon transmit time (TBTT). The length of WLAN subinterval is T_{WLAN}, which is specified in the offset field of the medium sharing element (MSE) in the beacon. The WPAN subinterval begins at the end of the WLAN interval. The length of the WPAN subinterval is T_{WPAN}, which is specified in the duration field of the MSE in the beacon. The combined length of these two subintervals shall not be greater than the beacon period.

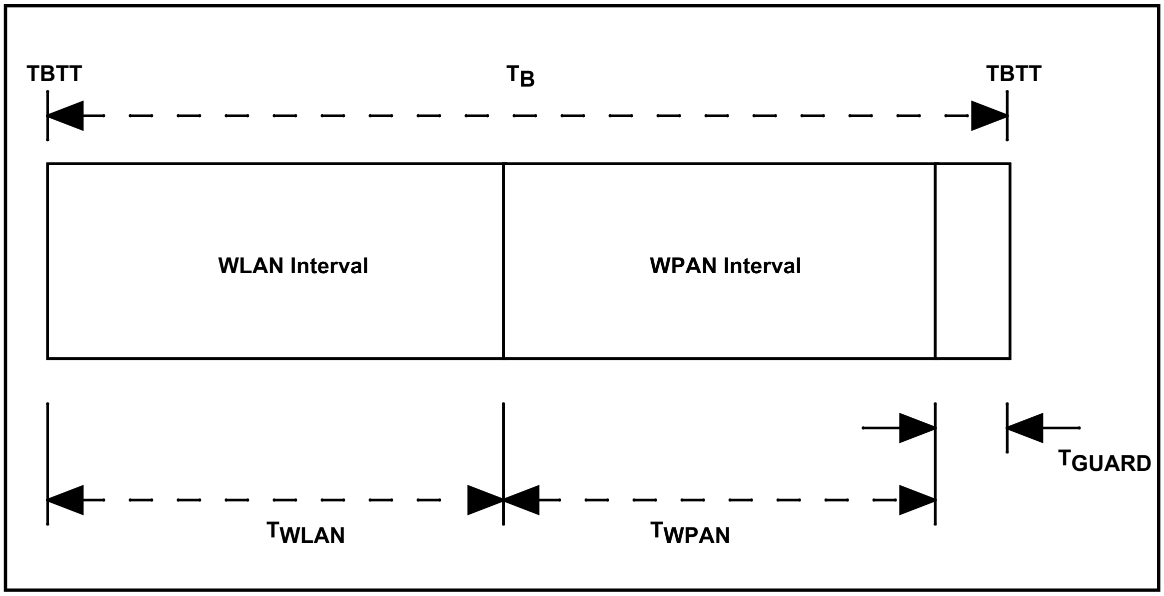

Figure 2—Timing of the WLAN and WPAN subintervals

The MSE in the beacon may also specify a guard band (T_{GUARD}) by setting a non-zero value in the guard field. The purpose of this guard band is to specify an interval immediately preceding the next expected beacon (i.e., TBTT) that is to be free of WPAN traffic. This guard band may be necessary to guarantee that all WPAN traffic has completed by the WLAN beacon time (i.e., before the next beacon needs to be sent).

If the offset field in the MSE of the beacon is greater than the beacon period, no WPAN subinterval shall exist. If the total value of the offset field and the duration field is greater than the beacon time, T_{WPAN} shall end at the next TBTT.

If the guard field in the MSE of the beacon is non-zero, and the beacon period minus the total value of the offset field and the duration field is less than the value of the guard field, T_{WPAN} shall end the value of the guard field prior to the next TBTT.

If the value in the offset field of the MSE is less than the beacon interval but the value of the offset field plus the value of the guard field is equal to or greater than the beacon interval, there shall be no WPAN subinterval.

Table 2 shows the range of values for these timers.

Table 2—Allowed range of values for T_{WPAN} and T_{GUARD}

Value	Minimum (TU)	Maximum (TU)
T_{WPAN}	0	32
T_{GUARD}	0	10

It is recommended to use AWMA whenever there is a high density of devices with collocated WLAN/WPAN radios. The AWMA mechanism not only eliminates interference between the collocated WLAN/WPAN radios, but also the radios in other nearby devices. The AWMA mechanism is also to be used when the WLAN and/or WPAN network bandwidth allocation needs to be deterministically controlled and not dependent on the traffic load of either the WLAN or WPAN.

Annex I provides information on the performance of WLAN and WPAN utilizing AWMA.

5.1 WLAN/WPAN synchronization

AWMA requires that a WLAN node and the WPAN master are collocated in the same physical unit (e.g., both within a single laptop computer). AWMA requires the WLAN node to control the timing of the WLAN and WPAN subintervals. All WLAN nodes connected to the same AP are synchronized, and hence have the same TBTT. As a result all units that implement AWMA have synchronized WLAN and WPAN subintervals. The WLAN node is required to send a physical synchronization signal to the WPAN master, which is in the same physical unit as the WLAN node. That synchronization signal specifies both the WLAN interval and the WPAN interval. This synchronization signal is called the medium free signal. Therefore, the medium is free of WLAN traffic when the medium free signal is true. Figure 3 illustrates the medium free signal.

The AWMA coexistence mechanism prevents interference between IEEE 802.11b and IEEE 802.15.1 by scheduling transmissions so that the WLAN and the WPAN radios do not transmit at the same time. For this mechanism to prevent interference between a WLAN and a WPAN device the two radios must be synchronized. There are three cases to consider in AWMA. They are the following:

a) The first case is when the WLAN and WPAN radios are collocated in the same physical device. These radios can easily be synchronized because they are in the same physical unit. This synchronization is implemented using the medium free signal sent from the WLAN radio to the collocated WPAN radio.
b) The second case is any WPAN device in the piconet with the collocated WPAN radio and any WLAN radio connected to the same AP as the collocated WLAN radio. Within the piconet all of the WPAN devices are synchronized to a common clock. Also, all of the WLAN stations attached to the same AP are also synchronized. The two sets of radios (WPAN piconet and the set of WLAN stations connected to the same AP) are all synchronized through the medium free signal sent between the collocated WLAN and WPAN radios. Therefore, in this case interference is also prevented because all these radios are synchronized.
c) The third case is that of a piconet device with the collocated WPAN and any WLAN station that is connected to a different AP than the collocated WLAN station. In this case the WPAN radio and the WLAN radio are not synchronized because the two APs are not synchronized. This situation can occur at the border between two WLAN cells, one cell covered by one AP and the other cell covered by the other AP. However, this third case can also be addressed by synchronizing the APs. This synchronization can be implemented by sending synchronization messages to the APs over the WLAN distribution medium. The implementation of the synchronization of WLAN APs is outside the scope of this recommended practice because this may be accomplished at higher layers.

An implementation of AWMA does not require synchronization of WLAN APs. If this AP synchronization is not implemented, interference is still prevented for the first two cases. However, the third case of WPAN/WLAN interference is not prevented. The interference in the third case is likely much lower than in the first case because the WLAN and the WPAN are not collocated in the same physical device. Therefore, this limitation with unsynchronized APs is not significant.

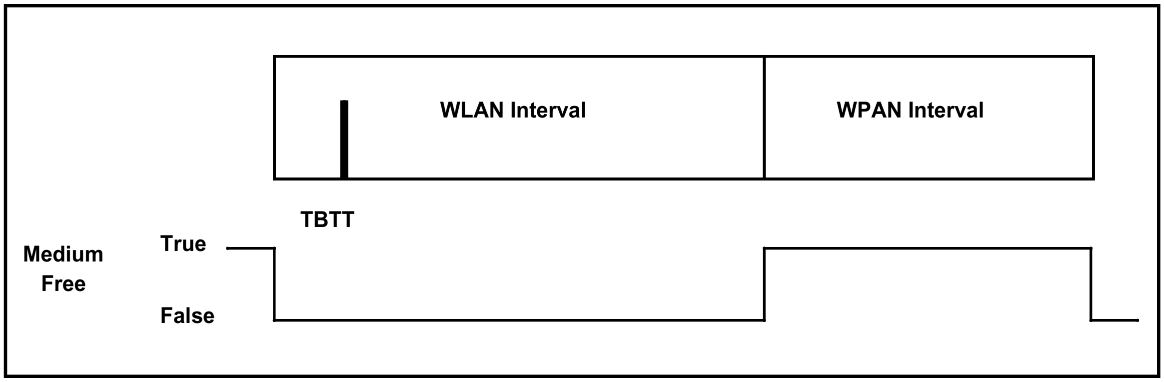

Figure 3—Medium free signal

5.2 Management of AWMA

Management of the AWMA coexistence mechanism is handled over the IEEE 802.11 network utilizing the MSE in the IEEE 802.11 beacon. The description of the time sharing values is given in Clause 5. The format of the MSE beacon element is as shown in Table 3.

It is assumed that a device will be reset after setting any of these new parameters using the MAC sublayer management entity (MLME) primitives and before any of the settings of the new parameters are applied.

Table 3—Medium sharing element format

	Element ID	**Element length**	**Offset (T_{WLAN})**	**Length (T_{WPAN})**	**Guard (T_{GUARD})**
Octets:	1	1	2	2	2

The Offset, Length, and Guard fields are integer values specifying times in units of TU. Offset contains T_{WLAN}. Length contains T_{WPAN}. Guard contains T_{GUARD}.

5.2.1 MLME-AWMAPARAMETERS.request

This primitive sets the value of the AWMA timing parameters: WLANInterval (T_{WLAN}) and WPANInterval (T_{WPAN}).

5.2.2 Semantics of the service primitive

MLME-AWMAPARAMETERS.request

(

WLANInterval,

WPANInterval,

WGUARDInterval

)

The descriptions of these parameters are shown in Table 4.

Table 4—Description of parameters for MLME-AWMAPARAMETERS.request

Name	Type	Valid range	Description
WLANInterval	Integer	>0	The duration (in time units) of the WLAN interval
WPANInterval	Integer	>=0	The duration (in time units) of the WPAN interval
WGUARDInterval	Integer	>=0	The duration (in time units) of the WGUARD interval

The sum of WLANInterval and WPANInterval shall not be greater than the IEEE 802.11 beacon interval.

5.2.2.1 When generated

This primitive is generated by the station management entity to set the AWMA timing parameters.

5.2.2.2 Effect of receipt

This request sets the AWMA timing parameters (T_{WLAN}, T_{WPAN}, and T_{GUARD}) in the station upon receipt of this primitive.

5.2.3 MLME-AWMAPARAMETERS.confirm

This primitive confirms setting the AWMA timing parameters.

5.2.3.1 Semantics of the service primitive

```
MLME-AWMAPARAMETERS.confirm
        (
                ResultCode
        )
```

The description of this parameter is shown in Table 5.

Table 5—Description of the parameter for MLME-AWMAPARAMETERS.confirm

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, NOT_SUPPORTED	Indicates the result of the MLME-AWMAPARAMETERS.request

5.2.3.2 When generated

This primitive is generated by the MLME as a result of the MLME-AWMAPARAMETERS.request to set the AWMA timing parameters. It is not generated until the timing parameters have been set.

5.2.4 MLME-AWMAENABLE.request

This primitive either enables or disables the AWMA coexistence mechanism.

5.2.4.1 Semantics of the service primitive

MLME-AWMAENABLE.request

(

Enable

)

The description of this parameter is shown in Table 6.

Table 6—Description of the parameter for MLME-AWMAENABLE.request

Name	Type	Valid range	Description
Enable	Boolean	TRUE or FALSE	TRUE enables AWMA operation. FALSE disables AWMA operation.

5.2.4.2 When generated

This primitive is generated by the station management entity to enable (or disable) AWMA operation.

5.2.4.3 Effect of receipt

This request enables or disables AWMA operation in the station upon receipt of this primitive. The AWMA timing parameters are not effected.

5.2.5 MLME-AWMAENABLE.confirm

This primitive confirms enabling or disabling AWMA operation.

5.2.5.1 Semantics of the service primitive

MLME-AWMAENABLE.confirm

(

ResultCode

)

The description of this parameter is shown in Table 7.

Table 7—Description of the parameter for MLME-AWMAENABLE.confirm

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, FAILURE, NOT_SUPPORTED	Indicates the result of MLME-AWMAENABLE.request

5.2.5.2 When generated

This primitive is generated by the MLME as a result of the MLME-AWMAENABLE.request to enable or disable AWMA operation. It is not generated until AWMA operation has been either enabled or disabled. A FAILURE is sent if the AWMA timing parameters have not previously been set by a MLME-AWMA-PARAMETERS.request.

5.2.6 Additional management information base definition

To support AWMA the IEEE 802.11 management information base (MIB) needs to be augmented with the following station management attributes.

5.2.6.1 agAWMAgrp

WLANInterval,

WPANInterval,

WGUARDInterval,

Enabled;

5.2.6.2 Station management attribute group templates

AWMAgrp ATTRIBUTE GROUP

GROUP ELEMENTS

WLANInterval,

WPANInterval,

WGUARDInterval,

Enabled;

REGISTERED AS FOLLOWS:

{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attributeGroup(8) AWMAgrp(1) };

5.2.6.3 WLANInterval

WLANInterval ATTRIBUTE

BEHAVIOR DEFINED AS FOLLOWS:

This attribute is the duration of the WLAN interval (in time units) used in AWMA.

REGISTERED AS FOLLOWS:

{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) StationID(1) };

5.2.6.4 WPANInterval

WPANInterval ATTRIBUTE

BEHAVIOR DEFINED AS FOLLOWS:

This attribute is the duration of the WPAN interval (in time units) used in AWMA.

REGISTERED AS FOLLOWS:

{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) StationID(1) };

5.2.6.5 WGUARDInterval

WGUARDInterval ATTRIBUTE

BEHAVIOR DEFINED AS FOLLOWS:

This attribute is the duration of the WGUARD interval (in time units) used in AWMA.

REGISTERED AS FOLLOWS:

{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) StationID(1) };

5.2.6.6 Enabled

Enabled ATTRIBUTE

BEHAVIOR DEFINED AS FOLLOWS:

This attribute indicates whether AWMA is enabled (true) or disabled (false).

REGISTERED AS FOLLOWS:

{ iso(1) member-body(2) us(840) ieee802dot11(10036) SMT(1) attribute(7) StationID(1) };

5.2.7 Frame formats

This subclause contains the modifications (i.e., additions) required in IEEE Std 802.11, 1999 Edition to accommodate these changes for coexistence.

5.2.7.1 Beacon frame format

Add a row and the accompanying note to Table 5 of IEEE Std 802.11b-1999 that is shown here as Table 8 and Note 6.

Table 8—Beacon frame body

Order	Information	Note
49	Media Sharing	6

Note 6—The Media Sharing information element is only present within Beacon frames generated by APs supporting Media Sharing.

5.2.7.2 Probe response frame format

Add a row and accompanying note to Table 12 of IEEE Std 802.11b-1999 that is shown here as Table 9 and Note 11.

Table 9—Probe response frame body

Order	Information	Note
49	Media sharing	11

Note 11—The media sharing information element is only present within Probe response frames generated by APs supporting media sharing.

5.2.7.3 Information elements

Add the following element, Media Sharing, with the value 49, assigned by the Naming Authority, and modify the Reserved value range accordingly in Table 20 (Element IDs) of IEEE Std 802.11b-1999.

5.3 Restriction on WLAN and WPAN transmissions

If AWMA is enabled on a device, then it is required that all WLAN transmissions are restricted to occur during the WLAN subinterval. Similarly, all WPAN transmissions are restricted to the WPAN subinterval. The WLAN mobile units and the WLAN AP all share a common TBTT, so along with shared knowledge of the value of T_{GUARD} and T_{WLAN}, all AWMA enabled WLAN devices shall restrict their transmissions to be within the common WLAN subinterval.

The WPAN device collocated with the WLAN node shall be a WPAN master device. In particular, if the WPAN device conforms to IEEE 802.15.1, then all ACL data transmissions are controlled by the WPAN master. In particular, WPAN slaves may only transmit ACL packets if in the previous time slot the WPAN slave received an ACL packet. Therefore, the WPAN master shall end transmission long enough before the end of the WPAN subinterval so that the longest slave packet allowed (e.g., a five-slot IEEE 802.15.1 packet) will complete its transmission prior to the end of the WPAN interval. Figure 4 illustrates the timing requirement. The value of T_M shall be large enough so as to ensure that the value of T_S is greater than zero.

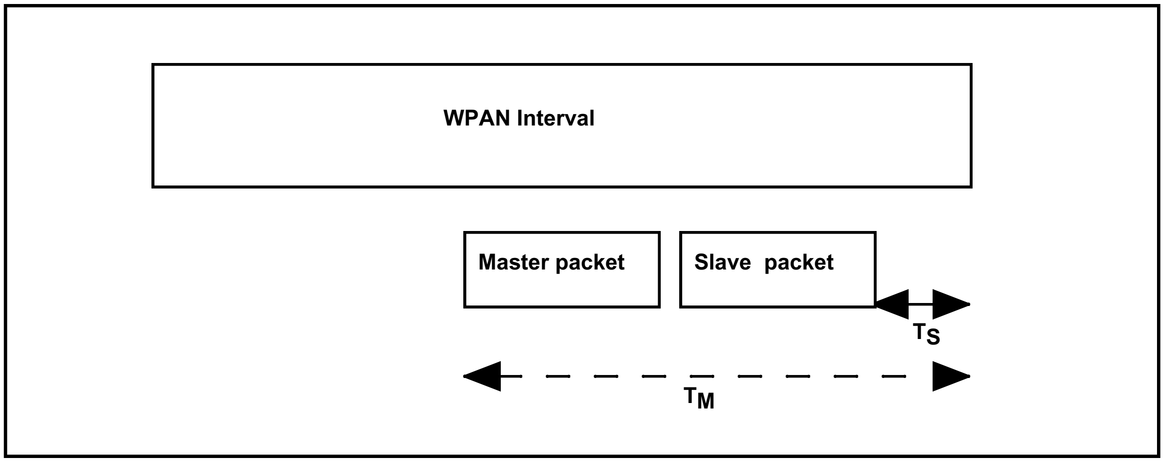

Figure 4—Timing of WPAN packets

IEEE 802.15.1 supports SCO packets, for voice traffic. These packets occur on a regular basis with a fixed period. There are several SCO packet types, depending on the level of FEC. For example, an HV3 link repeats every 6 slots. The first two slots are used for SCO packets and the last four packets may either be used for ACL packets or remain unused time slots. In IEEE 802.15.1 a time slot is 0.625 µs and the SCO HV3 period is 3.75 µs. This is a small fraction of the typical WLAN beacon period. As a result if the WLAN beacon period is subdivided into two subintervals, the WPAN SCO packets may not be restricted to the WPAN interval. As a result the AWMA coexistence mechanism does not support IEEE 802.15.1 SCO links.

The WLAN shall also restrict all WLAN transmission to the WLAN subinterval. Figure 5 illustrates the timing of WLAN traffic. Before a WLAN device may transmit a packet it shall ensure that the value T_S is greater than zero. The WLAN shall calculate T_S as follows:

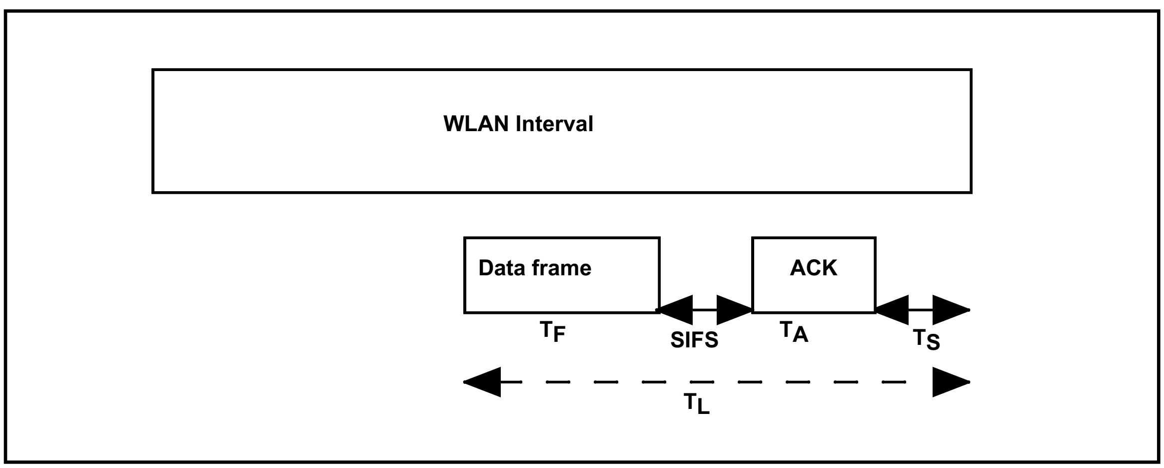

Figure 5—Timing of WLAN packets

$$T_S = T_L - T_F - \mathrm{SIFS} - T_A \qquad (1)$$

where

T_L is the time until the end of the WLAN interval,
T_F is the length of the frame to be sent,

SIFS is the short interframe space (SIFS) in the WLAN, and
T_A is the length of the acknowledgement packet (ACK).

As long as T_S is greater than zero, the WLAN may send the frame. If not, it shall defer transmission until the next WLAN subinterval.

6. Packet traffic arbitration

The PTA control entity provides per-packet authorization of all transmissions. In the PTA mechanism, the IEEE 802.11b station (STA) and IEEE 802.15.1 node are collocated. Each attempt to transmit by either the IEEE 802.11b or the IEEE 802.15.1 is submitted to PTA for approval. PTA may deny a transmit request that would result in collision. The PTA mechanism may support IEEE 802.15.1 SCO links.

The PTA mechanism coordinates sharing of the medium dynamically, based on the traffic load of the two wireless networks.

PTA uses its knowledge of the duration of IEEE 802.11b activity and future IEEE 802.15.1 activity of a number of slots into the future to predict collisions. When a collision would occur, PTA prioritizes transmissions based on simple rules that depend on the priorities of the various packets.

It is recommended to use PTA whenever there is a high variability in the WLAN and WPAN traffic load or whenever an IEEE 802.15.1 SCO link needs to be supported. The PTA mechanism uses a dynamic packet scheduling mechanism that automatically adapts to changes in traffic loads over the WLAN and WPAN networks. The PTA mechanism supports IEEE 802.15.1 SCO links while the AWMA mechanism does not.

Annex J contains information on the performance results for PTA and IEEE 802.11b.

6.1 Known physical layer characteristics

The IEEE 802.11b PHY layer operates on a known frequency-static channel. The IEEE 802.15.1 PHY layer hops following a known hopping pattern. At any time, the IEEE 802.15.1 signal may be within or outside the passband of the IEEE 802.11b PHY layer. These are the in-band and out-of-band cases, and they effect the probability of a collision.

The different collision cases are summarized in Table 10.

Table 10—Collision cases as a function of local activities

Local 802.11b activity	Local 802.15.1 activity			
	Transmit		Receive	
	In-band	Out-of-band	In-band	Out-of-band
Transmit	Transmit	None	Transmit-Receive or None	Transmit-Receive or None
Receive	Transmit-Receive or None	Transmit-Receive or None	Receive	None

The different collision types are defined in Table 11.

Table 11—Definition of collision types

Collision type	Definition
Transmit	Both radios are transmitting in-band. One or both of the packets might be received with errors.
Receive	Both radios are receiving in-band. One or both of the packets might be received with errors.
Transmit-Receive	One radio is transmitting and the other is receiving. The locally received packet is received with errors.
None	Simultaneous activity of the two radios does not increase the PER.

In the case of “Transmit-Receive or None” collisions, whether there is a collision or not depends on a number of PHY layer-related parameters that may include: transmit power, received signal strength and the difference between IEEE 802.11b and IEEE 802.15.1 center frequencies.

An implementation predicts the difference between these collision outcomes based on its knowledge of the operating parameters of its PHY layer. So, based on PHY-layer parameters, an implementation predicts whether a collision occurs. The algorithm for predicting packet collisions is outside the scope of this recommended practice.

Implementation constraints may also introduce additional types of “collisions” based on simultaneous conflicting demands for hardware resources. For example, a single-antenna system is unlikely to be able to transmit and receive simultaneously.

6.2 PTA structure

Figure 6 shows the structure of the PTA control entity. Each device has a corresponding control entity to which it submits its transmit requests. This control entity allows or denies the request based on the known state of both radios.

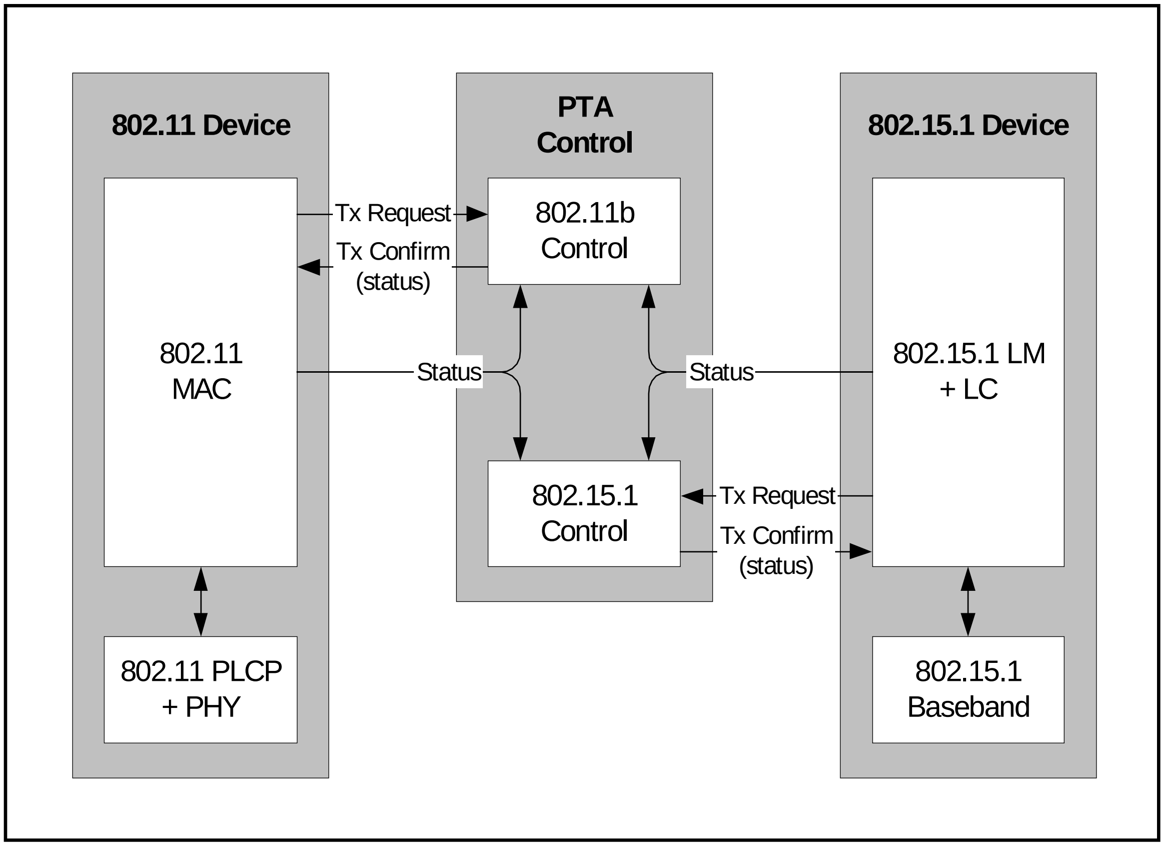

Figure 6—Structure of the PTA entity

6.3 Known 802.11b state

The PTA control assumes that the state defined in Table 12 is available from the IEEE 802.11b MAC.

Table 12—Known 802.11b state

802.11b state item	Definition
Current802.11bState	Indicates the current activity of the 802.11b MAC in terms of current or expected receive and transmit activity. The decision logic described in 6.6 requires that the state variable indicate if 802.11b radio is idle, transmitting, or receiving. Additional states may be exposed through this interface to support local priority policy as described in 6.7.
Channel	Channel number
End Time	Time of the end of the current activity. This may be based on the last duration value received or transmitted in a MAC protocol data unit (MPDU) header.

When a transmit request is made from the IEEE 802.11b MAC, the information described in Table 13 is known.

Table 13—802.11b TX request state

802.11b TX request parameter	Description
Packet type	Type of the MPDU
Duration	On-air duration of the MPDU

6.4 Known 802.15.1 state

The PTA control assumes that the state described in Table 14 is available from the IEEE 802.15.1 MAC.

Table 14—Known 802.15.1 state

802.15.1 state item	Description
Current802.15.1 state	Describes the current activity of the 802.15.1 baseband in terms of current or expected receive and transmit activity. The decision logic described in 6.6 requires that the state variable indicate if 802.15.1 stack is idle, transmitting, or receiving.
Channel list	List of channels for the current and future slots.
Packet type	Indicates the type of packet predicted for the current and future slots.
Duration	On-air duration of the current packet.
Slot end time	Time at the end of the current slot (i.e., at the next slot edge).

6.5 802.11b control

The purpose of the IEEE 802.11b control entity is to allow or deny transmit requests from the IEEE 802.11b MAC. The TX Request signal is sent when the IEEE 802.11b MAC has determined that it may transmit according to its own protocol (i.e., after any required backoff has completed).

On receipt of a TX Request signal, the IEEE 802.11b control immediately generates a TX Confirm signal containing a status value that is either allowed or denied. Figure 7 defines how the status value is selected.

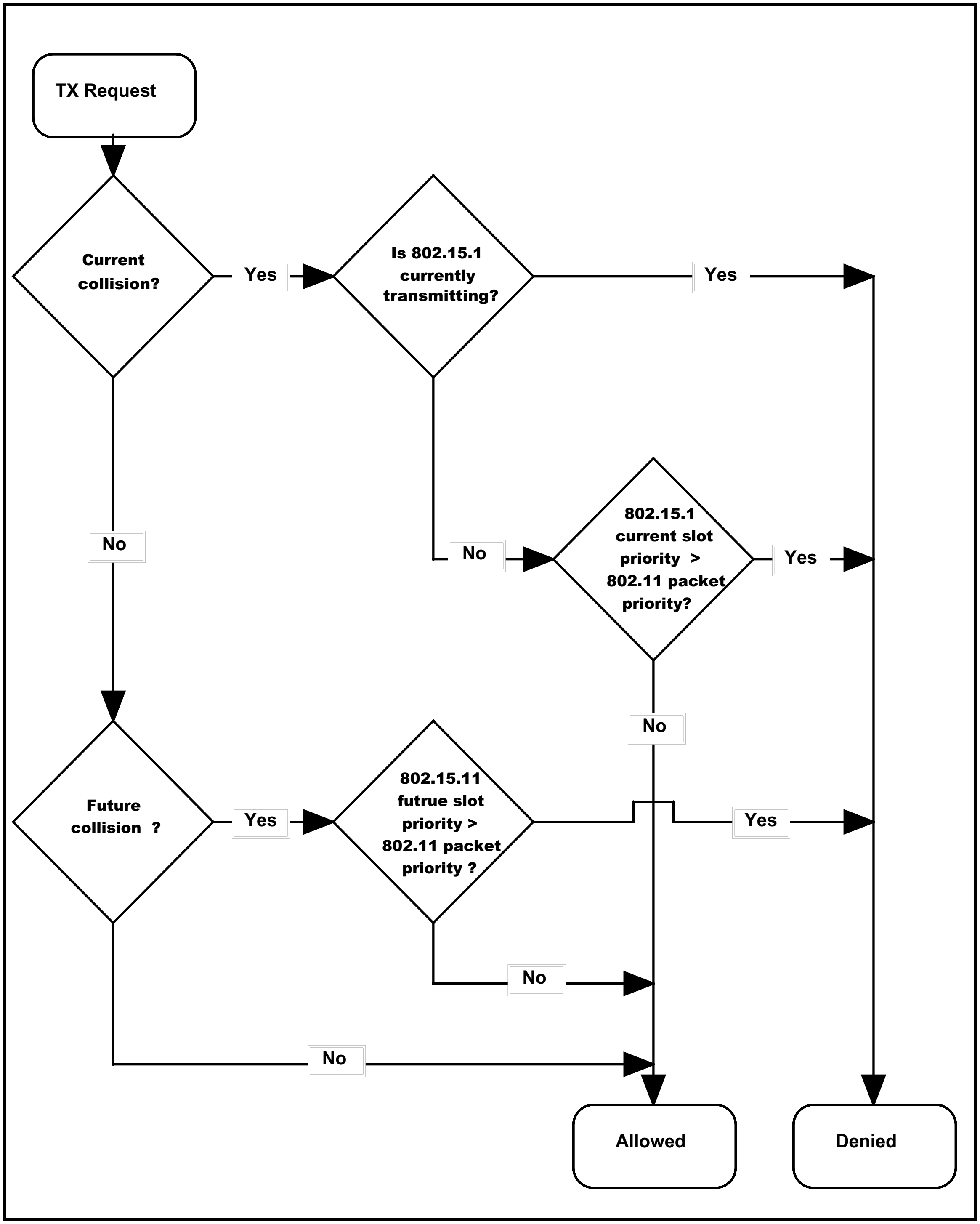

Figure 7—Decision logic for 802.11b TX request

The effect of a denied result on the IEEE 802.11b MAC protocol depends on the access mechanism currently in use. This is defined in Table 15.

Table 15—Effect of denied status on the 802.11b MAC

Access mechanism	Effect of TX confirm (status=denied)
DCF	The denied result appears to be a transient carrier-sense condition that requires a distributed (coordination function) interframe space (DIFS) time to expire before a subsequent transmit request may be made. The denied result has no effect on the contention window (CW) or retry variables because no transmission has occurred.
PCF(as CF-pollable STA)	No transmission from the STA occurs. The PC is unaware of the reason for the loss of an expected MPDU, and it will respond in an implementation-specific fashion.
PCF as PC	No transmission from the PC occurs. The PC may attempt a transmission after an additional SIFS. There is no requirement that it sense the medium prior to this transmission. Alternatively, the PC may perform a backoff. In either case, the NAV setting of STAs should prevent them from attempting to transmit during this time.
Note	PCF is only included for completeness in this table. PCF is not covered by this recommended practice.

Table 16 defines the conditions examined by the decision logic.

Table 16—Conditions examined by 802.11b TX request decision logic

Condition	Definition
Current collision	There is a transmit or transmit-receive collision between the current 802.15.1 activity and the 802.11b transmit request.
Future collision	There is a transmit or transmit-receive collision between the 802.15.1 activity scheduled for a future slot and the current 802.11b TX Request. For a collision to occur in a slot, the requested 802.11b transmit activity shall continue until at least the start of that slot.
802.15.1 current slot priority >802.11b packet priority	Does the priority of the current 802.15.1 activity have greater priority than the requested 802.11b packet? (See 6.7)
802.15.1 future slot priority >802.11b packet priority	Does the priority of the future colliding 802.15.1 activity have greater priority than the requested 802.11b packet? (See 6.7)
Is 802.15.1 currently transmitting?	The current 802.15.1 state is in a transmitting state.

6.6 802.15.1 control

In response to a TX Request signal, the IEEE 802.15.1 control immediately generates a TX Confirm signal containing a status value that is either allowed or denied. Figure 8 defines how the status value is selected.

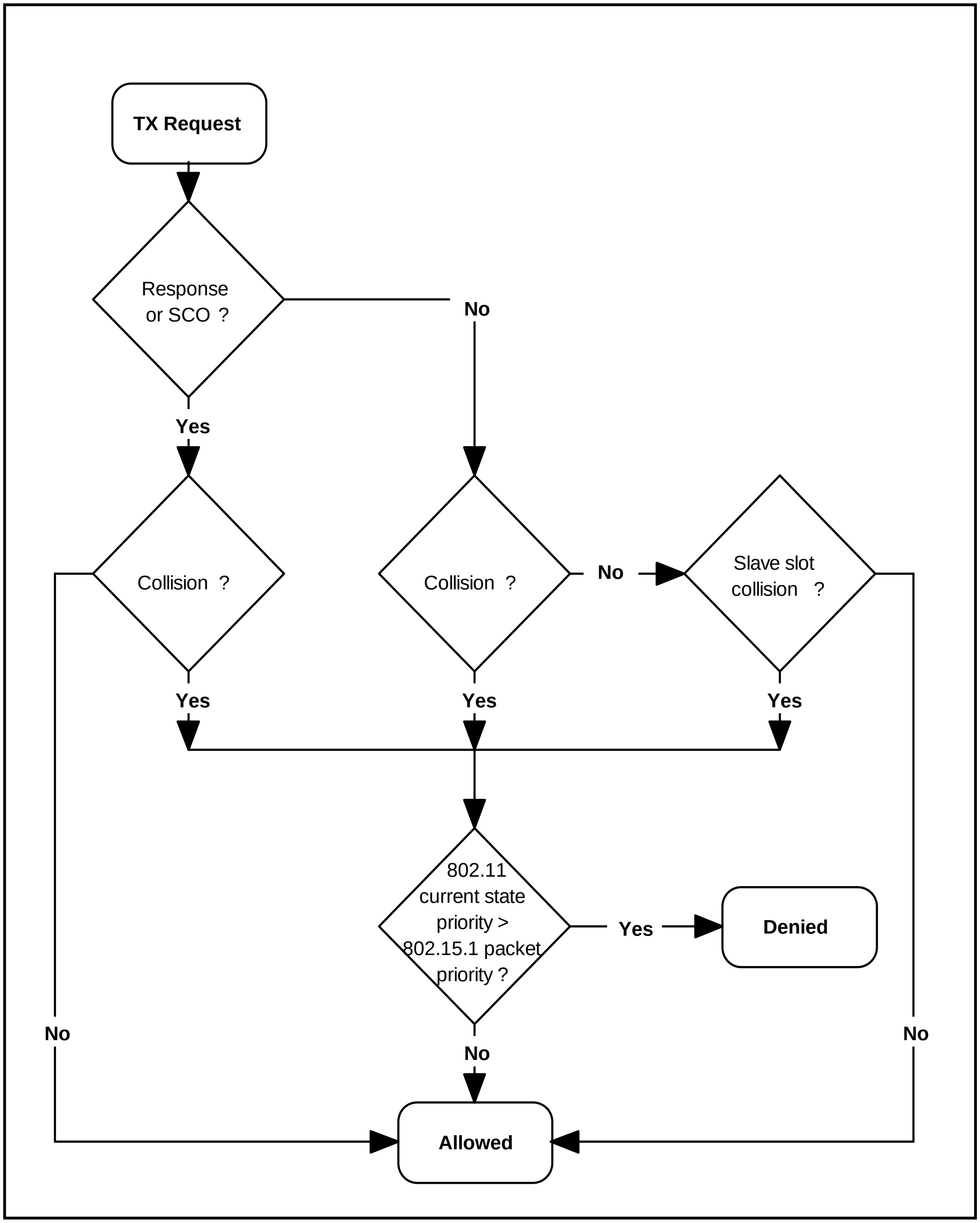

Figure 8—Design logic for 802.15.1 TX request

The effect of the denied result on the IEEE 802.15.1 stack is to prevent IEEE 802.15.1 transmission during the whole slot [or slot half in the case of scan (paging and inquiry) sequences].

Table 17 defines the conditions examined in the execution of this decision logic.

Table 17—Conditions examined by 802.15.1 TX request decision logic

Condition	Definition
Response or SCO?	True if the TX Request packet type is slave ACL, ID, FHS, or SCO.
Collision?	Does a transmit or transmit-receive collision occur between the 802.15.1 transmit request and the current state of the 802.11b stack?
Slave slot collision?	Does a transmit-receive collision occur between the slave response to the 802.15.1 transmit request and the current state of the 802.11b stack?
Current 802.11b state priority >802.15.1 packet priority?	Is the priority of the 802.11b current state greater than the 802.15.1 TX Request packet priority? (See 6.7)

6.7 Priority comparisons

The decision logic that allows or denies a packet transmit request uses a priority comparison between the state of the requested transmit packet and the known state of the other protocol stack.

An implementation defines priority values for each separate state value exposed by its protocol stack, and for each transmit packet type.

6.8 Recommended priority comparisons

Implementers of this recommended practice may choose various ways of assigning priorities to packets according to their applications. Subclauses 6.8.1 and 6.8.2 describe two possible implementations: fixed and randomized priorities.

6.8.1 Fixed priority

In this priority assignment, an IEEE 802.15.1 SCO packet should have a higher priority than IEEE 802.11b DATA MPDUs and an IEEE 802.11b ACK MPDU should have a higher priority than all IEEE 802.15.1 packets.

6.8.2 Randomized priority

The priorities of the packets may be assigned based on a randomized mechanism. A random variable, r, uniformly distributed between [0,1] along with a threshold, T ($0 <= T < 1$) are used. If the incoming packet is from an IEEE 802.11b device, a priority of 2 is assigned to it if the random number, r, is smaller than T. Otherwise, a priority of 0 is assigned. If the incoming packet is from an IEEE 802.15.1 device, a priority of 1 is assigned.

6.9 Maintaining quality of service

A device may optionally monitor quality of service (QoS) by defining metrics (such as PER and delay) per protocol stack. It may use these metrics to bias its priorities in order to meet locally-defined fairness criteria.

NOTE—An implementation may need additional communication not shown here to decide whether to admit a connection-setup with particular QoS requirements, given knowledge of QoS commitments in the other protocol stack.

An implementation may attempt to maintain SCO QoS so as not to exceed some level of SCO PER. It does this by monitoring the SCO PER and comparing with a threshold. The priority of the SCO packet is increased when the SCO PER is above the threshold.

7. Deterministic interference suppression

In this clause, an interference suppression mechanism, denoted deterministic interference suppression, designed to mitigate the effect of IEEE 802.15.1 interference on IEEE 802.11b, is discussed. On account of the IEEE 802.15.1 signal having a bandwidth of approximately 1 MHz, it may be considered a narrowband interferer for the 22 MHz wide IEEE 802.11b signal. The basic idea of the interference suppression mechanism is to put a null in the IEEE 802.11b's receiver at the frequency of the IEEE 802.15.1 signal. However, because IEEE 802.15.1 is hopping to a new frequency for each packet transmission, the IEEE 802.11b receiver needs to know the FH pattern, as well as the timing, of the IEEE 802.15.1 transmitter. This knowledge is obtained by employing an IEEE 802.15.1 receiver as part of the IEEE 802.11b receiver. Thus, this is a collocated, collaborative method. On account of this being primarily a physical layer solution, it may be integrated with the PTA MAC sublayer solution. This clause discusses the procedure, which is applicable to all basic rate sets in IEEE 802.11b (1, 2, 5.5, and 11 Mbit/s).

Figure 9 and Figure 10 show the block diagrams of the 1 Mbit/s IEEE 802.11b transmitter and receiver, respectively. Note that in Figure 10 between the chip matched filter and the pseudorandom noise (PN code sequence) PN correlators is an adjustable transversal filter. The optimal coefficients of this filter are estimated and then used to update the filter. Figure 11 shows the structure of the transversal filter.

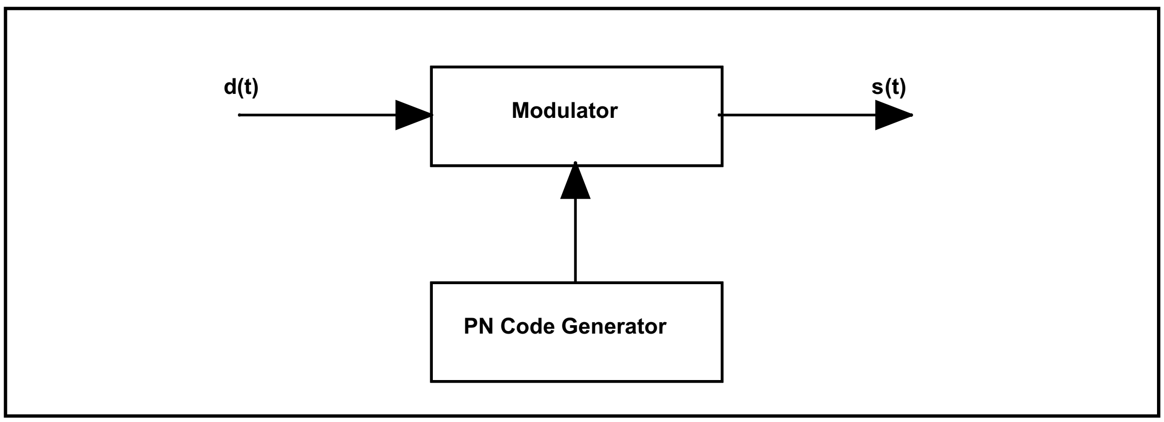

Figure 9—Block diagram of the 1 Mbit/s IEEE 802.11 system, employing frequency nulling for the transmitter

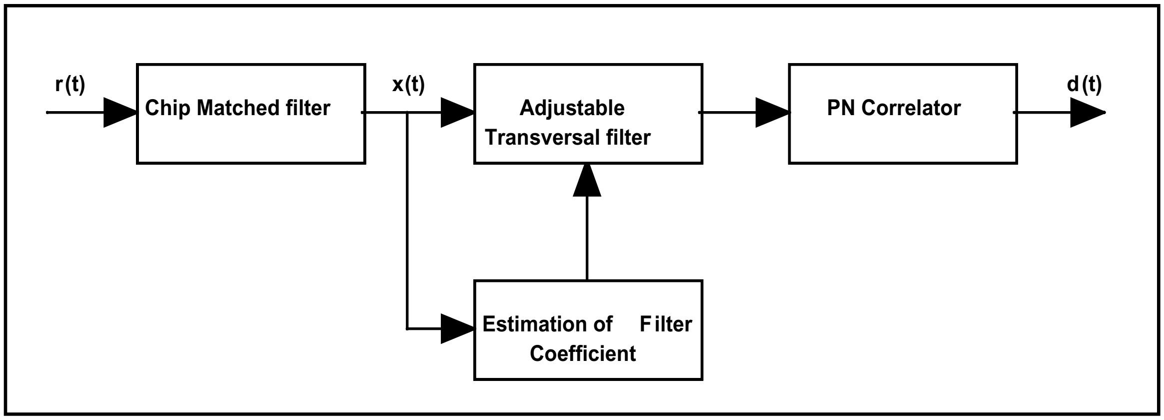

Figure 10—Block diagram of the 1 Mbit/s IEEE 802.11 system, employing frequency nulling for the receiver

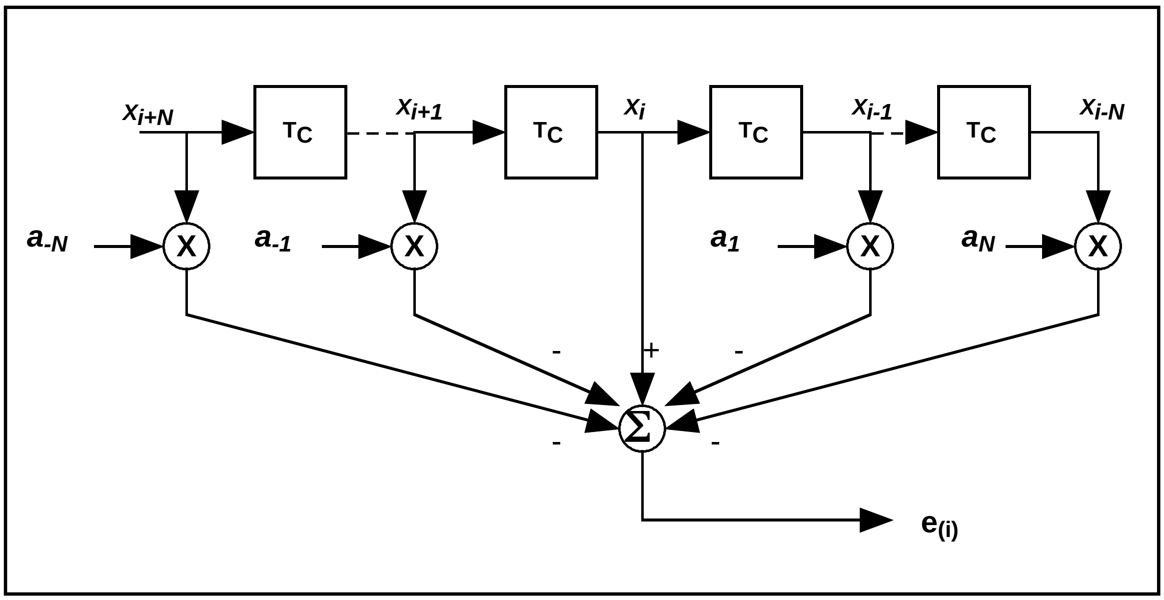

Figure 11—Adjustable transversal filter used in the 802.11 receiver

For the calculation of the adaptive tap weights let us first assume that the interferer is a pure tone. Consider the central tap in the transversal filter. At time iT_c, it may be written as (see [B6])

$$x_i = d_i + V_j \cos(2\pi f_j i T_c + \theta) + n_i$$

where

T_c	is the sampling interval equal to the chip time,
d_i	is the signal amplitude,
V_j	is the amplitude of the interferer,
f_j	is the frequency of the interferer,
n_i	is the random noise, and
θ	is a random phase angle with a uniform distribution.

The objective is to find the tap weights that minimize the error.

When the interference is stationary, one may employ the Wiener solution to find the optimum tap weights. These optimal tap coefficients, $a_{k,opt}$ are found by solving the following system of equations (see Ketchum[B4] and Milstein [B6]):

$$\sum_{\substack{k = -N \\ k \neq 0}}^{N} r(L-k)a_{k,opt} = r(L)$$

L= –N ... −1, 1 ... N

where the auto correlation function is given by

$$r(m) = E\{x_i x_{i-m}^*\}$$

and the samples, x_i, are as shown in Figure 11. There are a total of 2N taps.

The first assumption is that the PN sequence is sufficiently long. This implies that the PN signal samples at the different taps are not correlated. In this case, the solutions for the optimal tap weights have the simple form (see [B6])

$$a_{k,opt} = Ae^{jk\Omega T_c}$$

where

$\Omega=2\pi f_i$, and the parameter A is given by

$$A = \frac{I}{(S+\sigma_n^2)+2IN} \tag{2}$$

Equation (2) shows that one needs estimates of the signal power, S, the interferer power, I, and the noise power, $\sigma_n{}^2$. In many traditional military jamming scenarios, the signal to noise ratio (SNR) may be relatively low (see [B15]). Fortunately, for IEEE 802.11b systems in typical configurations, the SNR is often quite high. Therefore, the noise power in this equation is ignored. Still, one needs an estimate of the signal to interference ratio (SIR) to determine the optimal tap coefficients. The SIR value is fixed and equal to -20 dB; this is a typical value. Using this assumption, it is no longer necessary to estimate the SIR. One still needs an estimate of the offset in frequency, Ω, between the IEEE 802.11b signal and the interferer. In a collaborative system, this frequency offset is assumed to be known a priori, on account of it being provided by the IEEE 802.15.1 receiver.

Annex K contains information on the simulation results for deterministic interference suppression.

8. Adaptive interference suppression

The collaborative interference suppression method previously described in Clause 7 requires an IEEE 802.15.1 receiver collocated with the WLAN receiver. In this clause, a non-collaborative approach, based solely on signal processing in the physical layer of the WLAN, is recommended in order to cancel the IEEE 802.15.1 interference. In this method, the WLAN has no a priori knowledge of the timing or frequency used by the IEEE 802.15.1 system, and it uses an adaptive filter to estimate and cancel the interfering signal.

The block diagram of the adaptive interference suppression system is shown in Figure 12 (see [B5]). First of all, the received signal, *x(n)*, is delayed and passed through the adaptive filter, which exploits the uncorrelated nature of the wideband IEEE 802.11 signal to predict the unwanted narrowband IEEE 802.15.1 signal, *y(n)*. This estimate is subtracted from the received signal to generate the prediction error signal, *e(n)*, which is an approximation of the IEEE 802.11 signal. The prediction error signal is also used to adapt the filter as discussed below. It should be noted that the adaptive notch filter is operating as a whitening filter at complex baseband, and so one can use it as a “front-end” to a number of different receivers, chosen depending on the channel and given performance criteria. In this clause, the receiver has the same architecture as shown in Annex C in Figure C.6 and Figure C.8, for 1 Mbit/s and 11 Mbit/s rates, respectively. Additional discussion and more performance results are given in Soltanian, et al. [B21]. For multi-path channels, an alternative receiver can be designed based on the generalized likelihood-ratio test (GLRT) receiver of Iltis [B3].

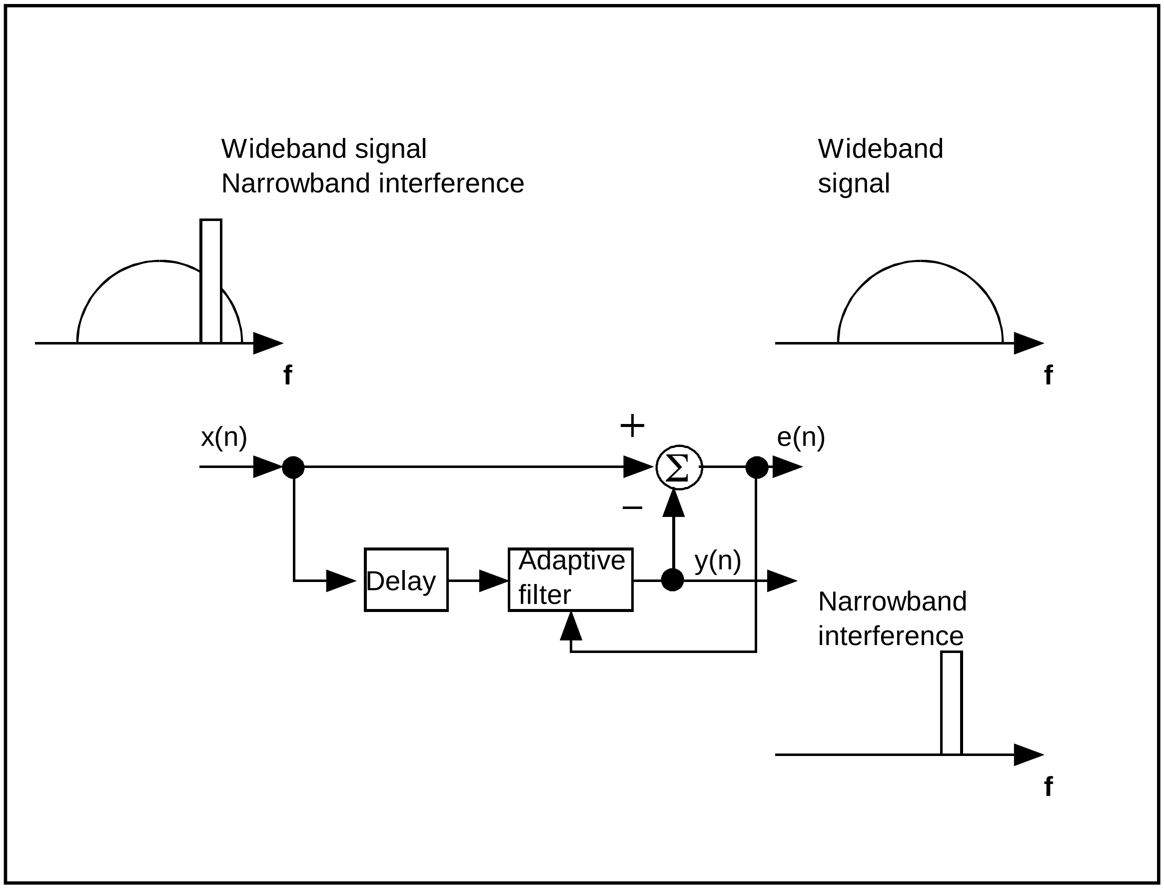

Figure 12—An adaptive notch filter or whitening filter

To reduce the amount of computation while providing numerical stability, the adaptive filter is implemented as a recursive least-squares lattice (RLSL) filter, which is shown in Figure 13. Note that the delays are explicitly incorporated into this structure, so it corresponds to both the delay and adaptive filter blocks in Figure 12. The two main parameters of the filter are $M = 3$, the order of the lattice, and λ, the forgetting factor. λ represents the memory of the algorithm, with $\lambda = 1$ corresponding to infinite memory[10]. For a time-varying interferer, λ should be chosen to be less than one, so that past data has an exponentially decreasing effect on the estimation. During the course of the simulation, it was found that $\lambda = 0.97$ gives the best results. As shown in Figure 13, the forward and backward reflection coefficients, $k_{f,i}$ and $k_{b,i}$, need to be updated for

[10]Infinite memory refers to the length of the impulse response, as in an infinite impulse response (IIR) filter. It does not mean that an infinite number of storage locations are used. In fact, only a very small number are needed, as shown in the figure.

each stage in the filter; the complete set of equations for updating the RLSL filter, including initialization, is given in Table 15.4 in Haykin [B11]. The forward prediction, $f_3(n)$ is then used as $y(n)$ in Figure 12.

Annex L contains information on the simulation results for adaptive interference suppression.

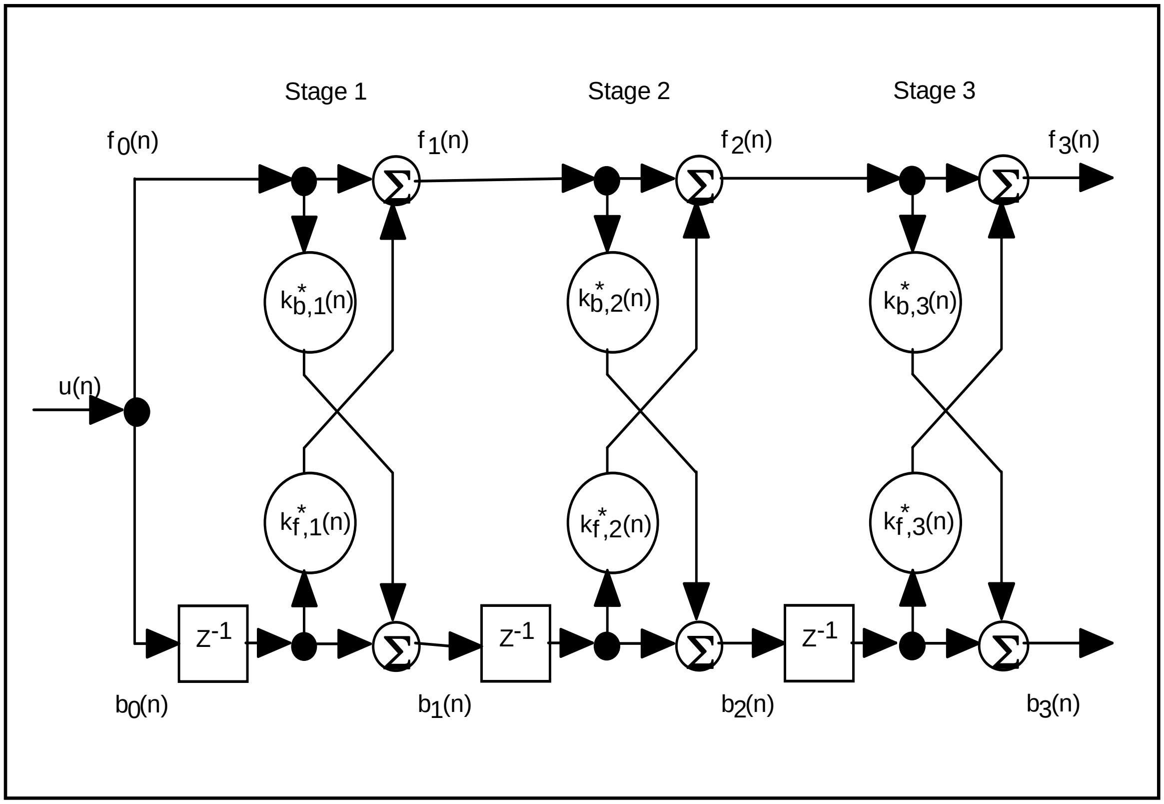

Figure 13—Three stage lattice filter

9. Adaptive packet selection

IEEE 802.15.1 specifies a variety of packet types with different combinations of payload length, slots occupied, FEC codes, and ARQ options. The motivation is to provide the necessary flexibility for the implementers and applications so that the packets may be chosen optimized for the traffic and channels presented. In this clause, a mechanism is described to take advantages of these different packet types for improving network capacity for coexistence scenarios.

9.1 IEEE 802.15.1 packet types for SCO and ACL

IEEE 802.15.1 provides 4 types of packets (i.e., HV1, HV2, HV3, and DV) that may be sent over a SCO link. Table 18 summarizes the different configurations for these packet types.

Table 18—IEEE 802.15.1 SCO packet types

Packet types	HV1	HV2	HV3	DV
Payload header (bytes)	None	None	None	1 D
Payload length (bytes)	10	20	30	20
Channel utilization (%)	100	50	33	100
FEC Code	1/3	2/3	None	2/3 D
NOTE—D refers to the data portion only				

The packets differ mostly in the FEC code used and the amount of channel occupied by the SCO link. Choice of different packet types provides intriguing trade-offs of error protection at the bit level and the amount of interference generated (or the bandwidth available for other links).

The ACL link, in addition to the use of different FEC protections, adds the choice of multi-slot packets. Table 19 summarizes the packets for ACL link.

Table 19—IEEE 802.15.1 ACL packet types

Packet types	DH1	DM1	DH3	DM3	DH5	DM5
Slot time	1	1	3	3	5	5
Packet header (bytes)	1	1	2	2	2	2
Payload length (bytes)	0-27	0-17	0-183	0-121	0-339	0-224
FEC Code	None	2/3	None	2/3	None	2/3

The different ACL packet types allow the applications to make trade-offs among different considerations of traffic flow, channel conditions of the current hop, duty cycles, and interference generated to neighboring networks.

9.2 Methods of adaptive packet selection

The basic idea is to dynamically select packet types, given either an ACL or SCO link, such that maximal total network capacity is achieved. This implies not only optimizing throughput for the IEEE 802.15.1 piconet but also reducing interference to the coexisting IEEE 802.11b network, which will increase the throughput of the IEEE 802.11b network.

For SCO links, when the network performance is range limited, (i.e., the stations are separated by a distance such that only small noise margin is maintained), random bit errors are the dominant problem for dropping packets. Choosing a packet type that uses more error protection will increase the performance of the SCO link. Therefore, for range limited applications, the HV1 packet is preferred over the HV2 packet, and the HV2 packet is preferred over the HV3 packet. By monitoring the RSSI and SNR of the IEEE 802.15.1 radio, the IEEE 802.15.1 may determine if the choice of more error protection is beneficial.

For SCO links in the coexistence scenarios, usually the dominant reason for packet drop is not due to noise or range, but rather is due to the strong interference produced by the collocated network such as an IEEE

802.11b network. In this case, increasing FEC protection will cause IEEE 802.15.1 device to generate more packets (HV1 packets occupy the channel 3 times more often than HV3 packets), and thus a lot more interference to the IEEE 802.11b network. As shown by the simulation results in Figure 14 and Figure 15, the total network throughput is severely degraded. Figure 14 illustrates the performance of the IEEE 802.11b network before and after the initiation (at 15 seconds) of an HV1 SCO link by the IEEE 802.15.1 piconet. The IEEE 802.11b throughput dropped from 5.8 Mbit/s to be significantly less than 1 Mbit/s. Figure 15 represents the results as seen from IEEE 802.15.1, where the sending of HV1 packets started at 15 seconds. Therefore, in interference-limited scenarios (as in IEEE 802.15.1 and IEEE 802.11b coexistence scenarios), HV3 packet is preferred over HV2 packet, and HV2 packet is preferred over HV1 packet.

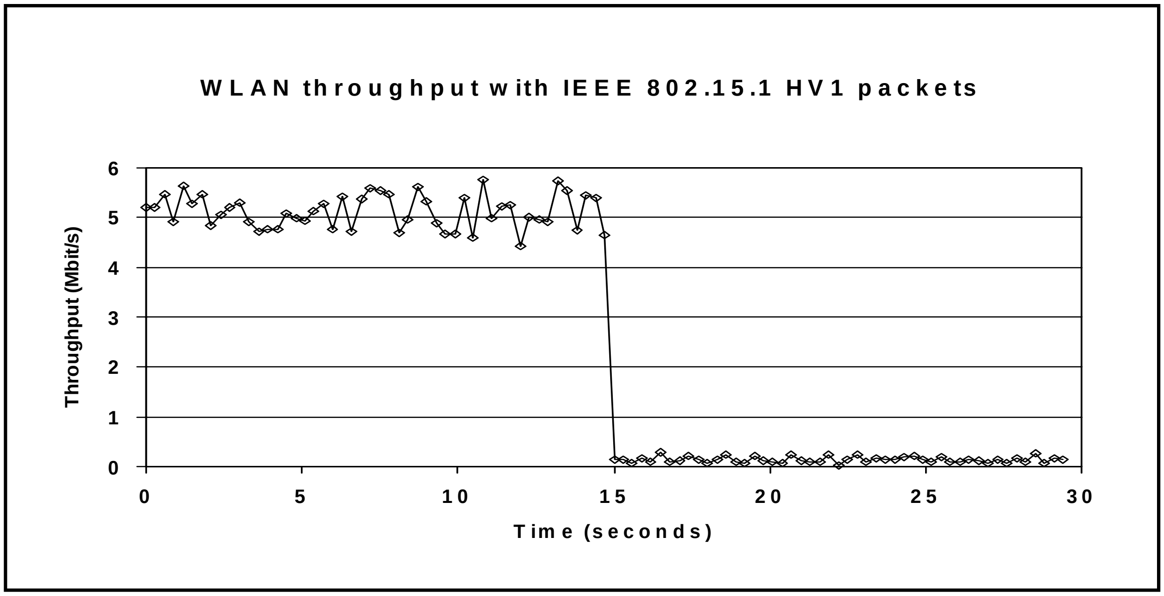

Figure 14—Impact of IEEE 802.15.1 HV1 packets on the performance of an 802.11b network

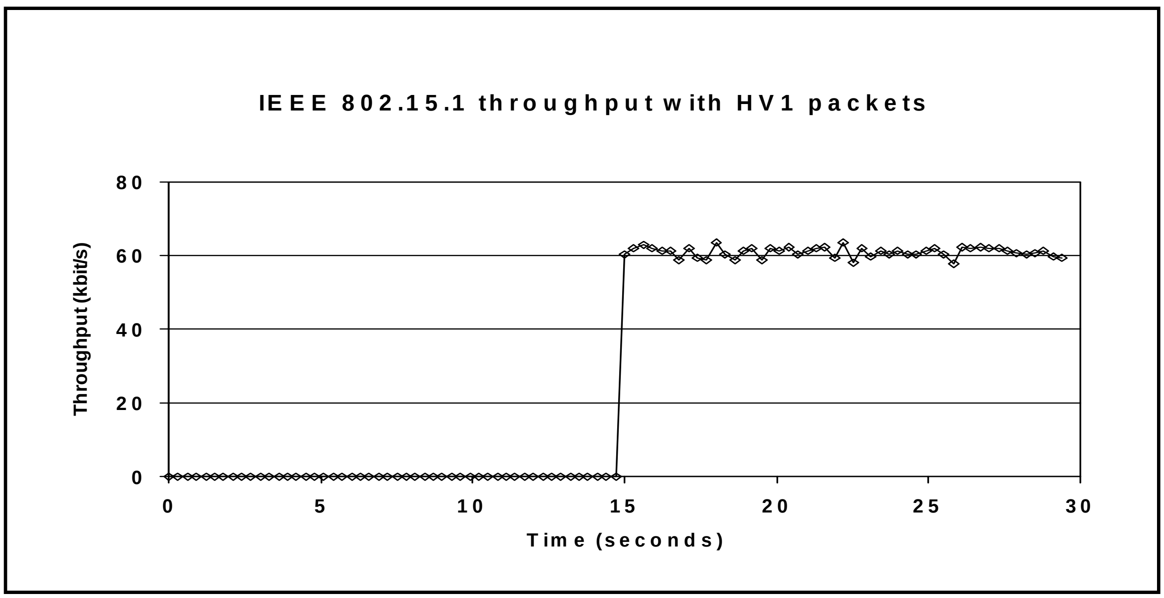

Figure 15—IEEE 802.15.1 throughput with HV1 packets in the presence of a IEEE 802.11b network

For similar reasons, the same guidelines apply to the selection of ACL packets. When the IEEE 802.15.1 network performance is range-limited, ACL packets with FEC protections, which include DM1, DM3 and DM5, should be used. On the other hand, when the system is interference limited, the 802.15.1 device should reduce the number of bits transmitted by choosing a more bandwidth efficient packet format such as DH1, DH3 or DH5.

10. Packet scheduling for ACL links

In this clause, a scheduling mechanism for IEEE 802.15.1 that alleviates the effect of interference with IEEE 802.11 DSSS systems is described. This scheduling mechanism consists of two components. These two components are channel classification and master delay policy.

Channel classification is performed on every IEEE 802.15.1 receiver and is based on measurements conducted per frequency or channel in order to determine the presence of interference. A frequency, *f*, is “good” if a device can correctly decode a packet received on it. Otherwise, *f* is “bad.” A number of criteria can be used in determining whether *f* is “good” or “bad,” such as RSSI, PER measurements, or negative ACKs. Clause 11 gives additional details on each classification criterion.

A channel classification table capturing the frequency status (“good”/”bad”) for each device in the piconet is kept at the master. Depending on the classification method used, an explicit message exchange between the master and the slave device may be required. Implicit methods such as negative ACKs do not require the slave to send any communication messages to the master concerning its channel classification.

The master delay policy makes use of the information available in the channel classification table in order to avoid packet transmission in a “bad” receiving channel. On account of the IEEE 802.15.1 master device controling all transmissions in the piconet, the delay rule has to only be implemented in the master device. Furthermore, following each master transmission there is a slave transmission. Therefore, the master checks both the slave's receiving frequency and its own receiving frequency before choosing to transmit a packet in a given frequency hop as illustrated in Figure 16.

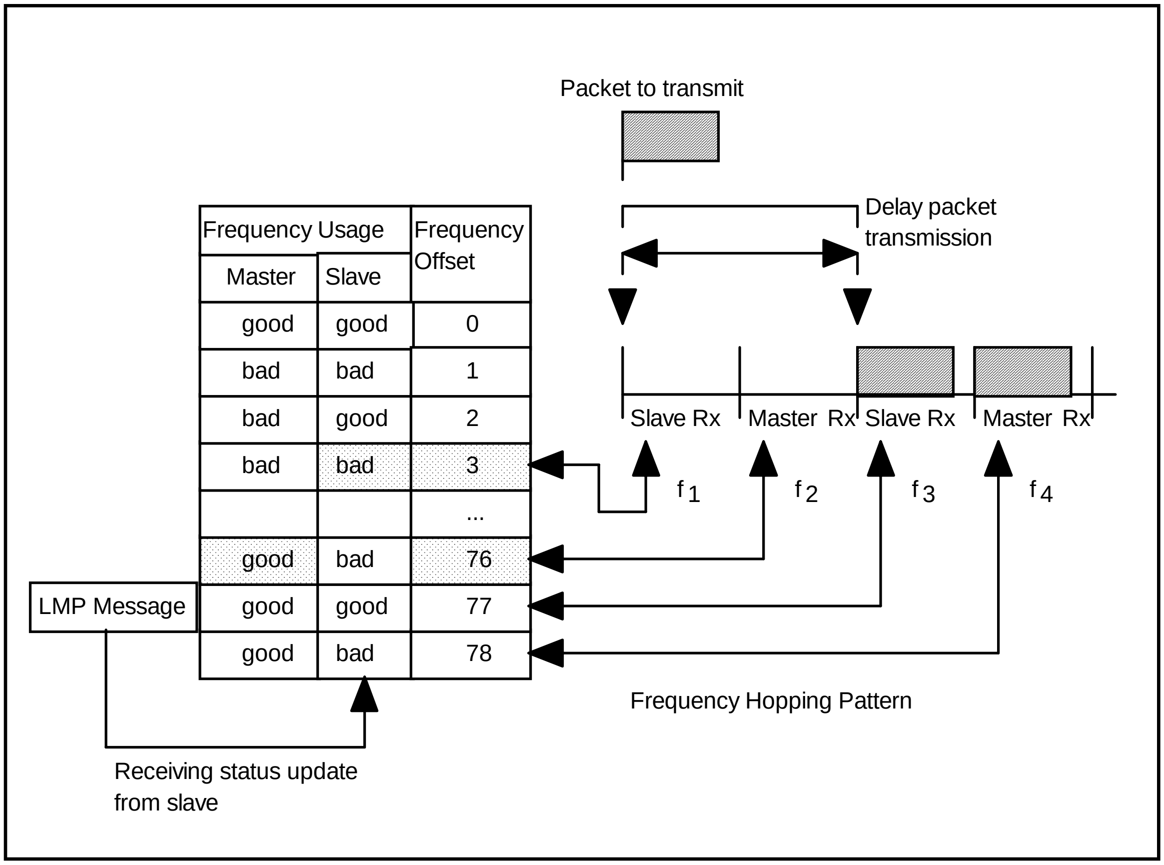

Figure 16—Delay scheduling policy at IEEE 802.15.1 master

The main steps of the scheduling policy implemented at the IEEE 802.15.1 master device are as follows:

a) Perform channel classification as described in 11.2.2.
b) Before sending a packet, check the slave's receiving frequency and the master's following receiving frequency, delay transmission until both master and slave's receiving frequencies are available.

Annex M contains numerical results for packet scheduling for ACL links.

11. Channel classification

Channel classification is required in the non-collaborative coexistence mechanisms for IEEE Std 802.15.1-2002. Adaptive packet selection and scheduling adjusts the packet types and transmission timing according to the channel condition of the current hopping channel. AFH generates the new hopping sequence based on the result of channel classification.

The purpose of channel classification is to determine the quality of each channel needed for packet or channel adaptation. The major concern of the quality should be interference. An interference-free (or low-interference) channel is classified as a “good” channel, while an interference laden (or high-interference) channel is classified as a “bad” channel. Channel classification information may then be passed between the master and the slave using link manager protocol (LMP) commands.

The channel classification implementation is up to coexistence mechanism solution vendors, so 11.1 and 11.2 show examples of channel classification methods and procedures. There may be vendor-specific variations and even implementations besides the examples described here. For this reason a method for qualifying and accepting a channel classification implementation requires an objective criterion and a testing procedure, which are beyond the scope of this recommended practice.

11.1 Methods of classification

There are multiple acceptable channel classification methods. This subclause exemplifies several channel classification schemes: RSSI, PER, and carrier sensing, which may be used separately or jointly. Once the channels have been classified, the classification list (a bit map standing for conditions of different channels) will be used to compile a final list of "good" and "bad" channels. The devices will then adaptively select and schedule packets or hop to a new sequence based on this classification list (with channel information exchanging via LMP commands).

The classification methods should use time based averaging to avoid incorrect classification due to instantaneous disturbances (e.g., other frequency-hoppers).

11.1.1 Received signal strength indication

RSSI may be used to evaluate channel condition and thus classify the channels. There may be different usages for RSSI.

One example is: if RSSI is high and an error is detected, the channel is likely to suffer from interference and is considered as "bad" channel. On the other hand, in time slots where no response is expected, the master may monitor RSSI. The averaged RSSI for each channel is recorded, and a threshold is applied at the end of the classification interval. The threshold is vendor specific. This then allows for the channel classification list to be compiled for later use.

Based on RSSI it is possible to distinguish whether the channel is classified as "bad" either due to interference or propagation effects. For example, if the packet has not been decoded successfully and RSSI has been low the error(s) nature is propagation effects. On the other hand if the packet has not been decoded successfully but RSSI has been high the error(s) nature is interference.

11.1.2 Packet error rate

The quality of transmission in a channel may be determined by the PER. A packet is deemed in-error due to failure to synchronize the access code (or access code correlator fails), HEC error, or cyclic redundancy check (CRC) error. By measuring the rate of in-error packets to received packets, it is possible to compile a list of PERs for each of the channels. At the expiration of the classification quantum, a channel shall be declared "bad" if its PER exceeds the system defined threshold. This threshold is vendor specific.

At any receiving time slot (i.e., each odd time slot), the master will know whether to expect a packet from one of the slaves. These packets (during connection) contain at least an access code and a header. A packet error is declared if the access code correlator fails, the HEC fails or, the CRC fails for a payload bearing packet.

Likewise, the slave may also compute on the received packets for channel classification. Each time that a packet is received by a slave, it requires that both the access code and header be received correctly, and the CRC on the payload shall be checked as well. If the CRC is correct, the packet has been received correctly, otherwise the packet is declared as in-error. In the same way, the slave may compute the PER and apply a threshold to compile the classification list.

Similar to the PER concept, it is also possible to consider separate metrics, such as HEC error profile, or BER profile.

The PER method is quite simple and straightforward, however it alone cannot directly distinguish whether the "bad" channel is due to interference or some other channel adverse conditions. Therefore, PER should be used in conjunction with some other method to better serve the coexistence mechanisms.

11.1.3 Carrier sensing

Carrier sensing is more robust and helps to classify the type of the interference. Within a specific time interval, an interfered channel is identified upon detection of a high-rate IEEE 802.11b PHY layer signal. The scheme is similar to that of clear channel assessment (CCA) Mode 4, defined in IEEE Std 802.11b-1999.

11.1.4 Packet acknowledgement

Channel estimation may be inferred from the built-in ACK mechanism implemented in IEEE 802.11 and IEEE 802.15 so that no explicit communication about the channel state is needed between the transmitter and the receiver. In the case of IEEE 802.15.1, the receipt of a packet with the NACK bit set in the header may indicate that the previous packet sent is lost. Similarly, a receiver expecting an acknowledgement, infers that a packet it sent is lost if it cannot correctly decode the packet containing the acknowledgment bit. In the case of IEEE 802.11, if no ACK is received for a frame that requires it, the transmitter infers that the packet it sent is lost.

11.2 Procedures of classification

This subclause describes the procedure for channel classification. The classification procedure may be executed at the slave side or at the master side. The master may integrate the channel classification returned from the slaves. The channel classification may be performed by blocks, during the connection state or offline. Subclauses 11.2.1 through 11.2.6 elaborate on each of these procedures.

11.2.1 Slave's classification data

A slave may perform channel classification and send the classification data to the master when it is requested by the master. Each channel is classified as one of the two types: "good" or "bad." The slave's classification data should be transmitted via LMP commands.

11.2.2 Master's classification

The master should perform channel classification. The master may collect slaves' classification data. The master should make the final decision for the channel classification of the piconet. Each channel is classified as one of the two types: "good" or "bad". The master may collectively use the information responded from the slaves to make the decision, or it may put more weight on the data collected by itself.

11.2.3 Integrating slaves' classification data

The slave may classify channels based on one of the methods described in 11.1. This subclause discusses the method that the master may use the classification information from multiple slaves to compile a list of "good" and "bad" channels.

There may be up to seven active slaves in a piconet, and each may support the function to produce a classification list. Once these classification lists have been received by the master, they should be integrated into the final classification list.

For the master to evaluate and classify for the overall channel conditions, the following are needed:

$S_{i,j}$ = slave *i*'s assessment of channel *j*, either “good” (1) or “bad” (0),
M_j = master's assessment of channel *j*, either “good” (1), or “bad” (0),
N_C = number of channels (79 or 23), depends on mode,
N_S = number of slaves which have sent back their classification data,
W = weighting function for the master-centric integration,

where

$$W(M_j,S_{i,j}) = \alpha \times M_j + (1-\alpha) \times S_{i,j}$$

α is the master-centric weighting factor, 0<= α<=1.

where the quality of channel *j* is given by:

$$Q_j = \frac{M_j + \sum_{i=1}^{N_s} W(M_j,S_{i,j})}{1 + N_s}$$

and

$1 <= N_S <= 7$ and

$0 <= j <= N_C$.

To determine if indeed a channel is “good,” a threshold should be applied to Q_j to determine if the quality of channel *j* is high enough.

The master then compiles the final list of “good” and “bad” channels to be distributed to every supporting device in the piconet.

11.2.4 Block channel classification

To reduce the time that classification will take, it is possible to reduce the number of measurements required at each channel. The procedure is to group channels into blocks and classify the blocks instead of the channels. This will compromise the accuracy of the measurements at each channel.

Using the PER and RSSI joint classification method as an example. If RSSI is above the threshold and a packet is detected in-error, the packet shall be deemed suffered from an interference collision event. The interference collision ratio (ICR), the ratio of interference collision events to sum of interference-free events and interference collision events, is used as the metric to assess channel conditions. It is recommended that the requirements be as follows:

N_C = number of channels (79 or 23), depends on mode,
N_{BLK} = new channel block size where,
ICR_{N_C} = interference collision ratio on each of the N_C channels,

where

$$ICR_{N_C} \in \Re[0,1]$$

$ICR_{\frac{N_C}{BLK}}$ = interference collision ratio on each of the $\frac{N_C}{BLK}$ blocks

where

$$ICR_{\frac{N_C}{BLK}} \in \Re[0,1]$$

thus:

$$ICR_{\frac{N_C}{BLK}}[k] = \begin{cases} \sum_{n=0}^{N_{BLK}-1} \frac{ICR_{N_C}[k \times N_{BLK} + n]}{N_{BLK}} & 0 \le k < \left\lfloor \frac{N_C}{N_{BLK}} \right\rfloor \\ \sum_{n=0}^{N_C \bmod N_{BLK}} \frac{ICR_{N_C}[k \times N_{BLK} + n]}{N_C \bmod N_{BLK}} & k = \left\lfloor \frac{N_C}{N_{BLK}} \right\rfloor, \text{ if } ((N_C \bmod N_{BLK}) \neq 0) \end{cases}$$

the resolution of the interference collision ratio is less accurate per channel, however the time required to complete the classification might be reduced by a factor of N_{BLK}.

11.2.5 Online classification

Online classification takes place at a time in which there is a connection with other devices (i.e., connection state). During the connection state, it is advantageous to use single slot packets (such as DM1 or DH1 packets) for channel classification. This will increase the number of packets that may be used for the channel classification measurements and decrease the likelihood of an incorrect classification. Using such packets will allow for the device to dedicate a much shorter period of time to channel classification.

Instead of sending a packet to actively probe the channels, the device may make background RSSI measurement during idle slots. This avoids extra traffic transmitted to the air due to active probing.

11.2.6 Offline classification

Offline classification takes place at a time in which there is no connection with other devices. This classification will involve background RSSI measurements. These measurements are completed quickly so that the classification interval shall be reduced.

To implement this kind of classification, the master will typically start scanning the channels in the background. Once the channels have been scanned for a long enough amount of time, a threshold is applied to the measurements, and those channels that exceed the threshold will be deemed “bad” channels.

Annex A

(informative)

Packet scheduling for SCO links

Voice applications are among the most sought-after applications for IEEE 802.15.1 devices, and they are most susceptible to interference. An in-band adjacent WLAN network will almost certainly make the voice quality of the IEEE 802.15.1 SCO link unacceptable for users. In this annex, improvements are described that may significantly improve the QoS for SCO links.

A.1 SCO scheduling algorithm for coexistence enhancement

The key idea is to allow the SCO link the flexibility of choosing the hops that are out-of-band with the collocating IEEE 802.11b network spectrum for transmission. The duty cycle or channel utilization of the SCO link does not change. The only change proposed is to allow the piconet master the flexibility of choosing when to initiate the transmission.

In particular, given that only the original HV3 packet allows for sufficient flexibility in moving the transmission slots around (2 additional choices), the focus is on modifying the HV3 packet. A new SCO packet type, EV3 packet, is defined, which has the following features:

a) no FEC coding,
b) 240 bits payload,
c) one EV3 packet for every 6 slots (delay<3.75 ms), and
d) slave will only transmit when addressed by the master.

Figure A.1 shows the difference between an HV3 packet and an EV3 packet. For HV3 packets, the transmission for master and slave shall happen at the fixed slots, no matter if the hops are "good" or "bad". In this example, the first pair of packets will be in-error because they are transmitted in "bad" channels. An EV3 packet is not transmitted during the two "bad" hops, but waits for the next pair of slots, which happens to be a "good" channel. The throughput for IEEE 802.15.1 will be higher while interference is reduced.

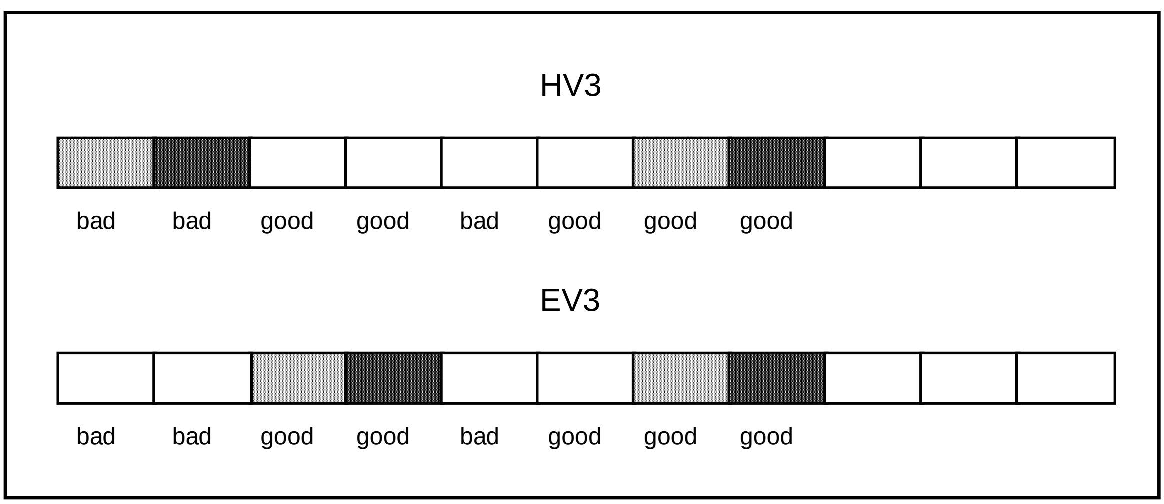

Figure A.1—Comparisons of HV3 and EV3 packet

The algorithms for selecting the best pair of slots out of the 3 available pairs shown in Figure A.1 are pretty straightforward. A score of 0 to 3 is assigned to each pair and the pair with the highest score is selected.

Algorithm for selecting the best TX slots

```
Score(n) = 0, if hop(2*n) and hop(2*n+1) are both "bad" channels
      1, if hop(2*n) is "bad" and hop(2*n+1) is "good"
      2, if hop(2*n) is "good" and hop(2*n+1) is "bad"
         3, if both are "good" channels
TXSlot=0; MaxScore=0;
For(n=0;n<3;n++)
   if(Score(n)>MaxScore)
            TXSlot=2*n;
            MaxScore = Score(n);
```

A.2 Performance simulation

This subclause provides simulation results comparing the new EV3 packet to the original HV3 and HV1 packets in coexistence environments. The simulation results are obtained with OPNET Modeler. Collisions in the radio link, which are in-band packets that overlap in time, only are considered. A collision results in a packet error for both packets. These are valid assumption for the considered scenario (< 1 meter separation), two IEEE 802.11b stations and two IEEE 802.15.1 stations in simulations. IEEE 802.15.1 devices are turned on after 15 seconds.

Figure A.2 shows the comparison of the three voice packet types.

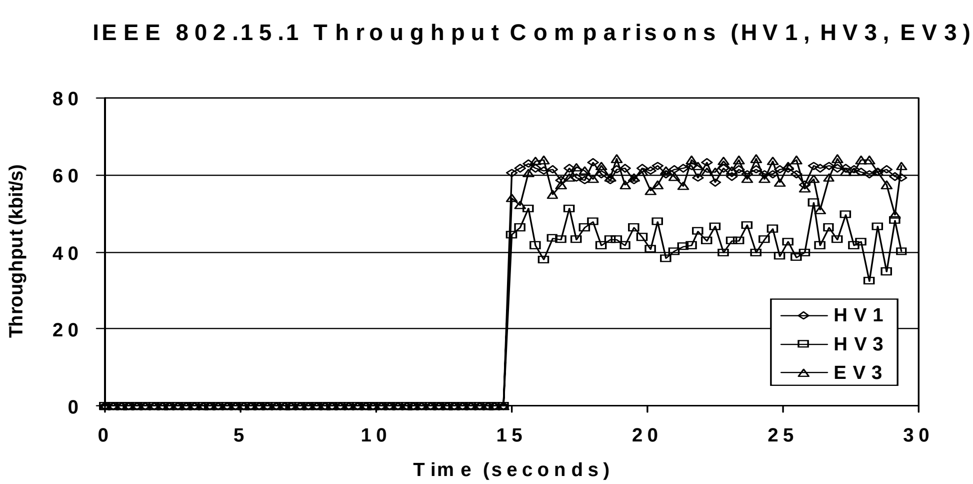

Figure A.2—Comparisons in IEEE 802.15.1 throughput for HV1, HV3, and EV3 packets

Figure A.3 shows that the EV3 packet, which does SCO packet scheduling, provides improvements in throughput for both the IEEE 802.11b and IEEE 802.15.1 devices. Because the SCO packets avoid the IEEE 802.11b band, the improvement for the IEEE 802.11b throughput is especially significant.

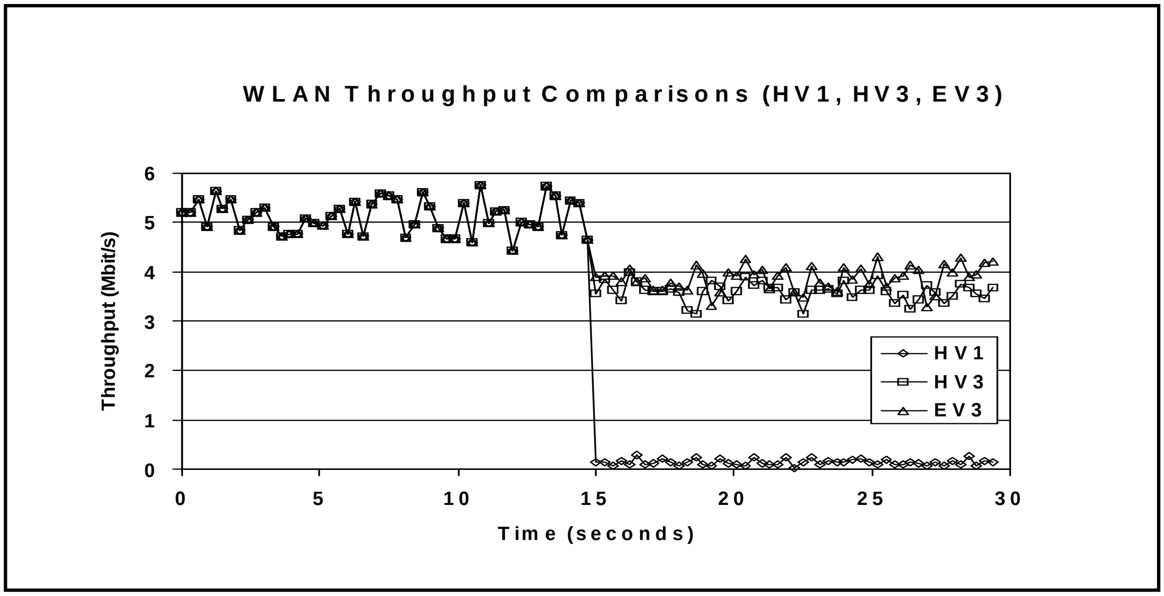

Figure A.3—Throughput comparisons for 802.11b network when collocated IEEE 802.15.1 network uses HV1, HV3, or EV3 packet

Annex B

(informative)

IEEE Std 802.15.1-2002 AFH

AFH is a non-collaborative mechanism that enables the coexistence of IEEE 802.15.1 devices with frequency static devices in the 2.4 GHz unlicensed frequency band, such as IEEE 802.11b (WLAN). This mechanism dynamically changes the FH sequence in order to avoid or mitigate the interference seen by the 802.15.1 device. This information is included for historical reference to the numerous attempts to harmonize this mechanism with the Bluetooth®[11] SIG.

There are four main elements of the adaptive hopping procedure:

a) AFH capability discovery: AFH capability discovery occurs to inform the master as to which slave supports AFH and the associated parameters.
b) Channel classification: Classification of the channels occurs in the master device and optionally in the slaves. Classification is the process by which channels are classified as either “good” or “bad.”
c) Channel classification information exchange: The channel classification information is exchanged between the master and the supporting slaves in the piconet. This is done in a reliable manner using special AFH LMP commands.
d) Adaptive hopping: Adaptive hopping is the operation of hopping over a subset of channels.

B.1 AFH mechanism

A block diagram of the AFH mechanism is shown in Figure B.1. This mechanism consists of the three distinct components: the selection box, the partition sequence generator, and the frequency re-mapping function. The first component of the AFH mechanism is the legacy hop kernel, which generates the hopping sequence defined in the IEEE Std 802.15.1-2002.

[11]Bluetooth® is a trademark owned by Bluetooth SIG, Inc.

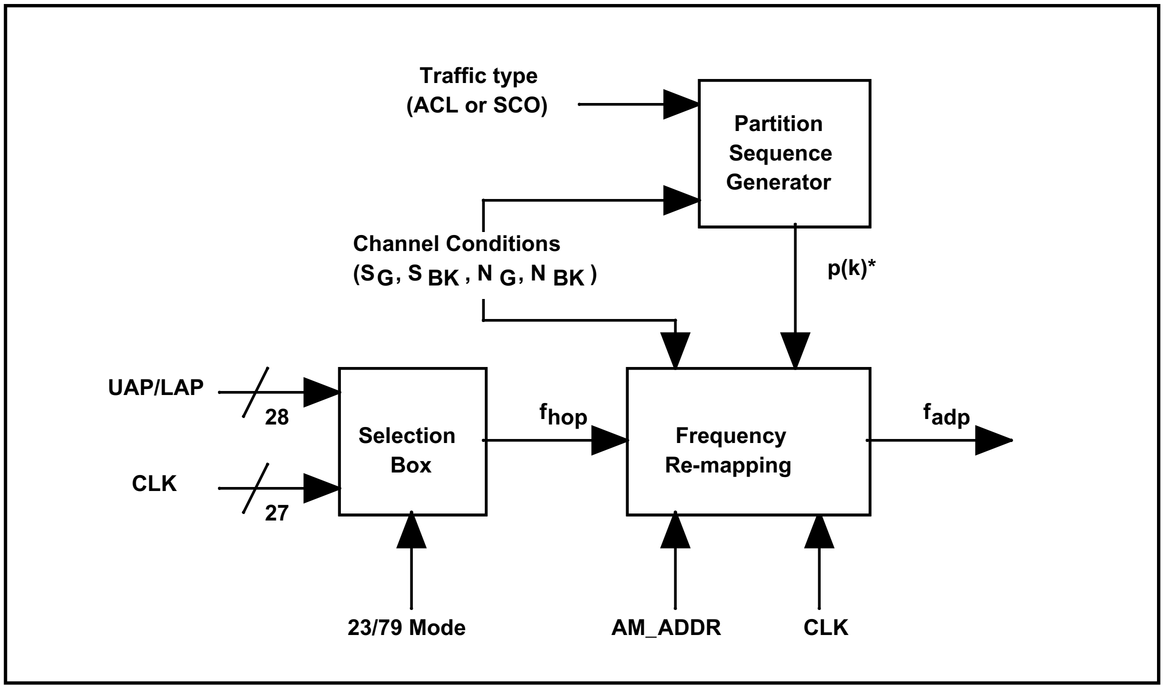

Figure B.1—Block diagram of the AFH mechanism

*when $N_G >= N_{min}$, $p(k)$ = 1; when $N_G < N_{min}$, $p(k)$ is an element of {0,1}

The second component of the AFH mechanism is the partition sequence generator, which imposes a structure on the original hopping sequence. When the new sequence is viewed from the perspective of the sets (either the set of “good” channels or the set of “bad” channels that are to be kept), there is a clear pattern and grouping of hopping frequencies from the same set. However, when the sequence is viewed from the perspective of the hopping frequencies, it still appears to be random. An example of a structured adaptive hopping sequence is illustrated in Figure B.2.

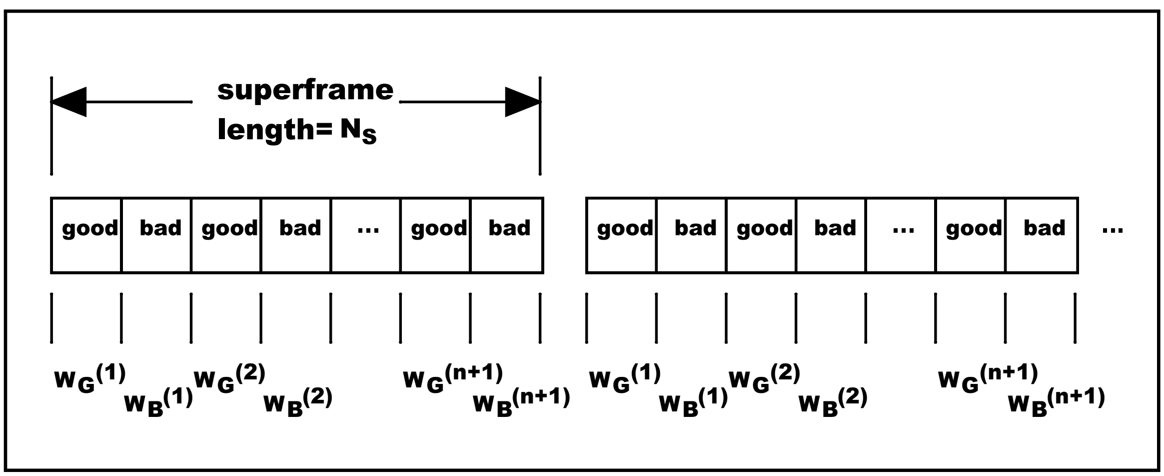

Figure B.2—An example of an adapted hopping sequence with structure

This particular hopping sequence has $W_G^{(1)}$ successive hop-frequencies from the set of “good” channels (S_G), followed by $W_B^{(1)}$ successive hop-frequencies from the set of “bad” channels to be kept (S_{BK}), followed by $W_G^{(2)}$ successive hop-frequencies from the set of “good” channels (S_G), and so on.

The window lengths of the partition sequence shall be even and satisfy the following two equations:

$$W_G^{(i)} \in \begin{cases} \{2,4,6,K,2N_G\} & \text{when } (i = 1) \\ \{0,2,4,K,2N_G\} & \text{when } (i = 2,3,K,(n+1)) \end{cases}$$

$$W_B^{(i)} \in \{0,2,4,6,K,2N_{BK}\} \qquad \text{when } (i = 1,2,K,(n+1))$$

It is possible for the “bad” windows to have length zero. The “good” windows may also have length zero, except for the first “good” window whose length shall be nonzero.

Note that the imposed structure does not specify the exact frequency at each slot, but does require that the hopping frequency be within a particular set.

The structure of the hopping sequence may be compactly represented by a partition sequence. This sequence specifies the set (either S_G or S_{BK}) of the next hopping frequency. At the k-th slot, the partition sequence may take on one of the following two values:

$$p(k) = \begin{cases} 1 & \text{if } f_{adp} \text{ should be an element of } S_G \\ 0 & \text{if } f_{adp} \text{ should be an element of } S_{BK} \end{cases} \tag{B1}$$

The output of the partition sequence is then used as an input to the final component of the AFH mechanism: the frequency re-mapping function, which generates an adapted hopping sequence with the appropriate structure. The basic idea behind the frequency re-mapping function is to re-map (if necessary) the hopping frequency produced by the legacy kernel uniformly on to the set (either S_G or S_{BK}) defined by the partition sequence. Note that when the input to the frequency re-mapping function is constant signal of one, i.e., $p(k) = 1$ for all k, the block diagram shown in Figure B.1 produces an adapted hopping sequence that only hops over the “good” channels.

In the remainder of this subclause, a detailed description of the partition sequence generator and the frequency re-mapping function is provided.

B.1.1 Partition sequence generator

The adaptive frequency-hopping mechanism shall be provided a list of “good” channels (S_G) and “bad” channels (S_B) in the spectrum. The set of “bad” channels shall then be further divided into the set of “bad” channels that are to be kept in the hopping sequence (S_{BK}) and into the set of “bad” channels that are to be removed from the hopping sequence (S_{BR}). The actual size of these partitions depends on the minimum number of hopping channels allowed (N_{min}). The size of each partition is given by the following two equations:

$$N_{BK} = \max(0, N_{min} - N_G) \text{ and} \tag{B2}$$

$$N_{BR} = N_B - N_{BK} \tag{B3}$$

To simplify the implementation complexity, the set S_{BK} should be comprised of the first N_{BK} elements of S_B, while the set S_{BR} should be comprised on the remaining elements of S_B.

In general, the optimal window lengths ($W_G^{(i)}$ and $W_B^{(i)}$) for the structure defined in Figure B.2 will depend upon the number of “good” and “bad” channels available in the band. First, consider the case when $N_G >= N_{min}$. The optimal window lengths, for this case, are given by:

$$n = 0,\ W_B^{(1)} = 0, \text{ and } W_G^{(1)} = 2N_G \Rightarrow p(k) = 1 \forall k \tag{B4}$$

Note that this result holds for both an ACL and SCO connection. Intuitively, this result implies that the optimal structure for the hopping sequence should be composed of only “good” hop-frequencies, i.e., $p(k) = 1$ for all k. In other words, when $N_G >= N_{min}$, reduced adaptive frequency-hopping (hopping only over the “good” channels) should always be used.

In the remainder of this subclause, the optimal window lengths when $N_G < N_{min}$ will be determined. The values for when there is no voice connection (ACL only) and when there is at least one voice connection (SCO+ACL) will be derived separately.

B.1.1.1 ACL only connection

For an ACL connection, the implementation complexity may be reduced by forcing the first n “good” windows to have equal length ($W_G^{(i)} = W_G^{(1)}$ for i an element of {2, ..., n}) and the first n "bad" windows to have equal length ($W_B^{(i)} = W_B^{(1)}$ for i an element of {2, ..., n}). Figure B.3 shows the structure of this new sequence.

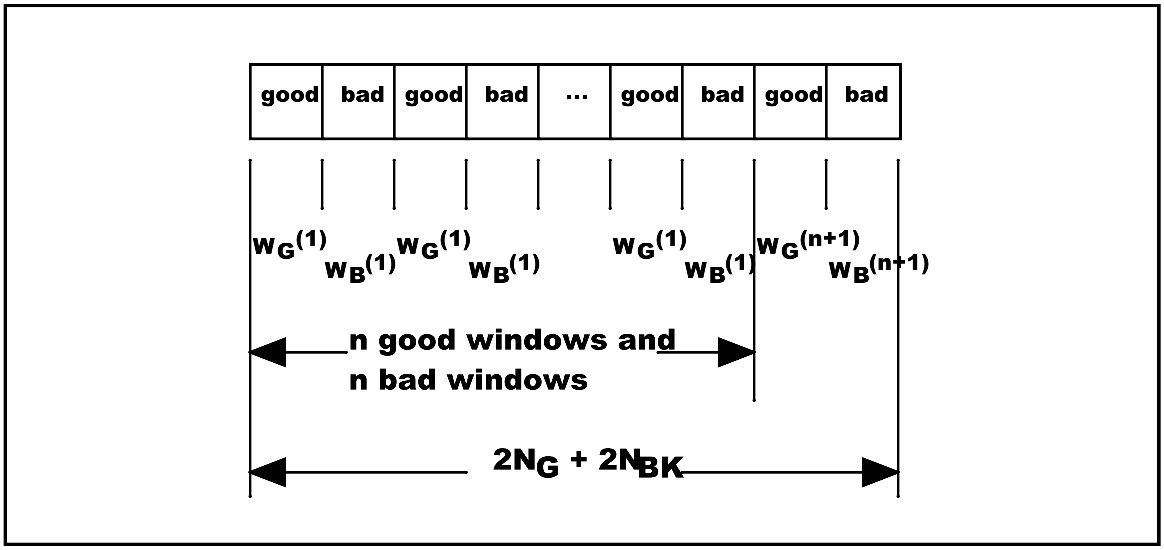

Figure B.3—A structured adaptive hopping sequence for an ACL link

To maintain a proper ratio of “good” hopping frequencies to “bad” hopping frequencies, the total number of “good” and “bad” hopping frequencies within a period of the partition sequence shall be equal to 2 N_G + 2 N_{BK}. Thus, the period of the partition sequence should also be equal to 2 N_G + 2 N_{BK}.

The length of the first “bad” window shall be constrained by the time-out value to prevent a loss in network connectivity. The size of $W_B^{(1)}$ is determined by this time-out value as follows:

$$W_B^{(1)} = \min\left\{2\left\lfloor\frac{N_{BK}}{2}\right\rfloor, 2\left\lfloor\frac{T_d}{2T_s}\right\rfloor\right\}, \tag{B5}$$

where

T_d is the time-out value for the higher layer and
T_s is the IEEE 802.15.1 slot time.

The implementer should select an appropriate value for T_d

The size of the last “bad” window is given by:

$$W_B^{(n+1)} = 2N_{BK} - nW_B^{(1)} \tag{B6}$$

where

n is defined by the following equation:

$$n = \left\lfloor \frac{2N_{BK}}{W_B^{(1)}} \right\rfloor \tag{B7}$$

The length of the last “bad” window is always guaranteed to be smaller than the length of the first “bad” window, and therefore, a time-out should never occur at the higher layers.

Given the value of *n*, the optimal values for the “good” window lengths may now be determined:

$$W_G^{(1)} = 2\left\lfloor \frac{N_G}{n+1} \right\rfloor \tag{B8}$$

$$W_G^{(n+1)} = 2N_G - nW_G^{(1)} \tag{B9}$$

Equations (B5) through (B9) define the optimal structure of the adapted hopping sequence for an ACL connection when $N_G < N_{min}$. An example of a partition sequence for an ACL connection is shown in Figure B.4.

$$\underbrace{1\,1\,\ldots\,1}_{w_G^{(1)}}\;\underbrace{0\,0\,\ldots\,0}_{w_B^{(1)}}\;\ldots\;\underbrace{1\,1\,\ldots\,1}_{w_G^{(1)}}\;\underbrace{0\,0\,\ldots\,0}_{w_B^{(1)}}\;\underbrace{1\,1\,\ldots\,1}_{w_G^{(n+1)}}\;\underbrace{0\,0\,\ldots\,0}_{w_B^{(n+1)}}\;\underbrace{1\,1\,\ldots\,1}_{w_G^{(1)}}\;\underbrace{0\,0\,\ldots\,0}_{w_B^{(1)}}\;\ldots$$

Figure B.4—An example partition sequence for an ACL connection

The following pseudo-code summarizes the partition sequence generator for an ACL only connection.

Partition Sequence Generator for an ACL only connection:

```
/* Check to see if reduced adaptive frequency-hopping may be used */
If (N_G >= N_min) Then,
/* Generate partition sequence for reduced adaptive frequency-hopping*/
While (afh_is_still_active),
   p(k) = 1
      End
/* The case when "bad" hopping frequencies shall be used in the adapted
*/
/* hopping sequence */
Else,
   /* Initialization - determine parameters for partition sequence */
   /* generator (performed only once)*/
   W_B^(1) = 2floor[ T_d / (2T_s)]
   n = floor[ 2 N_BK / W_B^(1)]
   W_B^(n+1) = 2 N_BK - n W_B^(1)
   W_G^(1) = 2floor[ N_G / (n + 1)]
   W_G^(n+1) = 2N_G - n W_G^(1)
   /* Generate partition sequence for structured adaptive */
   /* hopping sequence */
   While (afh_is_still_active),
       /* Loop through all of the "good" and "bad" windows */
       For index = 1 to n+1,

           /* Check to see if in the "good" and "bad" window*/
           If (index is not equal to n) Then
               W_G = W_G^(1) and W_B = W_B^(1)
           Else
               W_G = W_G^(n+1) and W_B = W_B^(n+1)
           End

           /* Loop through the "good" window and generate */
           /* partition sequence*/
           For loop = 1 to W_G
               p(k) = 1
           End
           /* Loop through the "bad" window and generate */
           /* partition sequence*/
           For loop = 1 to W_B
               p(k) = 0
           End
       End
   End
End
```

The same partition sequence value is assigned to both the master and the slave. By updating the partition sequence generator only on the master-to-slave slot, the complexity of this generator may be further reduced. To increase the robustness of the ACL link, the AFH mechanism should be used in conjunction with a packet-scheduling algorithm.

B.1.1.2 SCO + ACL connection

A water-filling approach is used to design the structure of the partition sequence when there is at least one voice connection active in the piconet. First, the “good” channels are distributed on slots where voice packets are to be transmitted. Because the SCO packets are not protected by an ARQ mechanism, they are given higher protection by the partition sequence generator. If there are any remaining “good” channels, then these channels are uniformly distributed to the ACL traffic. Finally, the “bad” channels are assigned to the remaining slots.

Let V be the voice link type (V = 1 for HV1, V = 2 for HV2, V = 3 for HV3) for the SCO connection. Because the voice connection is periodic, it is more convenient to view the structured adaptive hopping sequence in terms of ($N_G + N_{BK}$) frames of length 2V (see Figure B.5), where F_i denotes the i-th frame.

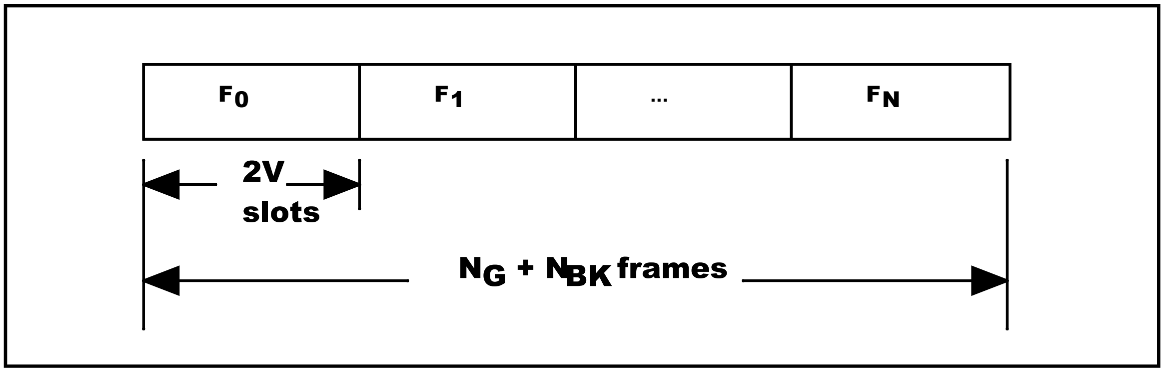

Figure B.5—A structured adaptive hopping sequence for an SCO link

The structure defined in this figure is perfectly aligned with the inherent structure of the SCO link. The period of the partition sequence should be equal to $2V(N_G + N_{BK})$.

To maintain a proper ratio of the "good" hopping frequencies to the “bad” hopping frequencies, $2VN_G$ “good” hopping frequencies shall be distributed among the ($N_G + N_{BK}$) frames. Before the “good” hopping frequencies may be distributed, the number of voice streams V’s that may be supported (i.e., place a “good” hopping frequency on each slot where a voice packet needs to be transmitted) shall be determined. The following relationship may be used to determine this value:

$$V_s = \left\lfloor \frac{VN_G}{N_G + N_{BK}} \right\rfloor \tag{B10}$$

This result implies that, at minimum, 2 V_s “good” hopping frequencies should be placed in each frame. The number of “good” hopping frequencies that remain is given by the following:

$$R_G = 2VN_G - 2V_s(N_G + N_{BK}) \tag{B11}$$

To ensure the best level of QoS, the residual “good” hopping frequencies should be uniformly distributed across the frames. The distance between frames that guarantees even placement of the residual “good” hopping frequencies is given by:

$$D = \left\lceil \frac{2(N_G + N_{BK})}{R_G} \right\rceil \tag{B12}$$

This last result implies that an additional two “good” hopping frequencies may be assigned to the following frames: F_0, F_D, F_{2D}, etc. In certain cases, it may be possible that a few “good” hopping frequencies have not yet been placed. The number of unplaced “good” hopping frequencies is given by:

$$E_G = R_G - 2\left\lceil \frac{(N_G + N_{BK})}{D} \right\rceil \tag{B13}$$

The most convenient way to distribute these “good” hopping frequencies is to assign them two at a time to the following frames:

F_1, F_{D+1}, F_{2D+1}, etc. until they have all been placed.

So in conclusion, the number of "good" hopping frequencies that are assigned to the *i*-th frames is given by:

$$G_i = \begin{cases} 2V_s + 2 & \text{if mod(i,D) = 0 or (mod(i,D) =1 and } (\lfloor i/D \rfloor < E_G/2) \\ 2V_s & \text{otherwise} \end{cases} \tag{B14}$$

The “bad” hopping frequencies are then used to ensure that $2V$ hopping frequencies have been assigned to each frame. The exact placement of the “good” and “bad” hopping frequencies within a frame depends on the number of voice streams that are active and the offset (D_{SCO}) for stream. Table B.1, Table B.2, and Table B.3 describe the partition sequences of entire frame for the various cases.

Table B.1—Partition sequence values for HV1 SCO connection (V = 1)

N_v = # of HV streams	D_{SCO}	G_i = # of “good” channels assigned to F_i	Partition sequence, p(k)
1	0	0	[0 0]
1	0	2	[1 1]

Table B.2—Partition sequence values for HV2 SCO connection (V = 2)

N_v = # of HV streams	D_{SCO}	G_i = # of “good” channels assigned to F_i	Partition sequence, p(k)
X	X	0	[0 0 0 0]
1	0	2	[1 1 0 0]
1	2	2	[0 0 1 1]
2	0,2	2	[1 1 0 0]
X	X	4	[1 1 1 1]
X means do not care.			

Table B.3—Partition sequence values for HV3 SCO connection (V = 3)

N_v = # of HV streams	D_{SCO}	G_i = # of “good” channels assigned to F_i	Partition sequence, p(k)
X	X	0	[0 0 0 0 0 0]
1	0	2	[1 1 0 0 0 0]
1	2	2	[0 0 1 1 0 0]
1	4	2	[0 0 0 0 1 1]
1	0	4	[1 1 1 1 0 0]
1	2	4	[1 1 1 1 0 0]
1	4	4	[1 1 0 0 1 1]
2	0,2	2	[1 1 0 0 0 0]
2	0,4	2	[1 1 0 0 0 0]
2	2,4	2	[0 0 1 1 0 0]
2	0,2	4	[1 1 1 1 0 0]
2	0,4	4	[1 1 0 0 1 1]
2	2,4	4	[0 0 1 1 1 1]
3	0,2,4	2	[1 1 0 0 0 0]
3	0,2,4	4	[1 1 1 1 0 0]
X	X	6	[1 1 1 1 1 1]

The following pseudo-code summarizes the partition sequence generator for a SCO + ACL connection.

Partition Sequence Generator for a SCO+ACL Connection:

```
/* Check to see if reduced adaptive frequency-hopping may be used */
If (N_G >= N_min) Then,
    /* Generate partition sequence for reduced adaptive frequency */
    /* hopping*/
    While (afh_is_still_active),
        p(k) = 1
               End
/* Shall use "bad" hop-frequencies in the adapted hopping sequence */
Else,
    /* Initialization section - determine parameters for */
    /* partition sequence generator */
    V_s = floor[ VN_G / (N_G + N_BK)]
    R_G = 2 VN_G - 2 V_s(N_G + N_BK)
    D = ceil[ 2(N_G + N_BK) / R_G]
    E_G = R_G - 2 ceil[ (N_G + N_BK) / D]
    /* Generate partition sequence for structured adaptive */
    /* hopping sequence */
    While (afh_is_still_active),
        /* Loop through all of the frames*/
        For loop = 0 to (N_G + N_BK) - 1
            /* Determine the number of "good" channels to be assigned */
            /* to the i-th frames */
            G_i = 2V_s
            /* See if any addition "good" channels are to be assigned */
            /* to the i-th frame */
            If (mod(loop, D) = 0) OR
                ((mod(loop, D) = 1) AND (floor(loop/D) < E_G/2))
            Then
                G_i = Gi + 2
            End
            /* Partition sequence for that frame may be found via */
            /* table look-up */
            p(k) = table_look_up(V, N_V, D_sco, G_i)
        End
    End
End
```

By exploiting the fact that the partition sequence needs to be generated only per frame (4 slots for HV2 and 6 slots for HV3), the complexity of the partition sequence generator may be reduced. This sequence generator is designed to work with a single HV1, HV2, and HV3 stream, as well as multiple HV2 and HV3 streams. Also note that the look-up table for the HV1 stream may be eliminated and replaced by the entries for two HV2 streams, because two HV2 streams is equivalent to a single HV1 stream.

B.1.2 Re-mapping function

The frequency re-mapping function generates an adaptive hopping sequence with a structure that has been specified by the partition sequence. The actual mechanism that re-maps the hopping frequencies is fairly straightforward. If the legacy hopping frequency is already in the set that is specified by the partition sequence, then the output of the frequency re-mapping function is the legacy hopping frequency. However, if the legacy hopping frequency is not in the required set, then the index pointing to the legacy hopping frequency is re-mapped using the mechanism defined in Figure B.6 and Figure B.7.

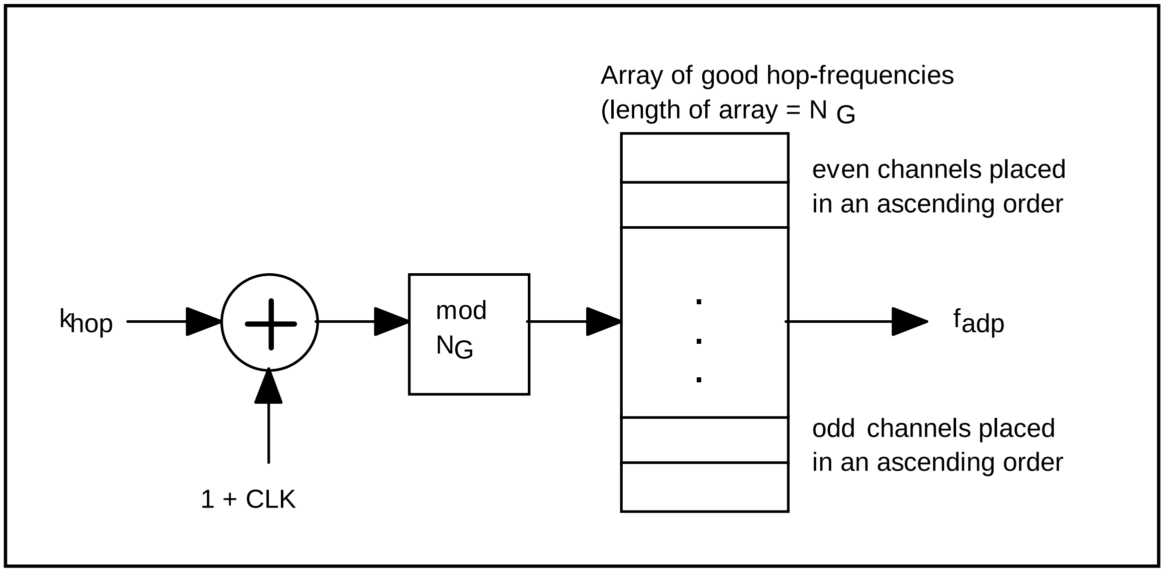

Figure B.6— Block diagram for the frequency re-mapping function on to the set $S_{G;}$

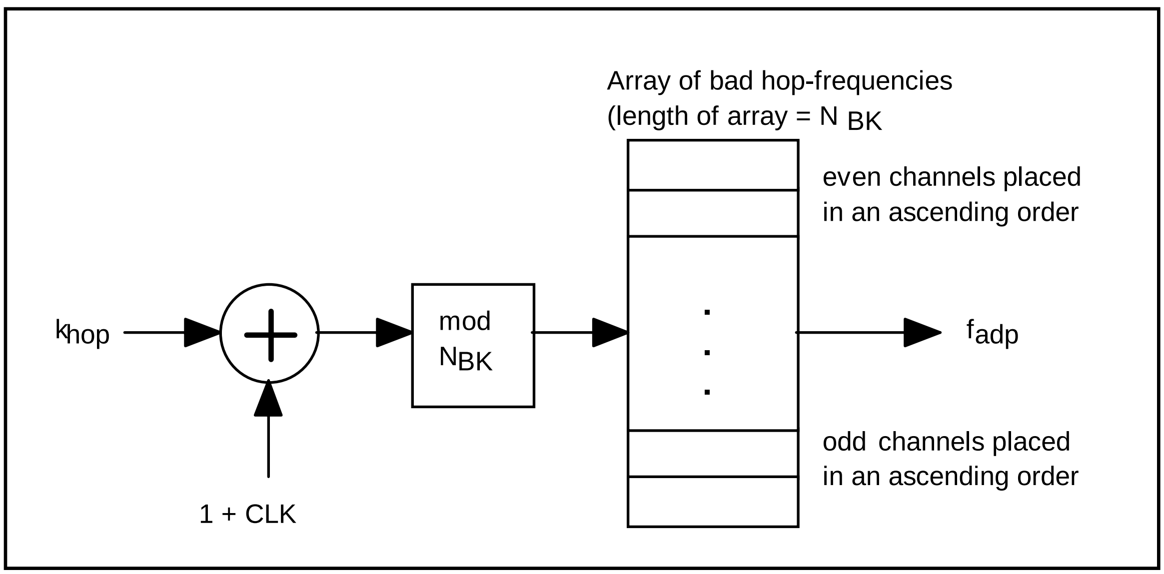

Figure B.7—Block diagram for the frequency re-mapping function on to the set S_{BK}.

The following pseudo-code summarizes the frequency re-mapping function for the AFH mechanism.

Frequency Re-mapping Function:

```
/* Find the next hopping frequency from the IEEE 802.15.1 hop kernel. */
f_hop = BT1.1_HS_generator (master_address, clock)
/* Partition sequence provides information about the set for the */
/* next channel */
p(k) = Partition_sequence_generator ()
/* If f_hop is in the required set, then re-mapping function should */
/* output f_hop */
If (p(k) = 1 AND f_hop is an element of S_G) OR
   (p(k) = 0 AND f_hop is an element of S_BK)
Then,
   f_adp = f_hop
/* If f_hop is not in the required set, then re-map f_hop to a frequency */
/*in the required set */
Else,
   /* First check to see if a "good" channel is needed */
   If (p(k) = 1) Then,
      /* Map the frequency onto a "good" hopping frequency. */
      /* First add the CLK to the frequency and then map */
      /* this result on to an element in S_G */
         Index = (k_hop + 1 + CLK) mod N_G
         f_adp = S_G (Index)
   Else,
      /* Map the frequency onto a "bad" hopping frequency that is */
      /* to be kept in the adapted hopping sequence. */
      /* First add the CLK to the frequency and then map this result */
      /* on to an element in S_BK */
      Index = (k_hop + 1 + CLK) mod N_BK
      f_adp = S_BK (Index)
   End
End
```

Note that a frequency re-mapping function is a necessity for all AFH schemes.

Annex C

(informative)

Physical layer models

The outline of this annex is as follows: C.1 introduces concepts that are useful for understanding the physical layer models, while C.2 gives the path loss model. C.3 describes an analytical model that is suitable for extended MAC-sublayer simulations. C.4 discusses a simulation-based model that is more accurate but also more computationally intensive. Presently, the results provided by the two models are not directly compared because of different definitions of signal to interference ratio.

C.1 Physical layer model concepts

This subclause introduces concepts that are common to the physical models described in this annex. The most powerful simplifying concept in this model is the period of stationarity. This is the period over which the parameters defining the transmissions of the devices being modeled do not change.

Consider the example shown in Figure C.1. Here an IEEE 802.15.1 device transmits two packets. An IEEE 802.11b device transmits a single physical protocol data unit (PPDU) using 11 Mbit/s modulation type for the physical service data unit (PSDU). The start of the PHY layer convergence protocol (PLCP) header overlaps the end of the first IEEE 802.15.1 packet. The end of the PSDU overlaps the start of the second IEEE 802.15.1 packet. There are six periods of stationarity. A new period of stationarity starts at the end of the PLCP header because the modulation type changes at this point.

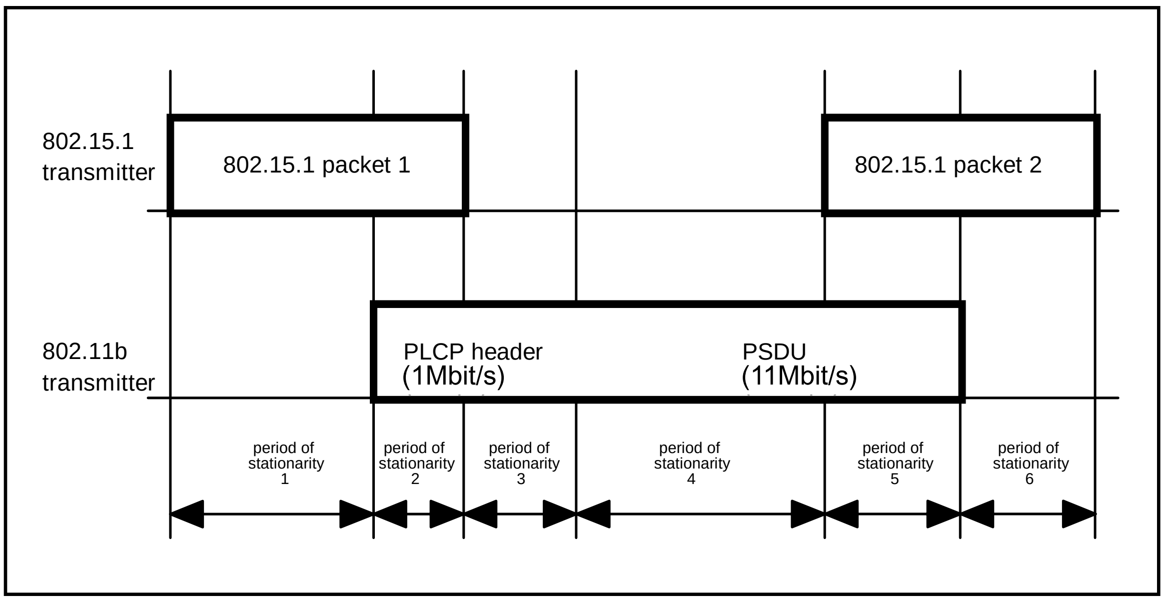

Figure C.1—Example showing periods of stationarity

By definition, during the period of stationarity the transmit power and modulation type do not change, and the position of the devices (and hence link loss) is assumed constant. So receiving nodes experience constant signal, noise and interference powers from which a BER value may be calculated or simulated.

C.2 Path loss model

The path loss model used is given by Table C.1 and shown in Figure C.2. This path loss model is described in Marquess [B18]. Path loss follows free-space propagation (coefficient is 2) up to 8 meters and then attenuates more rapidly (with a coefficient of 3.3).

The model does not apply below about 0.5 meter due to near-field and implementation effects.

Table C.1—Equations for path loss (dB) at 2.4 GHz versus distance (m)

Equation	Condition
Path loss = 40.2 + 20 $\log_{10}$(d),	0.5 m <= d <= 8 m
Path loss = 58.5 + 33 $\log_{10}$(d/8)	d > 8 m

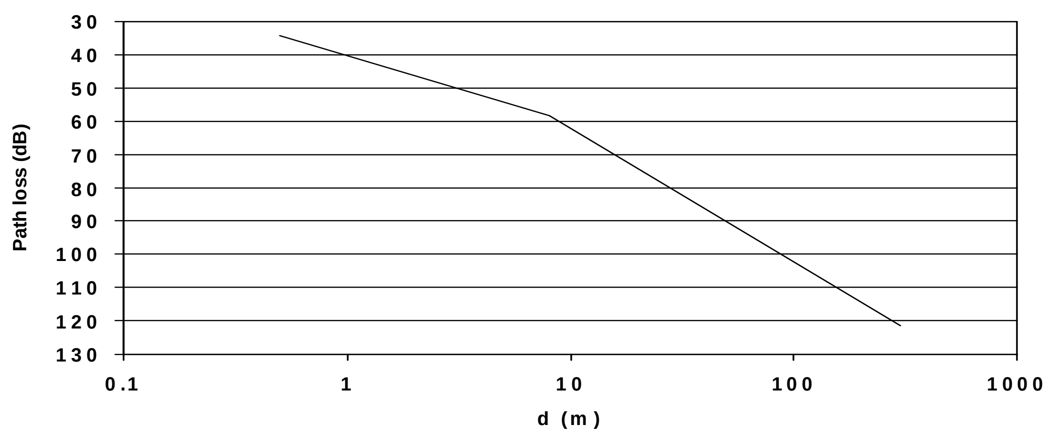

Figure C.2— Path loss (dB) versus distance (m) for empirical indoor model

C.3 Analytical model for IEEE 802.11b and IEEE 802.15.1 interference

This subclause describes the analytical model that allows the BER to be calculated for IEEE 802.11b and IEEE 802.15.1 packets in the presence of mutual interference.

C.3.1 Model interface

The model is supplied with device positions and transmission parameters. The model calculates the BER derived from those parameters.

The parameters described in Table C.2 are supplied to the PHY model for each transmission that is active during a period of stationarity.

Table C.2—Transmission parameters

Field	Description
Source position	Device position specified using Cartesian Coordinates
Destination position	
Modulation type	Type of modulation used by the transmitter. One of: 802.15.1 802.11b 11 Mbit/s 802.11b 5.5 Mbit/s 802.11b 1 Mbit/s 802.11b 2 Mbit/s
Transmit power	Transmit power
Frequency	Center frequency of transmission

The output of the PHY model is a BER value at the receiver of each transmission.

C.3.2 BER calculation

Figure C.3 shows the BER calculation in diagrammatic form.

The intended transmission is attenuated by the path loss as defined in C.2 to the receiver. The EIRP, less the path loss, is the signal power at the receiver. Each interfering transmission is attenuated by its path loss to the receiver and by the spectrum factor as defined in C.3.3 to account for the combined effect of receiver and transmitter masks and frequency offset. The resulting interference powers are added to give the total interference power. The SIR is the ratio of signal to total interference power at the receiver. The BER is given by BER (modulation type, SIR), as defined in C.3.6.

SIR is defined to be signal power/noise power,

where

a) signal power is
   1) wanted signal transmit power,
   2) path loss (distance);

and

b) noise power is the sum over all interferers of
   1) interferer transmit power,
   2) path loss (interferer distance), and
   3) spectrum factor (TX modulation type, RX modulation type, frequency offset).

The interferer transmit power is the transmit power of the interferer. The path loss is defined in C.2. The interferer distance is the distance between the interferer and the receiver. The spectrum factor is defined in C.3.3. The TX modulation type is the modulation type of the interferer. RX modulation type is the modulation type of the wanted signal. The frequency offset is the difference between the interference and the wanted signal center frequencies.

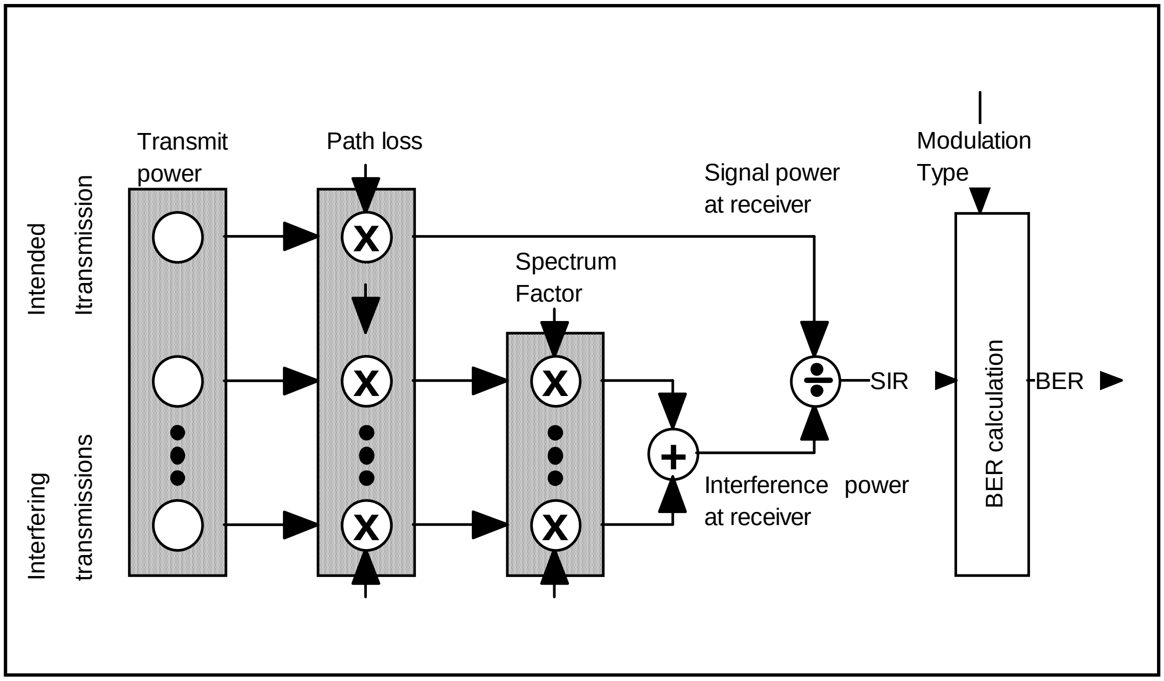

Figure C.3—BER calculation

C.3.3 Spectrum factor

The spectrum factor represents the combined effects of transmitter and receiver masks as defined in C.3.5 and frequency offset.

To calculate the spectrum factor, the transmitter mask is first normalized so that the total area under the curve is unity. The receiver mask is not normalized. The spectrum factor is equal to the integral under the curve formed by multiplying these masks together at a specified frequency offset. As a simplification, the spectrum factor of the same modulation type for receiver and transmitter with zero frequency offset is taken to be unity.

The SpectrumFactor() procedure defined in Annex D performs this operation for the IEEE 802.15.1 and IEEE 802.11b masks specified in C.3.5.

Spectrum factor values calculated using the SpectrumFactor() procedure in Annex D are shown in Table C.3 expressed in dB.

C.3.4 SIR computation

The SIR is given by the ratio of the received signal power to the total received interference power. The powers are calculated after the spectrum factor has been applied, and so this ratio corresponds to the value after the receiver filter.

Receiver noise is not considered in this model.

Table C.3—Spectrum factor values for IEEE 802.15.1 and IEEE 802.11b

Frequency offset (MHz)	Spectrum factor (dB)	
	802.15.1 to 802.11b	802.11b to 802.15.1
0–9	0.0	–12.6
10	0.0	–12.9
11	–11.4	–24.2
12	–30.1	–41.8
13	–35.9	–42.0
14–20	–36.0	–42.0
21	–52.9	–42.3
22	–55.6	–49.1
23–35	–55.7	–50.7
36–40	–55.8	–50.7
41–42	–55.8	–51.0
43–48	–55.9	–51.0

C.3.5 Transmit and receive masks

The transmit and receive masks used are defined in Table C.4 .

Table C.4—Transmit and receive masks

	Transmit[a]		Receive	
802.15.1	Frequency offset (MHz)	Attenuation (dB)	Frequency offset (MHz)	Attenuation (dB)
	0	0	0	0
	1	20	1	11
	2	40	2	41
	3	60	greater than 2	51
802.11b	Frequency offset (MHz)	Attenuation (dB)	Frequency offset (MHz)	Attenuation (dB)
	0 to 10	0	0 to 10	0
	11 to 21	30	11	12
	greater than 21	50	12 to 20	36
			greater than 20	56

[a]The transmit attenuation numbers come from the transmit power spectral density requirements of their respective standards. Typical implemenations will achieve better performance so these numbers can be considered as ‘worst case’ numbers.

C.3.6 BER calculation based on SIR

The symbol error rate (SER) is calculated for each modulation type based on the SIR at the receiver. Given the number of bits per symbol, the SER is converted into an effective BER.

Subclauses C.3.6.1 through C.3.6.8 describe the BER calculation for the different modulation types.

C.3.6.1 BER calculation for 802.15.1 modulation

Assuming envelope detection of FSK, the BER is given directly by

$$\mathrm{BER}_{802.15.1} = Q(a, b) - \left(\frac{1}{2}\right)e^{(-(a^2+b^2))/2} I_0(ab) \text{ (see Proakis [B14])}$$

where

$$a = \sqrt{\frac{\gamma}{2}(1 - \sqrt{1 - |\rho|^2})},$$

$$b = \sqrt{\frac{\gamma}{2}(1 + \sqrt{1 - |\rho|^2})},$$

$$\gamma = E_b / N_0,$$

$$\rho \in [0,1],$$

$I_0(x)$ is the modified Bessel function of order zero.

The formula for the correlation coefficient is:

$$\rho = \frac{\sin(2\pi\beta)}{2\pi\beta}$$

where β is the modulation index. The IEEE 802.15.1 standard specifies a minimum modulation index of 0.28 and a maximum modulation index of 0.35. Table C.5 gives the value of the correlation coefficient for the minimum, nominal, and maximum value of the modulation index.

Table C.5—Correlation coefficient for minimum, nominal, and maximum modulation index

Modulation index (β)	**Correlation coefficient (ρ)**
0.28	0.558
0.32	0.450
0.35	0.368

C.3.6.2 BER calculation for 802.11b 1Mbit/s

The probability of error in a symbol in the presence of additive white Gaussian noise (AWGN) is given by:

$$P = Q(\sqrt{d^2/(2N_0)})$$

where

d is the minimum distance between any two points in the signal constellation, and
N_0 is the in-band noise power at the receiver.

The Q function is defined in C.3.6.6.

In the case of a IEEE 802.11b 1 Mbit/s chip, the modulation scheme is differential binary phase shift keying (DBPSK). This has the effect of doubling the effective noise power at the receiver[12].

$$P_{DBPSK-CHIP} = Q(\sqrt{(d^2/4N_C)})$$

where

N_C is the noise energy per chip.

The value of *d* may be determined by plotting the modulation constellation of binary phase shift keying (BPSK) placing the signal points at a distance of

$$\sqrt{E_C}$$

from the origin, where

E_C is the received signal energy per chip.

Thus

$$d_{DBPSK-CHIP} = 2\sqrt{E_C}$$

So now:

$$P_{DBPSK-CHIP} = Q(\sqrt{(E_C/N_C)}) \tag{1}$$

This is the probability of an error in an individual 11 Mbit/s chip.

To include the effect of the spreading code, the squared distance is summed over each chip. In the case of IEEE 802.11b 1 Mbit/s modulation, the 11-chip spreading code results in the squared distance being multiplied by a factor of 11.[13]

[12]This doubling is slightly pessimistic for binary phase shift keying (BPSK) under conditions of high signal to interference plus noise ratio (SINR).
[13]An alternative approach giving the same result is to consider the spreading sequence to be a block code of length 11.

Giving

$$P_{1MBPS-SYMBOL} = Q(\sqrt{11 \times SIR})\,, \tag{2}$$

where

$SIR = E_C/N_C$.

This is the 1 Mbit/s SER. It is also the 1 Mbit/s BER, because each symbol encodes a single bit.

C.3.6.3 BER calculation for 802.11b 2 Mbit/s

This calculation follows the treatment for the 1 Mbit/s calculation with a few differences.

The 2 Mbit/s rate uses 11 Mbit/s differential quadrature phase shift keying (DQPSK) chips. The minimum distance between points in the quadrature phase shift keying (QPSK) constellation is reduced by a factor of

$\sqrt{2}$

(compared to BPSK) giving

$$d_{\mathrm{QBPSK-CHIP}} = \sqrt{2E_C}$$

This substitution results in

$$P_{\mathrm{2MBPS-SYMBOL}} = Q\sqrt{5.5 \times \mathrm{SIR}}$$

Each 2 Mbit/s symbol encodes two bits. However, because the symbols are Gray coded, a decoding error between adjacent DQPSK constellation points yields only a single bit error in the decoded 2 Mbit/s bit-stream[14]. Therefore, this SER is also the BER.

C.3.6.4 BER calculation for 802.11b 5.5 Mbit/s

The SER may be determined by treating the modulation as a block code in the presence of AWGN interference. The general SER[15] is

$$SER = \sum Q\sqrt{2 \times \mathrm{SIR} \times R_C \times W_m} \tag{C1}$$

where

R_C	is the code rate,
W_m	is the codeword distance, and
the sum	is over all other codewords.

For IEEE 802.11b 5.5 Mbit/s, the SER, $\mathrm{SER}_{5.5}$, is given by:

$$\mathrm{SER}_{5.5} \le 14 \times Q\sqrt{8 \times \mathrm{SIR}} + Q\sqrt{16 \times \mathrm{SIR}} \tag{C2}$$

[14]Errors between adjacent DQPSK constellation points are more likely than errors between opposing constellations points.
[15]These formulas are not accurate for small values of the SIR.

As each symbol encodes 4 bits, the BER is given by:

$$\mathrm{BER} \le \left(\frac{2^{4-1}}{2^4 - 1}\right)\mathrm{SER}_{5.5} = \frac{8}{15}\mathrm{SER}_{5.5} \qquad \text{(C3)}$$

C.3.6.5 BER calculation for 802.11b 11 Mbit/s

For IEEE 802.11b 11 Mbit/s, the SER, SER_{11}[16] is given by:

$$\begin{aligned} SER_{11} \le\ & 24 \times Q(\sqrt{4 \times SIR}) + 16 \times Q(\sqrt{6 \times SIR}) + 174 \times Q(\sqrt{8 \times SIR}) + \\ & 16 \times Q(\sqrt{10 \times SIR}) + 24 \times Q(\sqrt{12 \times SIR}) + Q(\sqrt{16 \times SIR}) \end{aligned} \qquad \text{(C4)}$$

As each symbol encodes 8 bits, the BER is

$$\mathrm{BER} \le \left(\frac{2^{8-1}}{2^8 - 1}\right)\mathrm{SER}_{11} = \frac{128}{255}\mathrm{SER}_{11} \qquad \text{(C5)}$$

C.3.6.6 Q function definition

The Q function is defined as the area under the tail of the Gaussian probability density function with zero mean and unit variance.

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-\left(\frac{t^2}{2}\right)} dt \qquad \text{(C6)}$$

In this model, a fifth-order approximation to *Q(x)* is used when *x* is greater than 1:

$$Q(x) = \left(\frac{1}{\sqrt{2\pi}}\right)e^{\left(-\frac{x^2}{2}\right)} \times \left(\frac{8 + 9x^2 + x^4}{15x + 10x^3 + x^5}\right) \qquad \text{(C7)}$$

C.3.6.7 SIR limits

The simulation is simplified by assuming that above a certain SIR the BER is effectively zero and below a certain SIR the BER is effectively 0.5. These limits are defined in Table C.6.

Table C.6—Assumed limits on SIR

Receiver	Upper limit on SIR	Lower limit on SIR[a]
802.11b	10 dB	-3 dB
802.15.1	20 dB	1 dB

[a]The lower limit handles the problems with Equation (C1) and Equation (C4).

[16]These formulas are not accurate for small values of the SIR.

C.3.6.8 BER versus SIR Results

Figure C.4 shows the results of calculating BER for SIR values in the range –2 to 10 dB for each modulation type.[17]

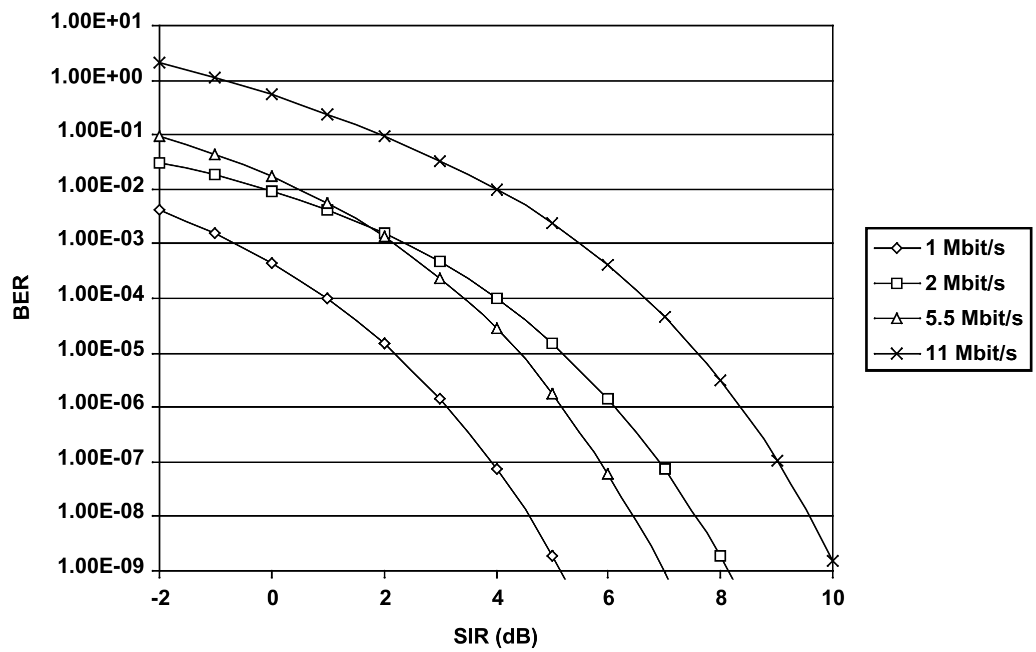

Figure C.4—BER versus SIR for 802.11b modulation types

C.4 Physical layer simulations

In this subclause, the modeling of the physical layers of the IEEE 802.15.1 and IEEE 802.11b (WLAN) systems are discussed followed by an examination of the BER performances in interference-limited environments. Complex baseband models are used for both IEEE 802.15.1 and WLAN, and the performance is determined using Monte Carlo simulation methods. While the analytical model uses transmitter power and distance as input parameters, the simulation model uses the SNR and the SIR. In both cases, the output is BER.

The outline of the clause is as follows: C.4.1 describes the model for IEEE 802.15.1 and C.4.2 describes the model for IEEE 802.11b. Subclause C.4.3 contains results for the IEEE 802.11b system in the presence of interference from IEEE 802.15.1, and C.4.4 provides the results for IEEE 802.15.1 in the presence of an IEEE 802.11b interferer. Soltanian [B20] contains additional results for flat fading channels.

C.4.1 IEEE 802.15.1 system model

The IEEE 802.15.1 system operates at a channel bit rate of 1 Mbit/s. The modulation is Gaussian frequency shift keying (GFSK) with a nominal modulation index of $h_f = 0.32$ and a normalized bandwidth of $BT = 0.5$, where B is the 3 dB bandwidth of the transmitter's Gaussian low pass filter, and T is the bit period. The IEEE 802.15.1 radio employs a FH scheme in which the carrier frequency is changed on a packet by packet basis.

[17]The results for IEEE 802.11b 11 Mbit/s do not show calculated values for SINR < –2 dB due to limitations of the tools used.

There are up to 79 (23) different channels, each with 1 MHz separation. The entire structure of the simulated system is presented in Figure C.5. It includes the transmitter, the channel, the receiver and the interference source. The IEEE 802.15.1 input data are denoted by a_i, and they are passed through a Gaussian filter and a phase modulator. Random data, denoted by b_i, are also phase modulated depending on the type of interference (either IEEE 802.15.1 or IEEE 802.11). A carrier frequency offset and a random phase are added to this signal. The simulation operates at complex baseband, so an interfering signal given by:

$$\cos(2\pi(f_c + f_d)t + \phi_d)$$

is represented by:

$$\exp(j(\omega_d t + \phi_d)),$$

where

$\omega_d = 2\pi f_d$ is the frequency offset, and
ϕ_d is the random phase.

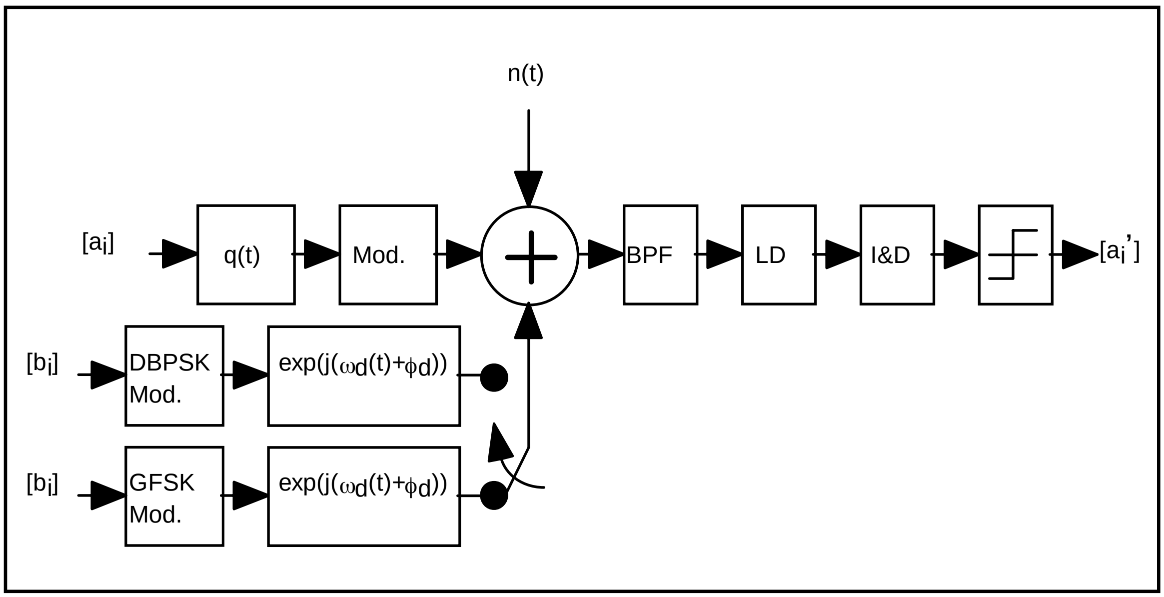

Figure C.5—IEEE 802.15.1 system model

C.4.1.1 The GFSK signal

The GFSK signal may be represented (by Murota [B8], Aulin [B1], and Steele [B16]) as

$$s(t, \alpha) = A\cos(2\pi f_c t + \phi(t, \alpha)) \tag{C8}$$

where

$$A = \sqrt{\frac{2E_b}{T}}$$

A is the peak amplitude,

E_b is the energy per data bit,
f_c is the carrier frequency,
α is the random input stream, comprised of the data bits α_i, and
$\phi(t,\alpha)$ is the output phase deviation, given by

$$\phi(t, \alpha) = 2\pi h_f \sum_{i = n-(L+1)}^{n} \alpha_i q(t - iT) + \pi h_f \sum_{i = -\infty}^{n-L} \alpha_i \tag{C9}$$

where

$$q(t) = \int_{-\infty}^{t} g(\tau) d\tau \text{, and}$$

$g(t)$ is the Gaussian-shaped pulse of the transmitter filter.

L is the length of $g(t)$, and it determines the number of consecutive symbol intervals required to transmit a single data bit. Sending a data bit over multiple symbols makes GFSK a partial response symbol, which reduces the required bandwidth. For IEEE 802.15.1 with BT= 0.5, L=2 means that a single data bit is spread over approximately two consecutive symbol intervals.

C.4.1.2 Interference model

Either a IEEE 802.15.1 or an IEEE 802.11b interference signal may be represented as

$$S_I(t, b) = B\cos(2\pi(f_c + f_d)t + \phi_2(t, b) + \phi_d) \tag{C10}$$

where

b is the random input data, which is independent of α,
ϕ_2 depends on the type of the interferer, and
f_d is the frequency difference between the desired signal and the interference.

The IEEE 802.15.1 radio channels are 1 MHz apart, so f_d may take values of 0,1,2,.... MHz. The bandwidth of the IEEE 802.11b system is 22 MHz, so the simulations used f_d <= 11 MHz. The sampling rate is N_s = 44 samples/bit, which equals 4 samples/chip for the IEEE 802.11 DSSS system. This sampling rate for f_d up to 22 MHz is appropriate.

A uniform random delay

t_d = [0,T)

and a random phase

$\phi_d \in [0, 2\pi)$

are applied to the interferer signal for each packet. It should be noted that the interference model is strictly concerned with the physical layer, and so it contains neither FEC nor retransmission protocols.

C.4.1.3 Limiter-discriminator with integrate and dump (LDI) receiver

This receiver consists of a pre-detection bandpass filter (BPF), a limiter-discriminator, and an integrate and dump (I&D) filter, as shown in Figure C.5. The final block is the hard limiter, which compares the output phase with a decision level. The pre-detection BPF is a Gaussian filter with an equivalent lowpass impulse response, $h_r(t)$, given by Simon [B10] as

$$h_r(t) = \sqrt{\frac{2\pi}{\ln 2}} B_r e^{-\left(\frac{2\pi^2}{\ln 2}\right)(B_r t)} \tag{C11}$$

where

B_r is the 3 dB bandwidth.

According to Simon and Wang [B10], the optimum bandwidth for this filter is

$$B_{IF} = 2B_r = \frac{1.1}{T} \tag{C12}$$

The discrete impulse response of this filter is obtained by sampling and truncating $h_r(t)$. The output of the receiver pre-detection filter may be represented using its inphase and quadrature components.

C.4.2 802.11b system model

The physical layer system models for 1, 2, 5.5, and 11 Mbit/s modes of the IEEE 802.11b standard are described in this subclause. The first rate is achieved by using differential BPSK (DBPSK) with DSSS and an 11 chip Barker code; the chip rate is 11 Mchip/s. The last rate is obtained using CCK, also at 11 Mchip/s. The communications system model for the 1 Mbit/s bit rate is presented in Figure C.6, again consisting of the transmitter, the channel, the receiver and the IEEE 802.15.1 interference source. The details of this model are explained in C.4.2.1 through C.4.2.4. The CCK system is shown in Figure C.8 and discussed in C.4.2.4.

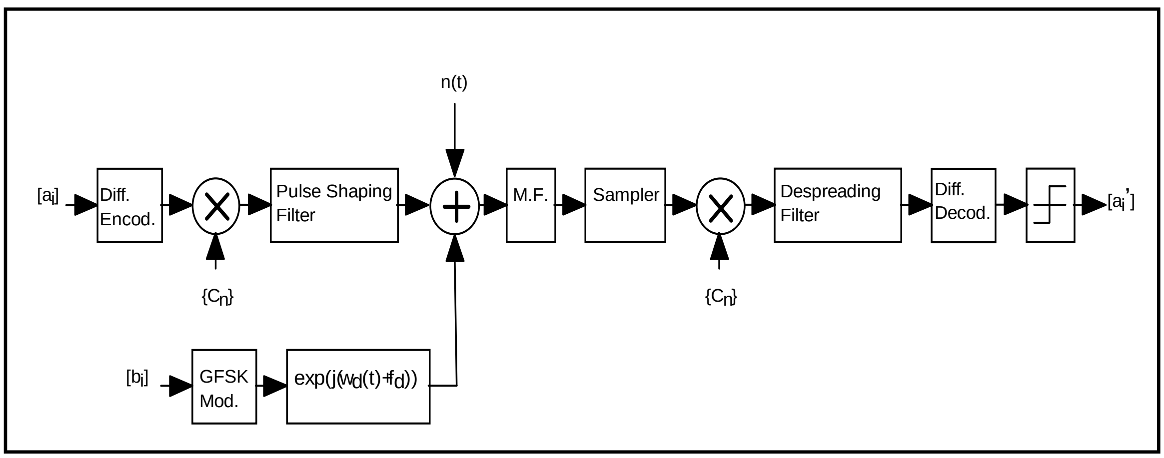

Figure C.6—802.11b DSSS system model

C.4.2.1 1 Mbit/s DSSS

The basic 1 Mbit/s rate is encoded using DBPSK. Thus, it is not necessary to have a coherent phase reference in the receiver to demodulate the received signal.

This system utilizes a spread spectrum scheme to mitigate the effect of interference. The Barker sequence with code length $P = 11$ is employed to spread the signal. The bit duration, T, is exactly 11 chip periods, T_c, long. The processing gain (PG) of this system is Proakis [B14]

$$PG = \frac{R_c}{R_b} = 11, \tag{C13}$$

where

$$R_b = \frac{1}{T} \text{ is the bit rate, and} \tag{C14}$$

$$R_c = \frac{1}{T_c} \text{ is the chip rate.} \tag{C15}$$

If the power spectrum of the Barker codes is calculated, then the following equation is the result: Lee [B13]

$$S(f) = \sum_{\substack{k = -\infty \\ k \neq 0}}^{\infty} \left(\frac{P+1}{P^2}\right) \operatorname{sinc}^2\left(\frac{k}{P}\right) \delta\left(f - \frac{k}{PT_c}\right) + \frac{1}{P^2}\delta(f) \tag{C16}$$

The function, $S(f)$, is illustrated in Figure C.7 for $P = 11$. As shown in Figure C.7, the Barker spreading code has a null at DC; thus, when IEEE 802.15.1 is exactly in the middle of the band, the despreading correlator will attenuate it. The result is that the middle of the spectrum will be attenuated more than an interferer located 1 MHz away.

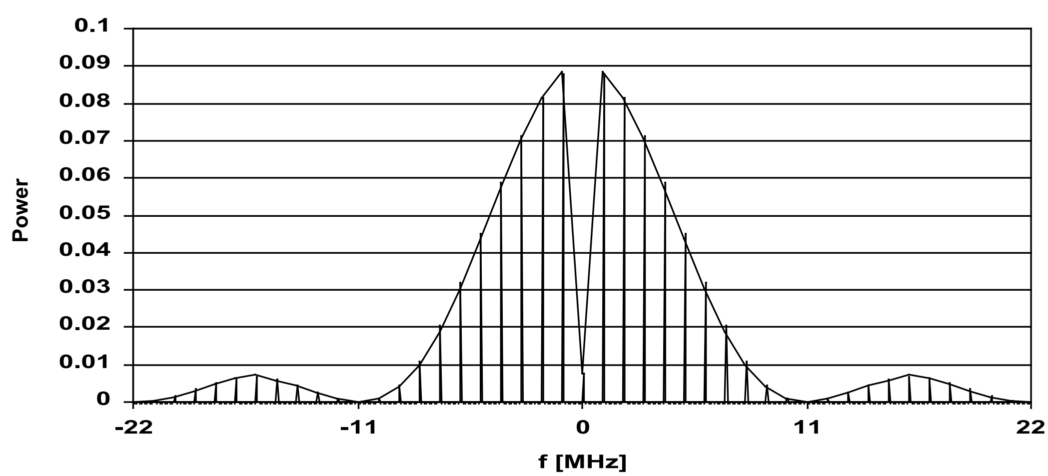

Figure C.7—Power spectrum of the Barker code

As shown in Figure C.6, the input data bits are first differentially encoded. The resulting sequence is spread by the Barker code. The output of the spreader is fed to a square-root raised-cosine pulse-shaping filter. The impulse response of this filter with a roll-off factor α may be found in Lee [B13]. The discrete time impulse response of this filter is obtained by sampling it.

At the receiver, the input samples are first passed through the square-root raised-cosine matched filter. The despreading filter is a rectangular filter that integrates the output of the multiplier during a bit period. The differential decoder compares the phase angle of the received symbol and the previous one to generate the output bit stream. It is assumed that the chip timing of the receiver is synchronized to the transmitter.

C.4.2.2 2 Mbit/s DSSS

The 2 Mbit/s rate employs DQPSK with the same Barker code as the 1 Mbit/s rate. The phase encoder is specified in Table C.7. The block diagram of the simulated system is the same as Figure C.6, except that the differential encoder and decoder are changed to DQPSK modulation.

Table C.7—Phase encoder for 2 Mbit/s DSSS

Dibit pattern (d0, d1)	Phase change (+jω)
00	0
01	π/2
11	π
10	3π/2

C.4.2.3 5.5 Mbit/s CCK

Complementary codes were originally conceived by Golay [B2] for infrared multi-slit spectrometry. The complementary codes in the IEEE 802.11b standards are defined by a set of 256 symbols. Each symbol has a duration of 8 chips. They are specified by

$$c = [e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_4)}, -e^{j(\phi_1+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_3)}, e^{j(\phi_1+\phi_3)}, -e^{j(\phi_1+\phi_4)}, e^{j\phi_1}] \quad \text{(C17)}$$

where

$$\phi_i \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\} \qquad for \qquad i = 1, 2, 3, 4$$

Note that each chip of a symbol is complex, and so may be transmitted using QPSK modulation as discussed below. The symbol rate is 11/8 Msymbol/s, giving 11 Mchip/s.

For 5.5 Mbit/s CCK, only 4 bits (*d0, d1, d2, d3*) are used to encode ϕ_1, ϕ_2, ϕ_3, and ϕ_4 and form an 8-chip CCK symbol. Specifically, the first dibit (*d0,d1*) encodes ϕ_1 based on DQPSK as in 11 Mbit/s CCK while the dibit (*d2*, *d3*) encode the basic symbol by setting

$$\phi_2 = (d2 \times \pi) + \frac{\pi}{2},$$

$$\phi_3 = 0,$$

and

$$\phi_4 = d3 \times \pi$$

Therefore, the transmitted CCK symbol may be rewritten as

$$c = [e^{j(\phi_1+\phi_2+\phi_4)}, e^{j(\phi_1+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_4)}, -e^{j(\phi_1+\phi_4)}, e^{j(\phi_1+\phi_2)}, e^{j\phi_1}, -e^{j(\phi_1+\phi_2)}, e^{j\phi_1}] \tag{C18}$$

Although 5.5 Mbit/s CCK has lower bit rate than 11 Mbit/s, its symbol rate and the chip rate remain the same at 11/8 Msymbol/s and 11 Mchip/s, respectively.

Because the complexity of the optimal decoder for CCK may be too high for practical implementation, the sub-optimal method may be used to decode the received phases with the following equations developed by Van Nee [B22].

$$\phi_2 = \arg\{r_1 r_2^* + r_3 r_4^* + r_5 r_6^* + r_7 r_8^*\} \tag{C19}$$

$$\phi_3 = \arg\{r_1 r_3^* + r_2 r_4^* + r_5 r_7^* + r_6 r_8^*\} \tag{C20}$$

$$\phi_4 = \arg\{r_1 r_5^* + r_2 r_6^* + r_3 r_7^* + r_4 r_8^*\} \tag{C21}$$

$$\phi_1 = \arg\{r_1 e^{-j(\phi_2+\phi_3+\phi_4)} + r_2 e^{-j(\phi_3+\phi_4)} + r_3 e^{-j(\phi_2+\phi_4)} + r_4 e^{-j\phi_4} + r_5 e^{-j(\phi_2+\phi_3)} + r_6 e^{-j\phi_3} + r_7 e^{-j\phi_2} + r_8\} \tag{C22}$$

where

$r = [r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8]$ is the received CCK symbol.

By setting $\phi_3 = 0$ and removing the terms with multiple phase estimates, ϕ_1 may be better estimated by

$$\phi_1 = \arg\{(r_2 + r_4)e^{-j\phi_4} + (r_5 + r_7)e^{-j\phi_2} + r_6 + r_8\} \tag{C23}$$

In this simulation, the sub-optimally coherent receiver with known initial phase to decode the received phases is used.

C.4.2.4 11 Mbit/s CCK

At 11 Mbit/s, 8 bits (*d0* to *d7*; *d0* first in time) are transmitted per symbol. The first dibit (*d0*, *d1*) encodes ϕ_1 based on DQPSK, which provides the possibility of employing differentially-coherent detection. Firstly a a coherent receiver is employed, assuming that the initial phase of the signal is known. The dibits, (*d2*, *d3*), (*d4*, *d5*), and (*d6*, *d7*) encode ϕ_2, ϕ_3 and ϕ_4, respectively, as specified in Table C.8 .

Table C.8—QPSK encoding

Dibit pattern (d_i,d_{i+1})	Phase
00	0
01	$\pi/2$
10	π
11	$3\pi/2$

The system model is presented in Figure C.8. Only an AWGN channel is considered in this case.

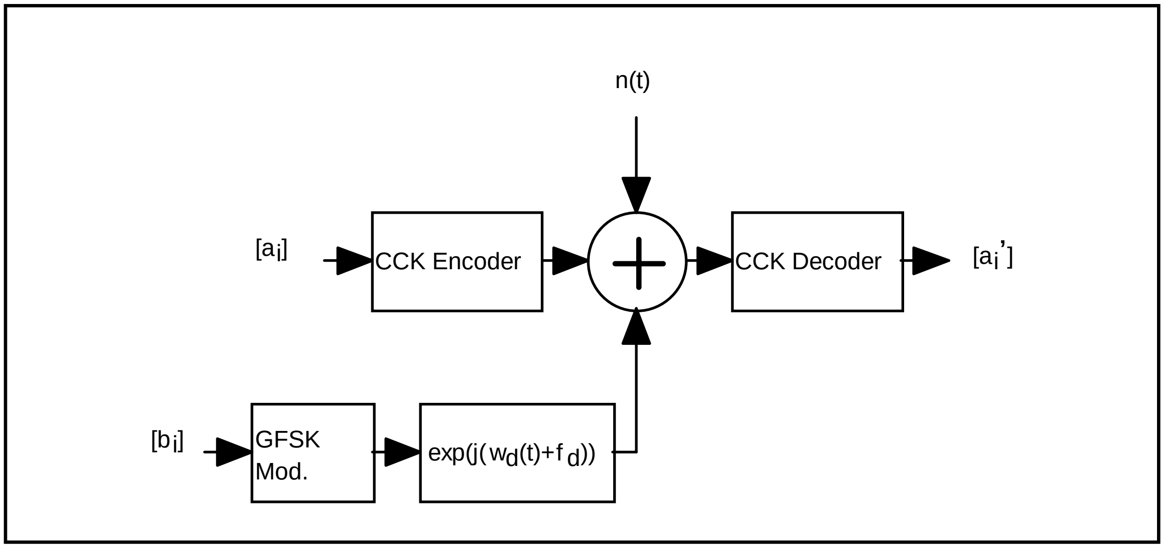

Figure C.8—CCK system model

A maximum likelihood decoder determines the valid symbol that is closest to the received symbol, and it maps that symbol back to eight data bits. This decoding method needs a bank of 256 correlators in the receiver. Although optimum, this method may be considered too complex for some implementations. There are also less complex sub-optimum algorithms. By looking at the code words of CCK, one may write these equations for the decoded phases as proposed by Van Nee [B22].

$$\phi_2 = \arg\{r_1 r_2^* + r_3 r_4^* + r_5 r_6^* + r_7 r_8^*\} \tag{C24}$$

$$\phi_3 = \arg\{r_1 r_3^* + r_2 r_4^* + r_5 r_7^* + r_6 r_8^*\} \tag{C25}$$

$$\phi_4 = \arg\{r_1 r_5^* + r_2 r_6^* + r_3 r_7^* + r_4 r_8^*\} \tag{C26}$$

$$\phi_1 = \arg\{r_4 e^{-(j\phi_4)} + r_6 e^{-(j\phi_3)} + r_7 e^{-(j\phi_2)} + r_8\} \tag{C27}$$

where

$r = [r_1\ r_2\ r_3\ r_4\ r_5\ r_6\ r_7\ r_8]$ is the received symbol.

The above sub-optimal receiver is employed to measure the performance in the presence of interference.

C.4.3 IEEE 802.11b in the presence of IEEE 802.15.1

The performance of the 1 Mbit/s IEEE 802.11b system, in an interference-limited environment with SNR = 35 dB, is given in Figure C.9. Both SNR and SIR are measured at the input to the receiver's BPF. The most disturbing interference is located at f_d = 1 MHz, which needs a minimum SIR of –5 dB. This difference stems from the null at the middle of the spectrum of the Barker code as described before. For frequency offsets greater than 8 MHz, the SIR value should be very low in order to get a high BER. This fact is due to the BPF in the IEEE 802.11b receiver having high attenuation at frequencies near 11 MHz. Further details of the simulation parameters, as well as some additional results, can be found in Soltanian [B20].

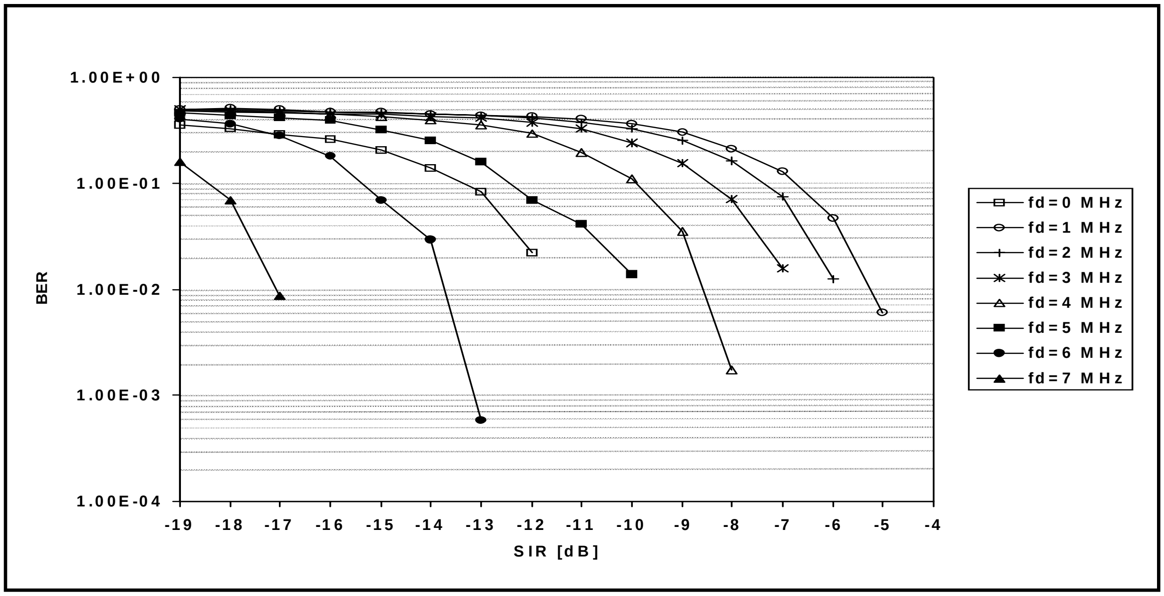

Figure C.9—1 Mbit/s 802.11b DSSS performance with IEEE 802.15.1 interference. AWGN channel. SNR=35 dB

Figure C.10 shows the performance of the 2 Mbit/s IEEE 802.11b system in the same environment. The use of DQPSK modulation doubles the bit rate, but makes the BER worse by about 5 dB. However, the packet transmission time is decreased due to the higher bit rate, and, therefore, the system performance may improve.

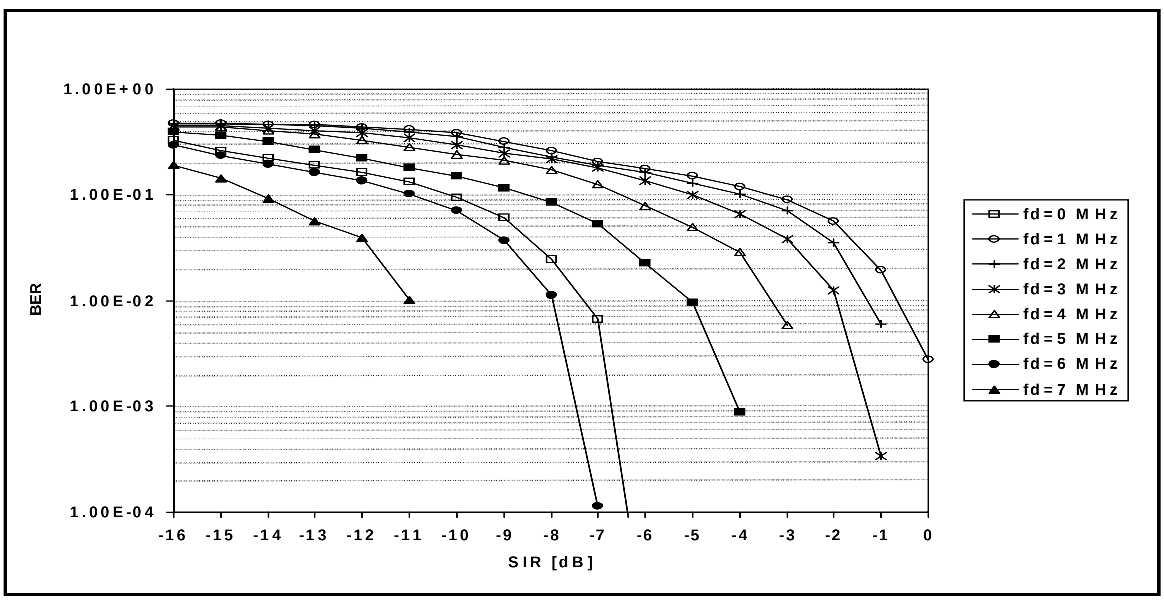

Figure C.10—2 Mbit/s 802.11b DSS performance with IEEE 802.15.1 interference. AWGN channel. SNR=35 dB

Figure C.11 shows the performance of the 5.5 Mbit/s IEEE 802.11b receiver with IEEE 802.15.1 interference in the AWGN channel. Unlike the case of the 1 Mbit/s IEEE 802.11b system, co-channel interference from IEEE 802.15.1 (f_d=0 MHz) significantly degrades the performance of the 5.5 Mbit/s IEEE 802.11b receiver. For frequency offsets greater than 2 MHz, an SIR of -1 dB is required to achieve the BER of 10^{-2}. In this case, the 5.5 Mbit/s IEEE 802.11b receiver achieves almost 3 dB improvement over the 11 Mbit/s IEEE 802.11b system.

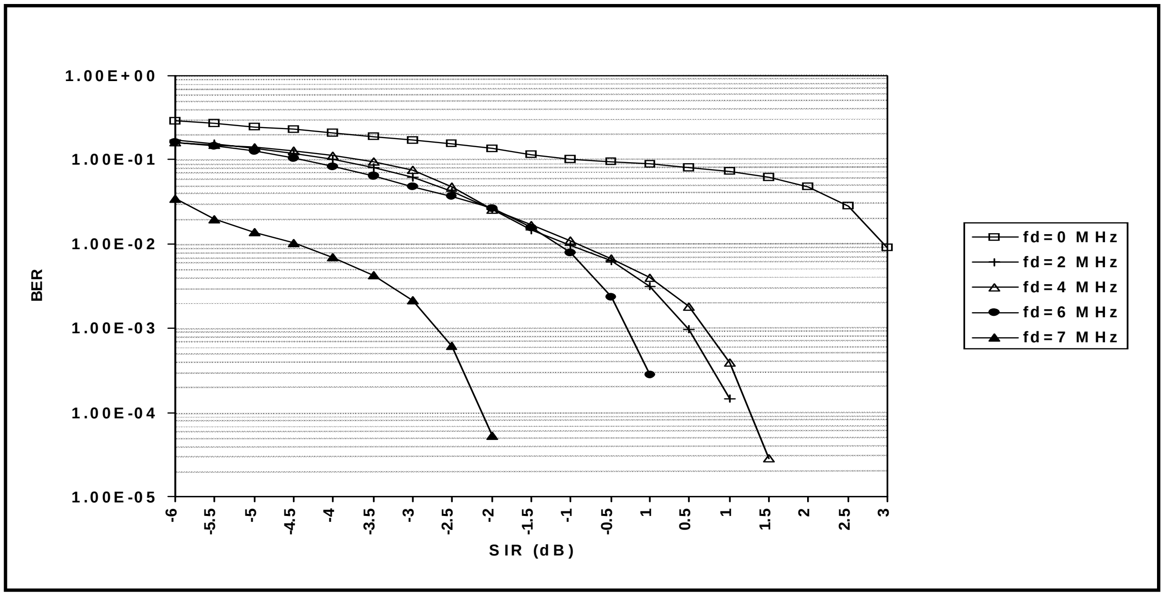

Figure C.11—5.5 Mbit/s 802.11b performance with IEEE 802.15.1 interference. SNR=35 dB

Figure C.12 illustrates the performance of the 11 Mbit/s IEEE 802.11b receiver with IEEE 802.15.1 interference. This figure indicates that the CCK modulation is more vulnerable to the interference signal than the 1 Mbit/s DSSS. A minimum SIR of 3 dB should be achieved to get BER=10^{-2} for all frequency offsets. This result is not surprising, because the CCK provides a higher bit rate but occupies the same 22-MHz bandwidth, thereby having less of a coding gain. Generally, the receivers used for both 1 Mbit/s and 11 Mbit/s are fairly simple, and improved performance may most likely be obtained using more sophisticated signal processing. This fact is especially true for the 11 Mbit/s CCK system.

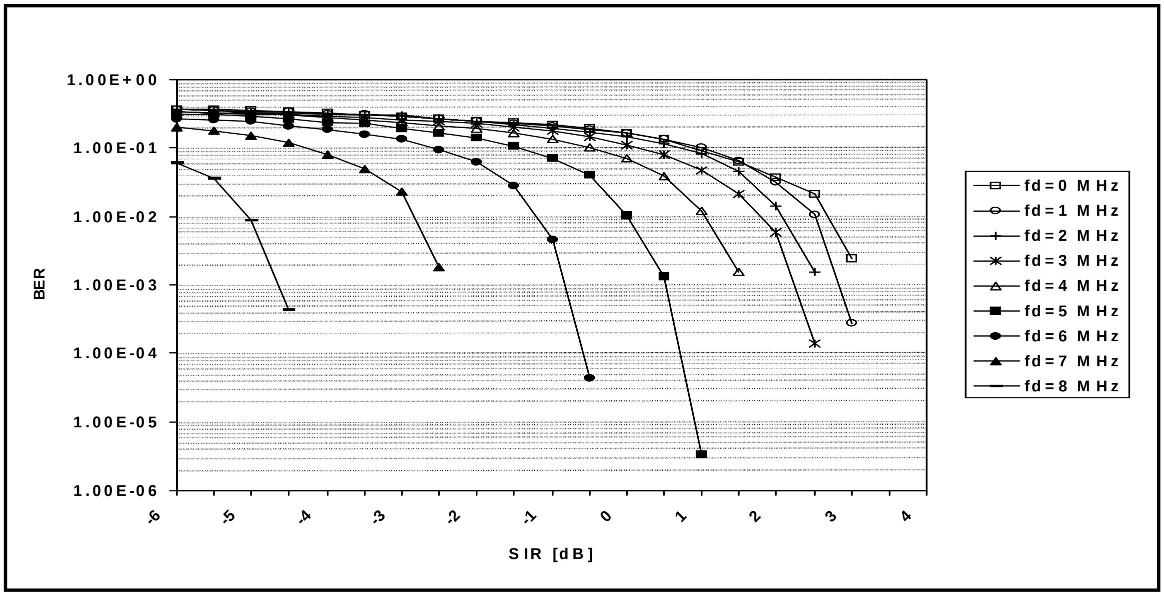

Figure C.12—11 Mbit/s 802.11b CCK performance with IEEE 802.15.1 interference. AWGN channel. SNR=35 dB

C.4.4 IEEE 802.15.1 in the presence of IEEE 802.11b

The LDI receiver design meets the IEEE 802.15.1 adjacent and co-channel interference specifications. While this model is not based on any particular implementation, it is meant to be indicative of a real implementation. Simulation results for the LDI receiver in the AWGN and Rician channels are presented in Soltanian [B20].

This subclause covers the performance of IEEE 802.15.1 with IEEE 802.11b interference. The SNR and SIR are measured at the input to the receiver's BPF. The curves in Figure C.13 are for an interference-limited environment with SNR = 30 dB. The IEEE 802.11b signal looks like broadband noise at the input to the IEEE 802.15.1 receiver. The performance degradation for carrier frequency differences up to 4 MHz is almost the same, and so the results for $f_d = 0$ is plotted as a representative case. The null in the Barker code spectrum does not improve the performance here, as it does for the IEEE 802.11b DSSS system. After 4 MHz, one gradually sees the effect of the pulse shaping filter of the IEEE 802.11b transmitter, which has a null at $f_d = 11$ MHz. In fact, the SIR value at $f_d = 11$ MHz has to be very low in order to cause high BER.

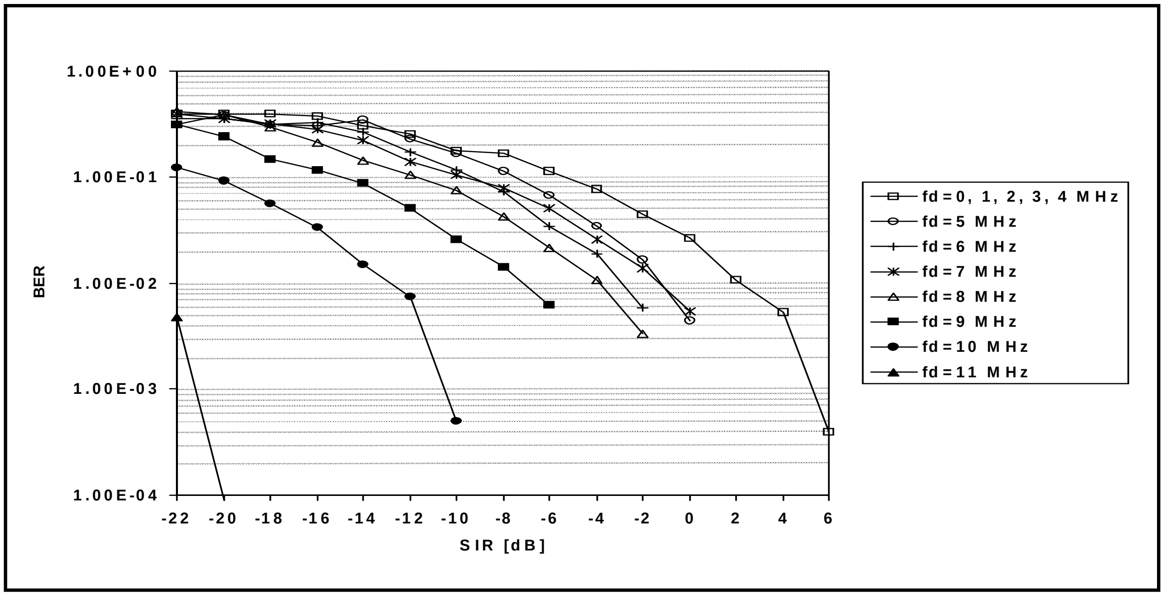

Figure C.13—IEEE 802.15.1 performance with 802.11b interference. AWGN channel. LDI receiver. SNR=30 dB

The roll-off factor α of the IEEE 802.11b transmitter determines the range of frequency offsets over which high BERs are observed. In this simulation, α=1 is chosen, so that the interference signal will occupy the maximum available bandwidth. Another observation from Figure C.13 is that if the SIR value is always greater than 6 dB, the BER for all frequency offsets is less than 10^{-3}. Note that as an interferer, the bit rate of the IEEE 802.11b physical layer is not important to the performance.

As a solution to mitigate the effect of interference, a simple two-state Viterbi receiver for IEEE 802.15.1 is used. The main problem with this Viterbi receiver is that it assumes that the modulation index is known. Unfortunately, the actual modulation index is allowed to vary over a large range. The phase of the transmitted signal is known to the receiver, and the SNR and SIR are measured at the input to the receiver's matched filter. The performance for IEEE 802.11 interference is shown in Figure C.14. A dramatic enhancement is observed in this figure, evidently at a cost of having a more complicated receiver.

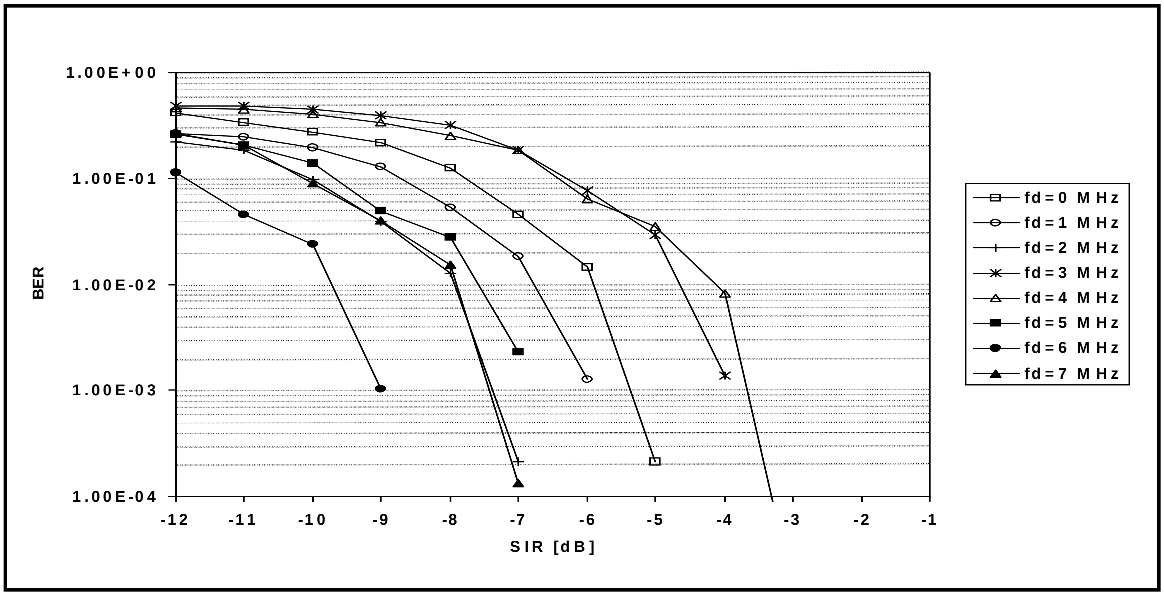

Figure C.14—IEEE 802.15.1 Viterbi receiver performance with 802.11b interference. AWGN channel. SNR=30 dB

Annex D

(informative)

Source code for the physical layer analytical model

This annex contains the source code for the physical analytical model.

```
/*---------------------------------------------------------------------------
 * Title:    Analytical Physical Layer Model for 802.15.2 BER Calculations
 * Authors:  Ron Nevo, Josie Ammer, Adrian Stephens
 *
 *
 * This module contains the analytical PHY-layer model used to calculate
 * BER values for 802.11b and 802.15.1 transmissions in the presence of
 * mutual interference.
 *---------------------------------------------------------------------------*/

/*-- Standard Includes ------------------------------------------------------*/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <limits.h>

/*-- Type Definitions -------------------------------------------------------*/


typedef enum
{
WPAN, WLAN11, WLAN55, WLAN1, WLAN2
// WPAN is non-FEC WPAN transmissions
// WLAN11 is 11Mbit/s 802.11
// WLAN55 is 5.5Mbit/s 802.11
// WLAN1 is 1Mbit/s 802.11
// WLAN2 is 2Mbit/s 802.11
} ModulationType;


struct Node
{
    double x,y;  // x and y positions in meters
};

typedef Node *aNodePtr;

struct Transmission
{
    aNodePtr src, dst;
    ModulationType type;
    float txpower;  // power in mW
    int frequency;
    // freq for WPAN is a number 1-79, is the center frequency
    // freq for WLAN is number 1-79, is the center frequency

    double BER;     // the resulting BER
};

typedef Transmission *aTransmissionPtr;
```

```
/*---------------------------------------------------------------------------
 * Function: CalculateAnalyticalBER
 *
 * Description:
 *             This function takes a list of transmissions and calculates
 *             the BER at each receiver using the analytical model.
 *
 * Parameters:  n       - the number of transmissions in the list
 *              tlist   - an array of Transmissions each corresponding to
 *                        an active transmission
 *
 * Returns: in every tlist element a calculated BER value
 *---------------------------------------------------------------------------*/
extern void CalculateAnalyticalBER(int n, Transmission tlist[]);



/*-- MACRO Definitions ------------------------------------------------------*/


#define WLAN_BANDWIDTH 22
#define HALF_WLAN_BANDWIDTH 11
#define PI acos(-1.0)



#define CCK_factor inverse_db(-8.0)
// 8dB gain for CCK coding


#define WLAN_SIR_perfect 10.0   // if SIR > 10dB, perfect reception
#define WLAN_SIR_impossible 0.1 // if SIR < -10dB, impossible to receive
#define WPAN_SIR_perfect 20.0   // if SIR > 13dB, perfect reception
#define WPAN_SIR_impossible 1.0 // if SIR < 0dB, impossible to receive


#define MIN_DISTANCE 0.1
// two nodes in the same spot act as if they are 0.1 apart

/*---------------------------------------------------------------------------*/

#define abs(a) ((a)>0 ? (a) : -(a))
#define sqr(x) ((x)*(x))
#define min(a,b) ((a)<(b) ? (a) : (b))




/*-- Local Functions --------------------------------------------------------*/
double WLAN_TxMask(int f)
{
        f = abs(f);
        if (f  <= 10)
        {
                return inverse_db(0.0);
        }
        else if (f  <= 21)
        {
                return inverse_db(-30.0);
        }
        else return inverse_db(-50.0);
}

/* Normalised Tx Mask */
double WLAN_NormTxMask(int f)
```

```
{
        int i;
        double sum = 0.0;
        for (i=-21; i<=21; i++)
        {
                sum += WLAN_TxMask(i);
        }
        return WLAN_TxMask(f) / sum;
}


double WLAN_RxMask(int f)
{
        f = abs(f);
        if (f  <= 10)
        {
                return inverse_db(0.0);
        }
        else if (f  == 11)
        {
                return inverse_db(-12.0);
        }
        else if (f  <= 20)
        {
                return inverse_db(-36.0);
        }
        else return inverse_db(-56.0);
}

/* -- WPAN MASKS -------------- */

double WPAN_TxMask(int f)
{
        f = abs(f);
        if (f  == 0)
        {
                return inverse_db(0.0);
        }
        else if (f  == 1)
        {
                return inverse_db(-20.0);
        }
        else if (f  == 2)
        {
                return inverse_db(-40.0);
        }
        else if (f  == 3)
        {
                return inverse_db(-60.0);
        }
        else return inverse_db(-80.0);
}

/* Normalised Tx Mask */
double WPAN_NormTxMask(int f)
{
        int i;
        double sum = 0.0;
        for (i=-3; i<=3; i++)
        {
                sum += WPAN_TxMask(i);
        }
        return WPAN_TxMask(f) / sum;
}
```

```
double WPAN_RxMask(int f)
{
        f = abs(f);
        if (f  == 0)
        {
                return inverse_db(0.0);
        }
        else if (f  == 1)
        {
                return inverse_db(-11.0);
        }
        else if (f  == 2)
        {
                return inverse_db(-41.0);
        }
        else return inverse_db(-51.0);
}




int isModulationTypeWPAN(ModulationType foo)
{
    switch (foo){
    case WPAN: return 1;
    default: return 0;
    }
}


int isModulationTypeWLAN(ModulationType foo)
{
    switch (foo){
    case WLAN11: return 1;
    case WLAN55: return 1;
    case WLAN1: return 1;
    case WLAN2: return 1;
    default: return 0;
    }
}

// compute db from real
double db(double x)
{
    return(10 * log10(x));
}

// compute real from db
double inverse_db(double x)
{
    return(pow(10.0, x /10));
}


// compute the Q function using approximation Q_5
double Q_5(double x)
{
    double x2,x3,x4,x5,x6;
    x2 = x*x;
    x3 = x2*x;
    x4 = x3*x;
    x5 = x4*x;
    x6 = x5*x;
```



```
    return(exp(-x2/2) * (x4+9*x2+8) /(x5+10*x3+15*x) / sqrt(2*PI));
}


// compute the codeword error probability of 802.11b 11Mbit/s
double SER11(double SIR)
{
    double res;
    res = 24*Q_5(sqrt(4*SIR)) +
        16*Q_5(sqrt(6*SIR)) +
        174*Q_5(sqrt(8*SIR)) +
        16*Q_5(sqrt(10*SIR)) +
        24*Q_5(sqrt(12*SIR)) +
        Q_5(sqrt(16*SIR));
    return(min(res,0.99999));
}

// compute bit error rate from Eb/No for 802.11b 11Mbit/s
double WLAN_BER_11(double SIR)
{
    if(SIR > WLAN_SIR_perfect)
        return 0;
    // if Eb/No more than some threshold, perfect reception
    else if(SIR < WLAN_SIR_impossible)
        return 0.5;
    // if Eb/No less than some threshold, impossible to receive
    else
        return((128.0/255.0)*SER11(SIR));
}


// compute the codeword error probability of 802.11b 5.5Mbit/s
double SER55(double SIR)
{
    double res;
    res = 14*Q_5(sqrt(8*SIR)) +
        Q_5(sqrt(16*SIR));
    return(min(res,0.99999));
}

// compute bit error rate from Eb/No for 802.11b 11Mbit/s
double WLAN_BER_55(double SIR)
{
    if(SIR > WLAN_SIR_perfect)
        return 0;
    // if Eb/No more than some threshold, perfect reception
    else if(SIR < WLAN_SIR_impossible)
        return 0.5;
    // if Eb/No less than some threshold, impossible to receive
    else
        return((4.0/7.0) * SER55(SIR));
}


// compute the function number of choice of k elements from n
int choose(int k,int n)
{
    int i;
    int res = 1;
    for(i=n;i>n-k;i--)
        res *= i;
    for(i=1;i<=k;i++)
        res /= i;
    return(i);
}
```

```
// compute the BER for WLAN 1Mbit/s, the BER is Q(sqrt(11*2*SIR/2))
double WLAN_BER_1(double SIR)
{
    if(SIR > WLAN_SIR_perfect)
        return 0;
    // if Eb/No more than some threshold, perfect reception
    else if(SIR < WLAN_SIR_impossible)
        return 0.5;
    // if Eb/No less than some threshold, impossible to receive
    else
        return(min(Q_5(sqrt(11*2*SIR/2)),0.5));
}

// compute the BER for WLAN 2Mbit/s, the BER is Q(sqrt(5.5*2*SIR/2))
double WLAN_BER_2(double SIR)
{
    if(SIR > WLAN_SIR_perfect)
        return 0;
    // if Eb/No more than some threshold, perfect reception
    else if(SIR < WLAN_SIR_impossible)
        return 0.5;
    // if Eb/No less than some threshold, impossible to receive
    else
        return(min(Q_5(sqrt(5.5*2*SIR/2)),0.5));
}

// compute the BER for WPAN
double WPAN_BER(double SIR)
{
    if(SIR > 20)
        return 0;
    // if Eb/No more than 13dB, perfect reception
    else if(SIR < 1)
        return 0.5;
    // if Eb/No less than 0dB, impossible to receive
    else
        return(min(exp(-SIR/2),0.5));
}



double Distance(Transmission &Src, Transmission &Dest)
{
    return(sqrt(sqr(Src.src->x-Dest.dst->x) + sqr(Src.src->y-Dest.dst->y)));
}

// power as function of distance
double PowerDistance(Transmission &Src, Transmission &Dest)
{
    double power_d0,dist;

    power_d0 = Src.txpower;

       dist=Distance(Src,Dest); // calc distance function
    if (dist < MIN_DISTANCE)
        dist = MIN_DISTANCE;

    if(dist < 8)  // use 40.2+20log d for <8M power loss
        return(power_d0/(pow(dist,2.0) * pow(10.0, 4.02)));
    else         // use 58.5 + 33log(d/8) for >8M power loss
        return(power_d0/(pow(dist/8.0,3.3) * pow(10.0, 5.85)));
}
```

```
// Calculate effect of transmit mask and offset receiver mask
// The result can be multiplied by the Transmitted power to
// Obtain the power at the detector
double SpectrumFactor(Transmission &Src, Transmission &Dest)
{
        double spectrumFactor = 0;
        int f;

        // freq_dif is the difference in center frequencies
        // of the transmitter and receiver
    int freq_dif = abs(Src.frequency - Dest.frequency);

        if (isModulationTypeWPAN(Src.type) == isModulationTypeWPAN(Dest.type))
        {
                // In the interests of brevity, this model only supports
                // WLAN/WPAN and WPAN/WLAN calculations.
                // The extensions to remove this limitation are straightforward.
                assert (0);    // Unsupported
        }

    if(isModulationTypeWPAN(Dest.type))
        {
                for (f = -40; f <= 40; f++)
                // Note, the bounds are unimportant, provided they are big enough
                {
    spectrumFactor += WPAN_RxMask(f-freq_dif) *
                       WLAN_NormTxMask(f);
                }
        }
        else
        {
                for (f = -40; f <= 40; f++)
                {
                        spectrumFactor += WLAN_RxMask(f-freq_dif) *
                                       WPAN_NormTxMask(f);
                }
        }

        return spectrumFactor;
}




/*-- Global Function ---------------------------------------------------------*/

void CalculateAnalyticalBER(int n, Transmission tlist[])
{// n should be the length of tlist
    double SIR;
    for (int dst= 0; dst < n ; dst++) { //for each dest
        double signal=0.0, interference=0.0;

        for (int src = 0; src < n; src++) { //calculate the power from each source
            double power;

                        power = PowerDistance(tlist[src],tlist[dst]) *
                                        SpectrumFactor(tlist[src],tlist[dst]);

            if (src==dst)
                // if src and dest are from the same transmission pair,
                // pwr is signal power
                signal = power;
            else
                // if not from the same transmission pair,
```

```
                // pwr is interference power
                interference += power;
        }
        SIR = signal/interference; // calculate the SNR for each dest

        //need to calc BER from SIR
        double ber0;
        switch (tlist[dst].type) {
            case WPAN:
                ber0 = WPAN_BER(SIR);
                break;
            case WLAN11:
                ber0 = WLAN_BER_11(SIR);
                break;
            case WLAN55:
                ber0 = WLAN_BER_55(SIR);
                break;
            case WLAN1:
                ber0 = WLAN_BER_1(SIR);
                break;
            case WLAN2:
                ber0 = WLAN_BER_2(SIR);
                break;
            default:
                printf("Unknown ModulationType");
        }

        tlist[dst].BER = ber0;
    }
}
```


Annex E

(informative)

Medium access control (MAC) sublayer models

The OPNET Modeler, a network technology development environment, was used to develop a simulation model for the IEEE 802.15.1 and IEEE 802.11 protocols. The IEEE 802.11 model available in the OPNET library was extended to interface to the channel and physical layer models described in Annex C.

For the IEEE 802.15.1 protocol, the baseband and logical link control and adaptation protocol (L2CAP) layers were partially implemented. The assumption was made that a connection is already established between the master and the slave and that the synchronization process is complete. The connection type is either SCO for voice or ACL for data traffic.

A MAC protocol generally consists of a collection of components, each performing a special function, such as the support of higher layer traffic, the synchronization process, the bandwidth allocation, and the contention resolution mechanism.

In this annex, the features that are the most relevant to the interference evaluation are described. They are a description of the MAC state machine, the FH, the error detection and correction schemes, and the interface to the physical layer.

E.1 MAC state machine

Each of the IEEE 802.15.1 and IEEE 802.11 MAC protocols is implemented as a state machine. Transitions from one state to another are generally triggered by the occurrence of events such as the reception or transmission of packets. Higher layer message arrivals require packet encapsulation and often segmentation if the message is too long. The information available in the packet determines the type of packet processing and encapsulation required. For example, IEEE 802.15.1 ACL connections require L2CAP encapsulation while SCO connections only require baseband encapsulation. The packet is then enqueued and awaits a transmission opportunity. Because SCO packets need to be transmitted at fixed intervals, IEEE 802.15.1 SCO packets have priority over IEEE 802.15.1 ACL packets.

Transmission of packets follows each protocol’s rules. IEEE 802.15.1 transmission is based on a polling mechanism where the master controls the usage of the medium including its own transmission. In order to model the slotted nature of the channel, a virtual clock is implemented that generates self-interrupts every 625 µs. A master device starts its transmission in an odd-numbered slot, while an even-numbered slot is reserved for a slave transmission.

On the other hand, the IEEE 802.11 protocol uses CSMA/CA defined in the DCF operation that allows a station to access the medium if the station is not receiving a packet or waiting for an acknowledgement from a previous transmission, after the medium has been idle for a period of time.

E.2 Frequency-hopping

The IEEE 802.15.1 (i.e., Bluetooth®) hopping pattern algorithm is implemented. Details of the algorithm are as follows.

Given a window of 32 frequencies in the 2.402-2.483 GHz range, a sequence of 32 frequencies is chosen randomly. Once all 32 frequencies in that set have been visited once, a new window of 32 frequencies is selected. This new window includes 16 of the frequencies previously visited and 16 new frequencies. A pseudo-random number generator is used instead of the implementation specific circuitry that uses the master's clock and 48-bit address to derive a random number.

Similarly, the IEEE 802.11 pseudorandom hopping pattern is implemented according to the base-hopping sequence defined for North America in the IEEE 802.11 specifications. The time spent on each frequency is set to a packet transmission time which depends on the simulation scenario used.

In the IEEE 802.11 Direct Sequence model, 14 channels are defined according to the IEEE 802.11 specifications DSSS PHY frequency channel plan, however for most cases and for this recommended practice, only 11 channels are used. The center frequency parameter is set to channel 6 (2.437 GHz) in the simulation results.

E.3 Error detection and correction

Error detection and correction is an essential component in the interference study.

For IEEE 802.15.1, the device first applies the error correction algorithm corresponding to the packet encapsulation used. HV1 packets have a fixed packet length of 366 bits including a header and an access code of 126 bits; they use a payload of 80 information bits, a 1/3 FEC rate and are sent every T_SCO=2 slots or 1250 µs. In case of an error occurrence in the payload, the packet is never dropped. A 1/3 FEC, as specified in the IEEE 802.15.1, is applied to the packet header while a code with a Hamming distance (d=14) is applied to the access code. Uncorrected errors in the header and access code lead to a packet drop.

On the other hand, DM5 packets use a 2/3 rate FEC to correct payload. Errors in the header or access code are corrected/detected by a 1/3 FEC and a correction code with a Hamming distance (d=14), respectively. Uncorrected errors lead to dropping packets and the use of the ARQ scheme.

For IEEE 802.11, errors are detected by checking the frame check sequence (FCS) that is appended to the packet payload. In case an error is found, the packet is dropped and is then later retransmitted. Otherwise, a positive acknowledgement packet (ACK) notifies the source of a correct reception as specified by the IEEE 802.11 standard. Note that not all 802.11 packets require an acknowledgement.

E.4 Interface to physical layer

The MAC models are interfaced to the simulated PHY layer models described in Annex D in order to simulate the overall system. The step-by-step simulation process works as follows. Traffic is generated by sources located above the MAC sublayer. The message is then passed to the MAC sublayer where it undergoes encapsulation and obeys the MAC transmission rules. The packet is then sent to an interface module before it is passed to the PHY layer.

This interface module is required to capture all changes in the channel state (mainly in the energy level) while a packet is transmitted. At the end of each packet transmission, a list is generated consisting of all interfering packets, collision duration, timing offset, frequency, power and the topology of the scenario used. This list is then passed to the physical layer module along with a stream of bits representing the packet being transmitted. The physical layer returns the bit stream after placing the errors resulting from the interference as shown in Figure E.1.

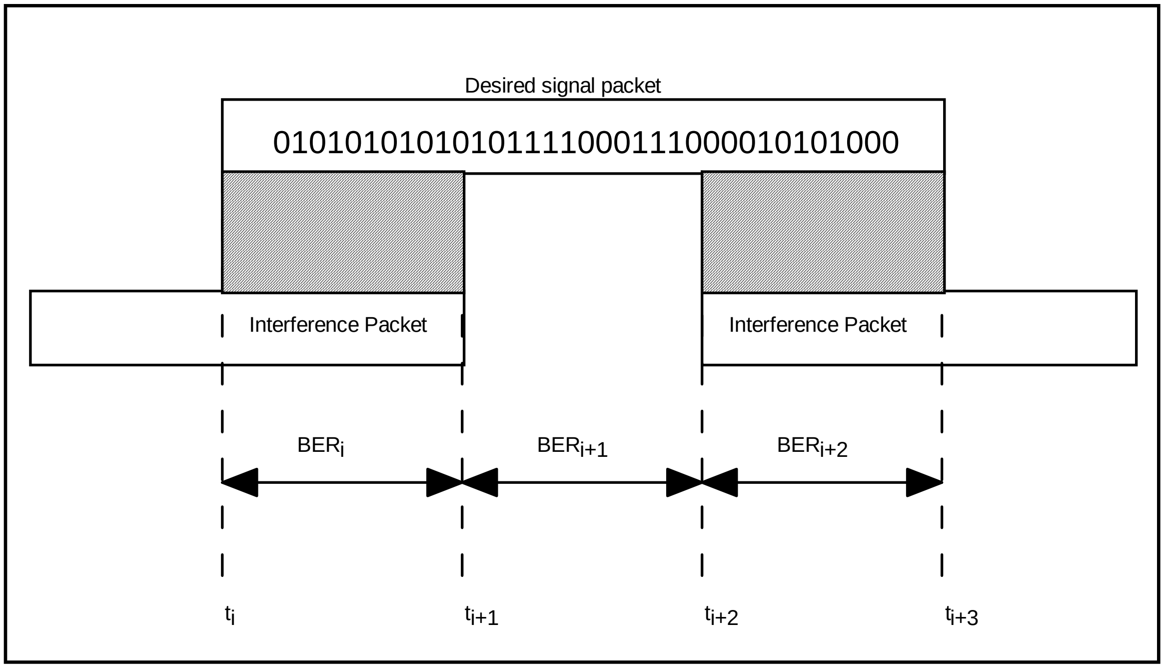

Figure E.1—MAC/PHY interface

NOTE–Each bit is corrupted according to the receiver's performance given the SIR computed from the collision information.

Annex F

(informative)

Data traffic models

For IEEE 802.15.1, two types of applications, namely voice and data traffic, are considered. For voice, it is assumed that a symmetric stream of 64 kbit/s each way using HV1 packet encapsulation is used. For data, DH5 packets are used. The packet inter arrival time is exponentially distributed, and its mean in seconds is computed according to:

$$t_b = 2 \times N \times (T_s / L) \tag{F1}$$

where

L is the offered load,
N is the number of slots occupied by a packet. For DH5, N=5, and
T_s is the slot size equal to 625 μs.

For the WLAN, the packet payload is fixed to 12,000 bits and L is varied. The packet inter arrival time in seconds, t_w is exponentially distributed, and its mean is computed according to

$$t_w = ((192 / (1000000)) + (12224 / \text{payload\_data\_rate})) / (L)$$

where the 192-bit PLCP header is sent at 1 Mbit/s and the payload_data_rate is either 1 or 11 Mbit/s.

Annex G

(informative)

Performance metrics for IEEE 802.15.1

At the MAC sublayer, a set of performance metrics is defined to include access delay, PER, and residual number of errors in the IEEE 802.15.1 voice packets. The access delay measures the time it takes to transmit a packet from the time it is passed to the MAC sublayer until it is successfully received at the destination. The access delay for the IEEE 802.15.1 LAN traffic is measured at the L2CAP layer in order to account for retransmission delays. PER measures the number of packets discarded at the MAC sublayer due to errors in the bit stream. This measure is calculated after performing error correction.

The residual number of errors in the IEEE 802.15.1 voice packets measures the number of errors that remain in the packet payload after error correction is performed.

Annex H

(informative)

Coexistence modeling results

This annex describes results of simulations that evaluate the performance of WPAN (i.e., IEEE 802.15.1) in the presence of WLAN (i.e., IEEE 802.11b) interference and vice versa. A simple 4-node topology is chosen in order to better identify the interference problem and the parameters effecting it. This is the simplest topology required that could lead to interference. A mix of data and voice traffic is chosen as a representative set of applications running on IEEE 802.11 and IEEE 802.15.1.

The configuration and system parameters used are shown in Table H.1.

Table H.1—Simulation parameters

Simulation parameters	Values
Length of simulation run	30 seconds
IEEE 802.15.1 parameters	
Data packet inter arrival time	12.5 milliseconds
Data offered load	50%
ACL baseband packet encapsulation	DM5
SCO baseband packet encapsulation	HV1
Transmitted power	1 mW
Slave coordinates	(0,0) meters
Master coordinates	(1,0) meters
WLAN parameters	
Packet inter arrival time for 1 Mbit/s	24.8 ms
Packet inter arrival time for 11 Mbit/s	2.6 ms
Offered load	50%
Transmitted power	25 mW
AP coordinates	(0,15) meters
Mobile coordinates	(0,d) meters
Packet header	224 bits

Results from four different simulation experiments that show the impact of WLAN interference on IEEE Std 802.15.1 devices and vice versa for two different applications, namely voice and data traffic, are presented.

Table H.2 provides a summary of these four cases, while Figure H.1 shows the experimental topology.

Table H.2—Summary of the experiments

Experiment	Desired signal	Interferer signal	WLAN AP	WLAN mobile
1	IEEE Std 802.15.1 voice	IEEE Std 802.11	sink	source
2	IEEE Std 802.15.1 data	IEEE Std 802.11	sink	source
3	IEEE Std 802.11	IEEE Std 802.15.1 voice	source	sink
4	IEEE Std 802.11	IEEE Std 802.15.1 data	source	sink

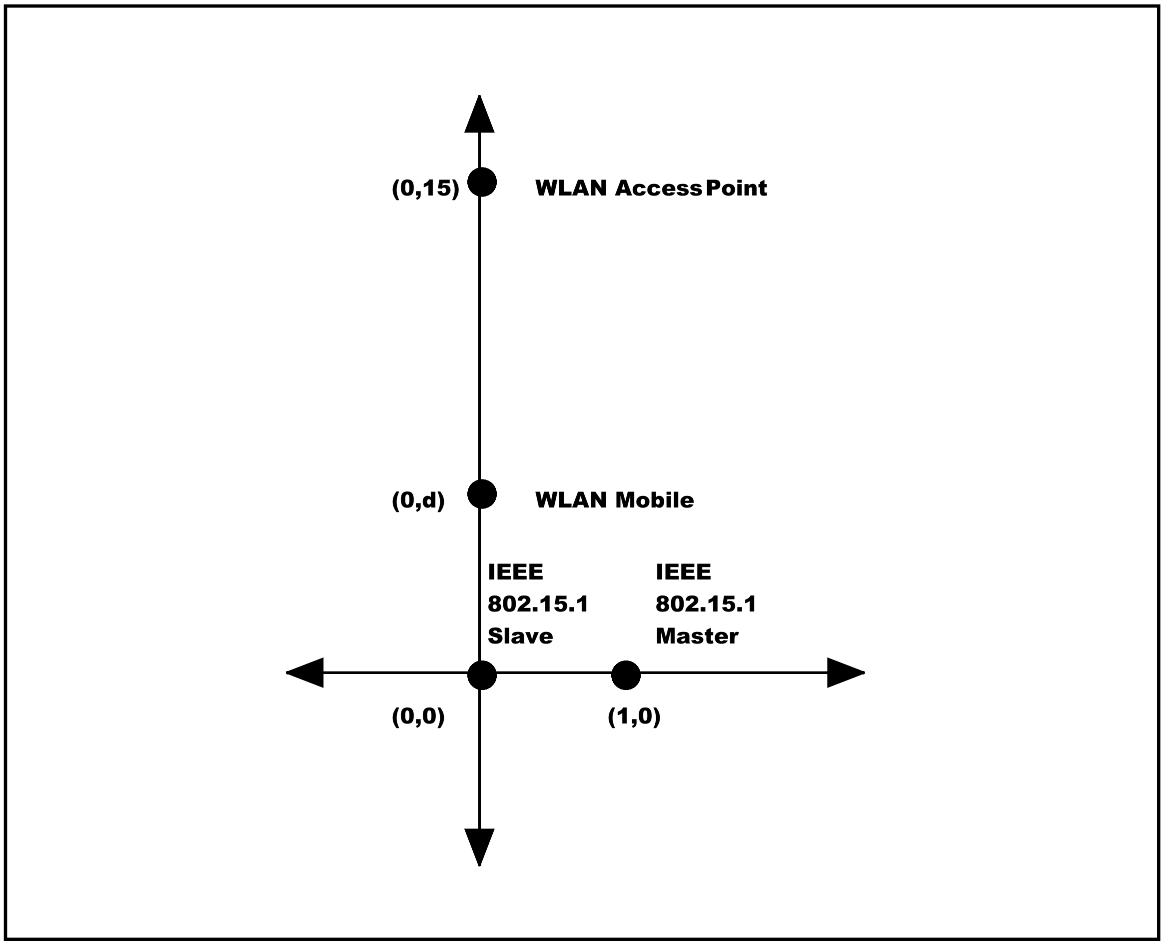

Figure H.1—Experiment topology

Please note that the WLAN AP is fixed at (0,15) meters, while the WLAN mobile is free to move along the vertical axis, i.e. its coordinates are (0,*d*). The IEEE 802.15.1 devices are fixed at the given locations. In the first two experiments, the mobile is the generator of the IEEE 802.11 data, while the AP is the sink. In the last two experiments the traffic is generated at the AP.

All four experiments are repeated for IEEE 802.11 1 Mbit/s and 11 Mbit/s DSSS and 1 Mbit/s FH systems using the simulated physical layer curves from C.4. All simulations are run for 30 seconds of simulated time. The performance measurements are logged at the slave device for IEEE 802.15.1 and at the AP and Mobile devices for WLAN.

H.1 802.11 1 Mbit/s direct sequence and IEEE 802.15.1 interference

Figure H.2 depicts the PER for experiments 1 and 2 where the IEEE 802.15.1 piconet is closer to the WLAN source.

The packet error rate for both IEEE 802.15.1 voice (experiment 1) and data (experiment 2) is 13% at 0.5 meter. The PER drops gradually for IEEE 802.15.1 voice for distances greater than 2 meters. However, it remains at 7% for IEEE 802.15.1 data when the WLAN source is 5 m away.

The PER for the WLAN corresponds to the loss of ACK messages at the WLAN mobile device. Observe a WLAN PER of 18% in experiment 1 where IEEE 802.15.1 voice is the interferer, as opposed to 12% in experiment 2 where IEEE 802.15.1 data is the interferer signal.

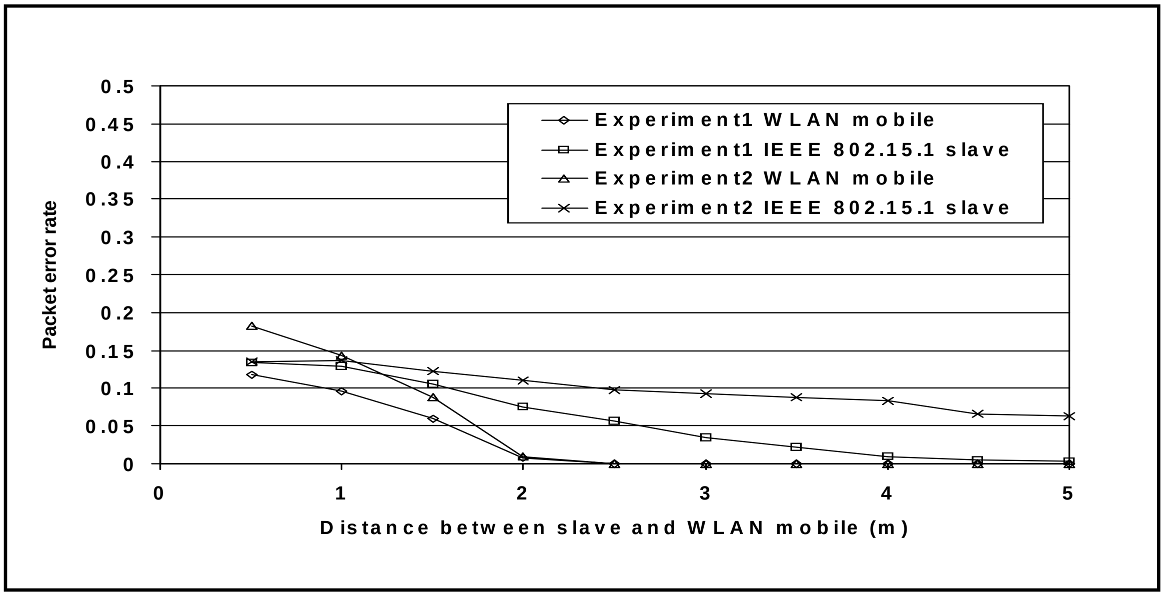

Figure H.2—PER for experiments 1 and 2—1 Mbit/s direct sequence

The access delay curves given in Figure H.3 closely follow the PER trends described in Figure H.2. The delay for WLAN (observed at the sink) is around 23 ms for distances less than 2 m, and drops to 19 ms beyond 2 m where the PER is zero.

For IEEE 802.15.1 data the delay curve remains at 7 ms between 0.5 meter and 5 m because the PER is still high at 5 m.

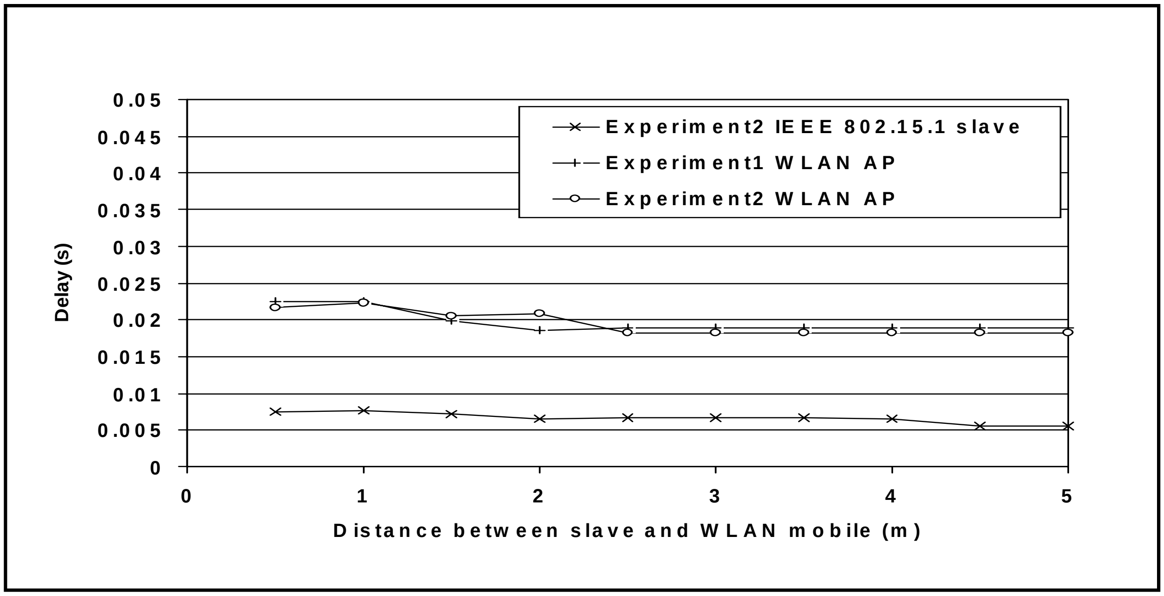

Figure H.3—Access delay for experiments 1 and 2—1 Mbit/s direct sequence

Figure H.4 shows the PER for experiments 3 and 4. Note that the PER when the WLAN receiver is close to a IEEE 802.15.1 voice connection (95%) is double that when it is close to a IEEE 802.15.1 data connection (45%).

The PER for IEEE 802.15.1 is negligible in this case because the WLAN source is far from the IEEE 802.15.1 piconet (15 m) and does not effect the receiver.

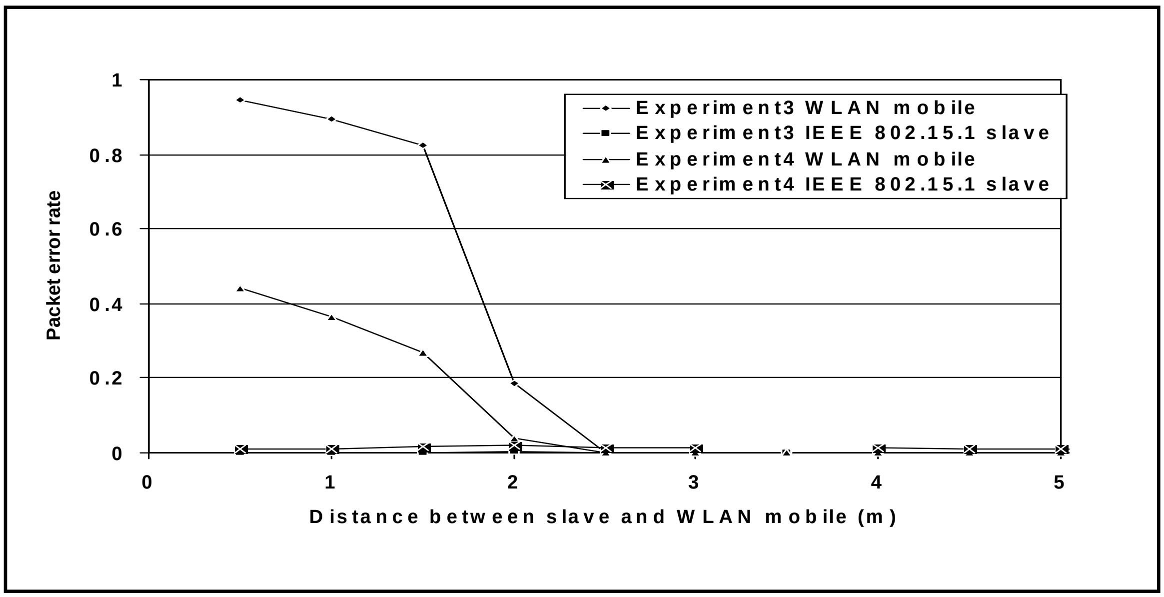

Figure H.4—PER for experiments 3 and 4—1 Mbit/s direct sequence

Figure H.5 shows the delay curves for experiments 3 and 4. Note that delays generally follow the PER trends. The peak observed at 1.5 m for experiment 3 between 0.5 meter and 2 m is probably due to the artifact of how delays are computed. When packets are dropped at the receiver, no delays are recorded. It is only when packets are successfully received that their access delay is recorded. At the end of the simulation, delays are added and an average is computed. As explained in the text, when the packet loss is extremely high the delay average is computed over a much small number of packets (only those that make it to the receiver). It may appear that the delay is lower, but only because packets are dropped. Note that after 7 attempts, the packets are dropped at the sender. The delay at 1.5 m is 150 ms, that is an order of magnitude greater than the delay at 2 m, which is around 18 ms.

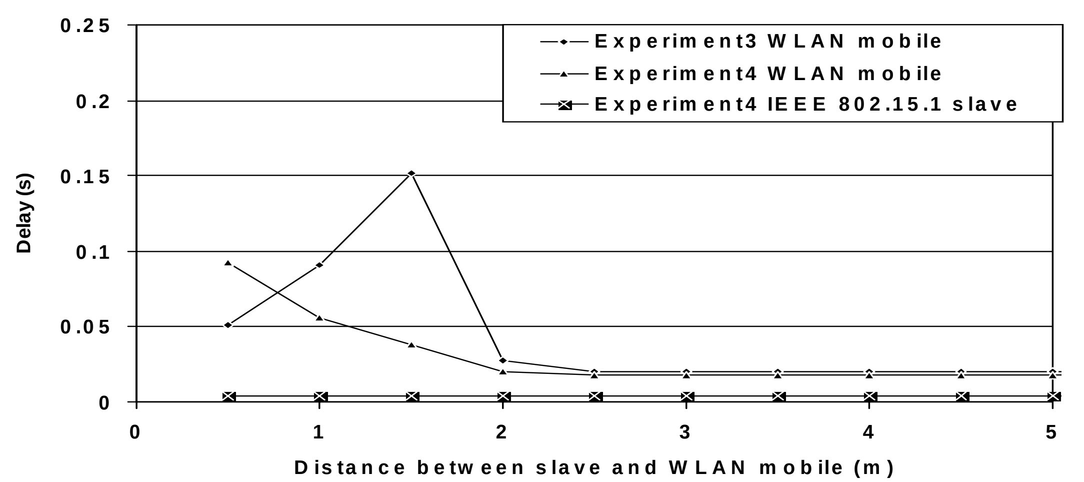

Figure H.5—Access delay for experiments 3 and 4—1 Mbit/s direct sequence

H.2 802.11 11 Mbit/s direct sequence and IEEE 802.15.1 interference

Figure H.6 shows the PER for experiments 1 and 2. The effect of the WLAN 11 Mbit/s interference on IEEE 802.15.1 leads to slightly higher PER (20%) for IEEE 802.15.1 data compared with the 1 Mbit/s WLAN interference (13% in Figure H.2). The PER for the IEEE 802.15.1 voice is comparable to the results obtained with the WLAN 1 Mbit/s interference. The 11 Mbit/s WLAN ACK error rate is also comparable to the 1 Mbit/s WLAN ACK rate obtained in Figure H.2 because the ACK packet is always sent at 1 Mbit/s.

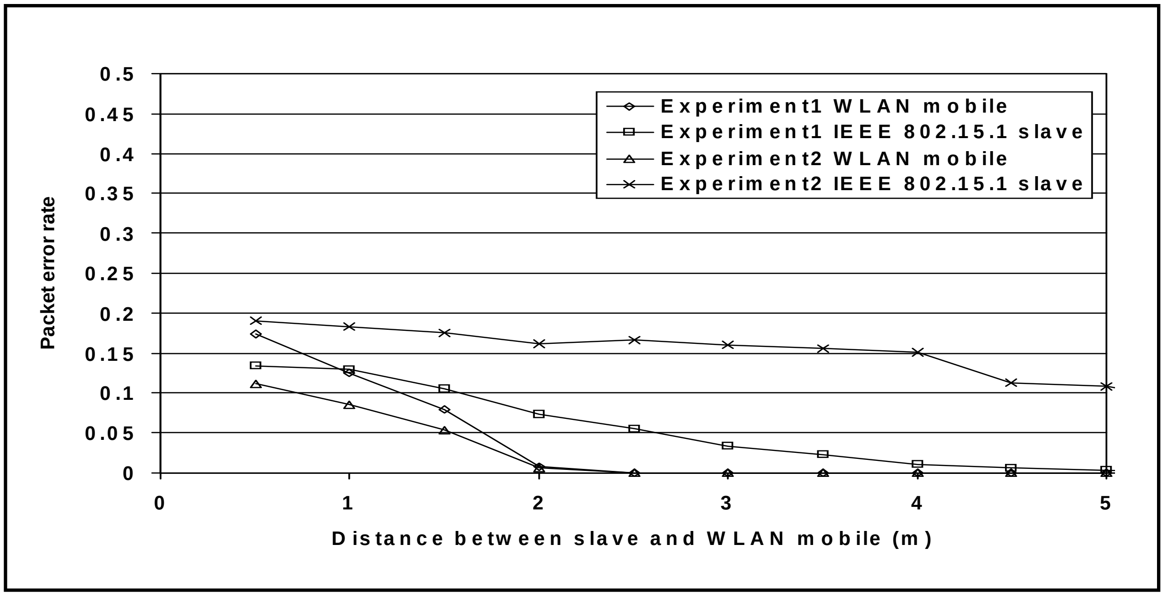

Figure H.6—PER for experiments 1 and 2—11 Mbit/s direct sequence

Figure H.7 depicts the delay for experiments 1 and 2. The delay for the IEEE 802.15.1 data connection starts at 12 ms for a distance of 0.5 meter and drops to 7 ms for a distance of 5 m.

The delay for the WLAN in experiments 1 and 2 start at 20 ms and 13 ms respectively at a distance of 0.5 meter and converge to 5 ms beyond 2 meters.

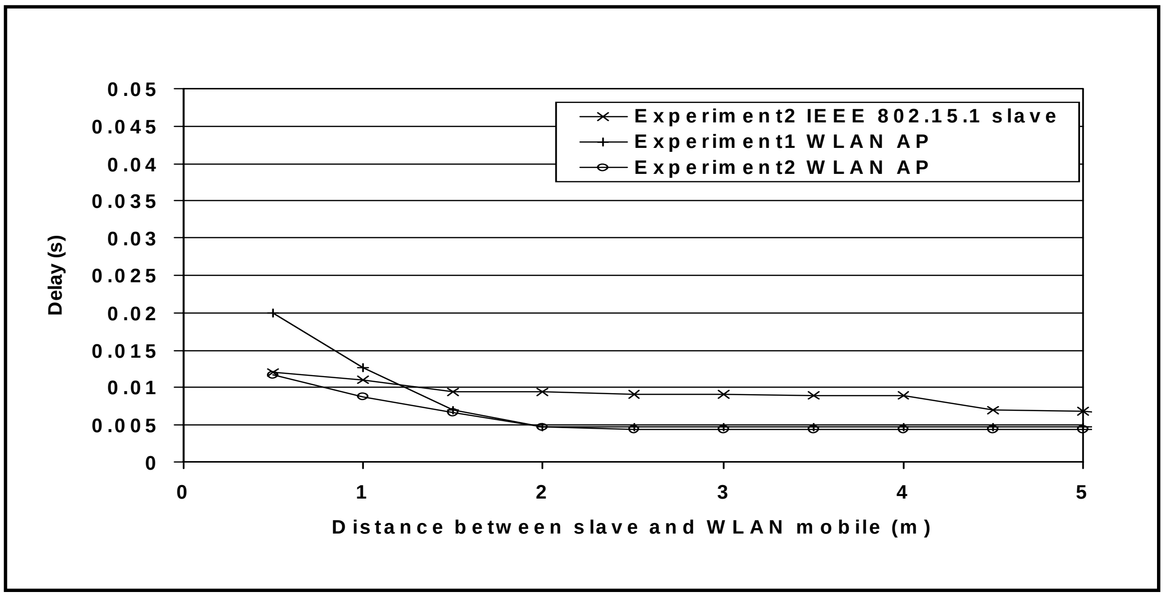

Figure H.7—Access delay for experiments 1 and 2—11 Mbit/s direct sequence

Figure H.8 shows the PER for experiments 3 and 4. Note that the PER for the 11 Mbit/s WLAN direct sequence is half the PER for the 1 Mbit/s WLAN direct sequence at 0.5 meter for experiment 3 (Figure H.4). However, unlike the sharp drop in PER observed for the 1 Mbit/s WLAN for distances beyond 2 m, the PER for the 11 Mbit/s WLAN remains greater than 25% until a distance of 4 meters. This is due to the robustness of the Barker code used in the 1 Mbit/s WLAN as opposed to the CCK used in the 11 Mbit/s WLAN.

The same applies to experiment 4. The PER observed for the 11 Mbit/s WLAN is also about half the PER obtained for the 1 Mbit/s WLAN.

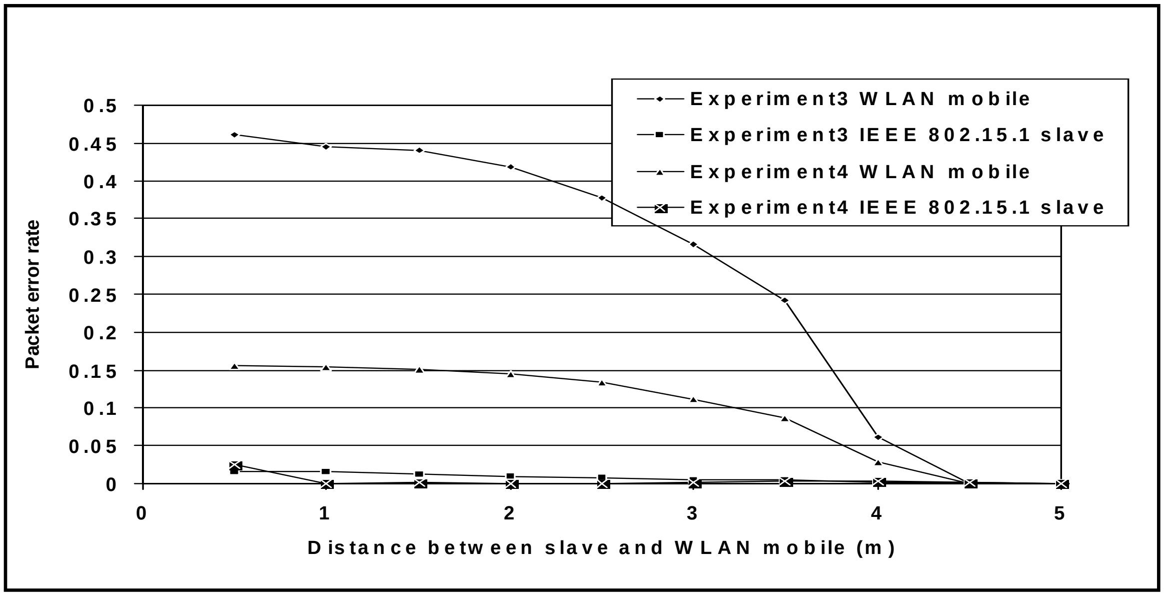

Figure H.8—PER for experiments 3 and 4—11 Mbit/s direct sequence

Figure H.9 illustrates the delay for experiments 3 and 4. The delay curves follow the PER trends described previously.

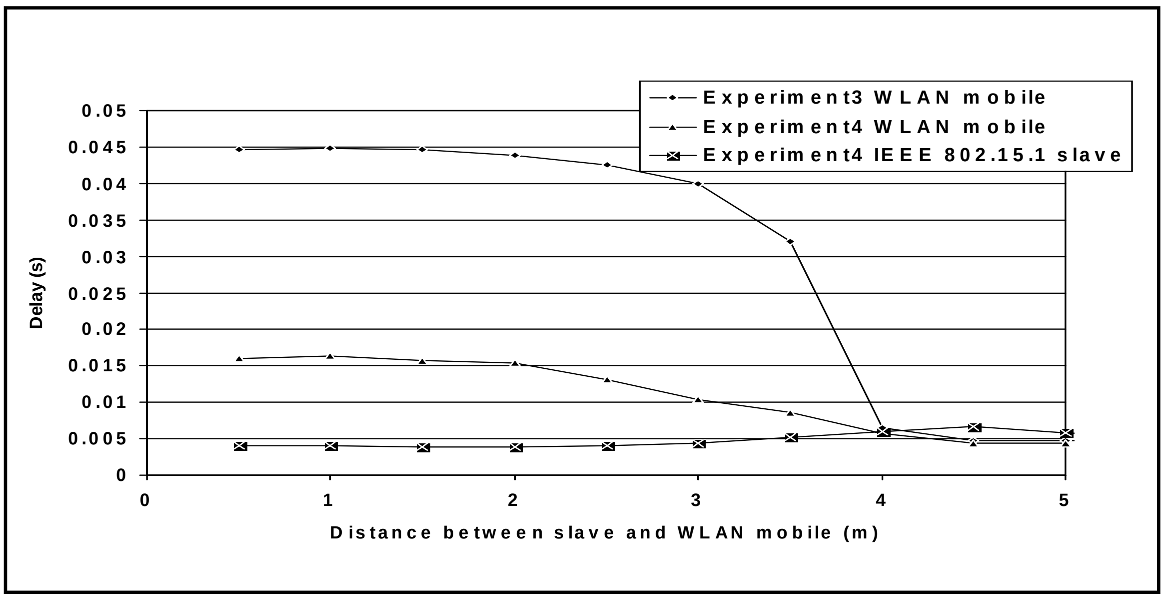

Figure H.9—Access delay for experiments 3 and 4—11 Mbit/s direct sequence

H.3 802.11 1 Mbit/s FH and IEEE 802.15.1 interference

Figure H.10 depicts the PER for experiments 1 and 2. The PER for both the IEEE 802.15.1 and WLAN is negligible (below 5%). Thus, the interference between the WLAN FH and the IEEE 802.15.1 systems is limited.

In Figure H.10, 802.15.1 is the desired signal while IEEE 802.11 (1 Mbit/s FH) is the interferer. The 802.15.1 devices are located far from the IEEE 802.11 receiver, and therefore the packet loss is small. The prediction of 4.1.1 is verified in Figure H.12 where the packet loss for IEEE 802.11 is almost 60%.

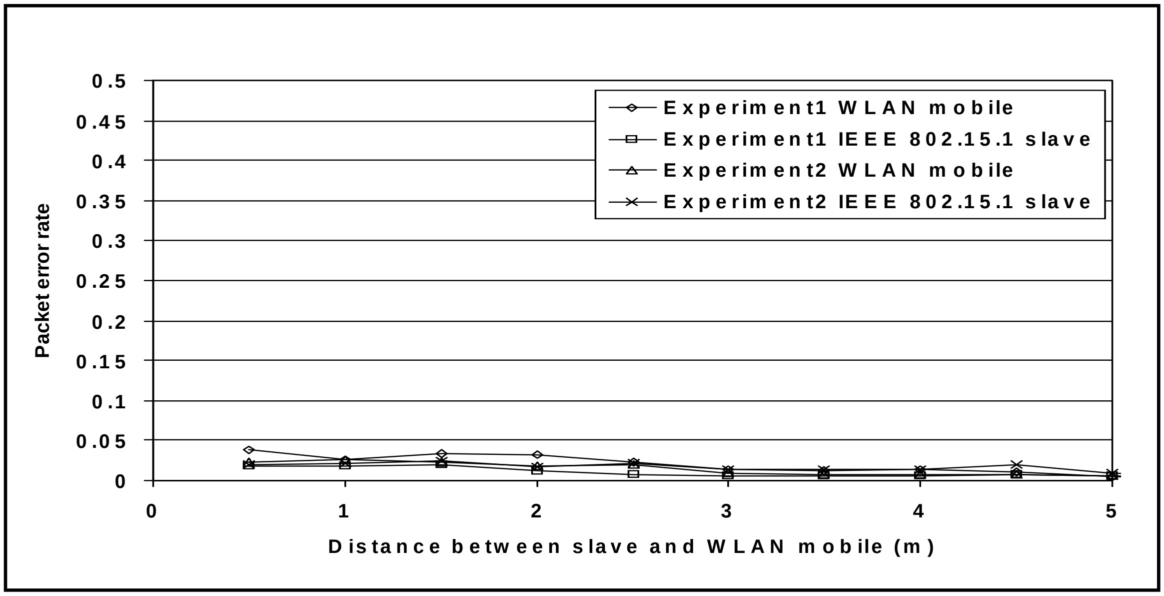

Figure H.10—PER for experiments 1 and 2—1 Mbit/s FH

Figure H.11 shows the delay for experiments 1 and 2. The curves are flat and reflect the PER curves illustrated in Figure H.10.

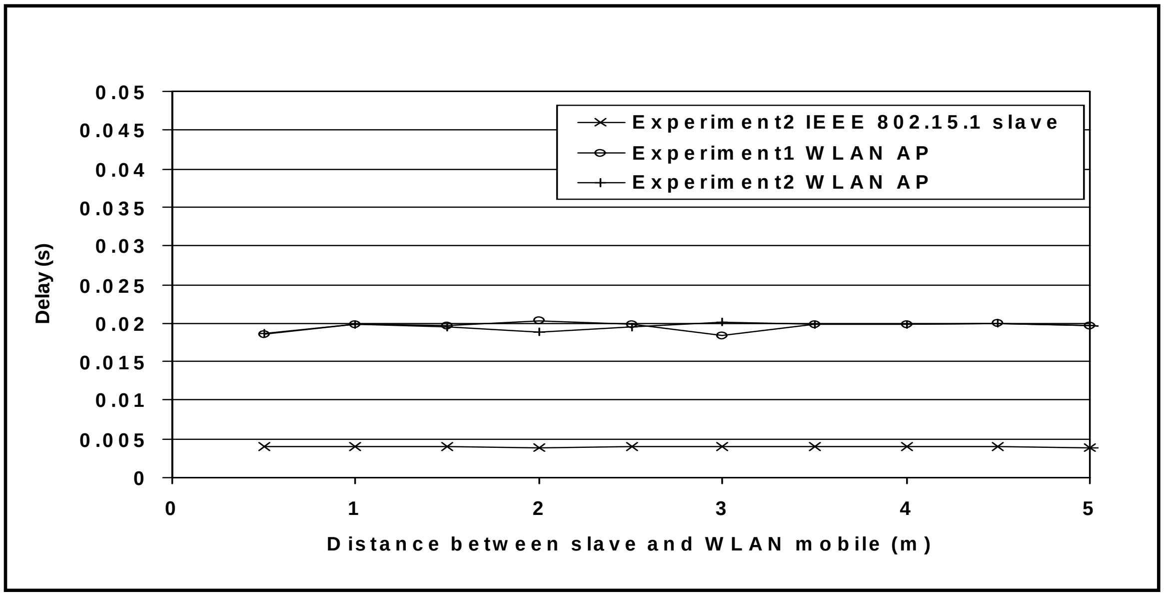

Figure H.11—Access delay for experiments 1 and 2—1 Mbit/s FH

Figure H.12 shows the PER for experiments 3 and 4. The effect of IEEE 802.15.1 voice interference on the WLAN FH system (experiment 3) leads to 60% of PER at 0.5 meter. The PER drops to 10% at 5 meters. The impact of IEEE 802.15.1 data on WLAN results in 17% of PER.

The PER of IEEE 802.15.1 is zero for experiments 3 and 4 due to the fact that the WLAN source is 15 meters away from the IEEE 802.15.1 receiver.

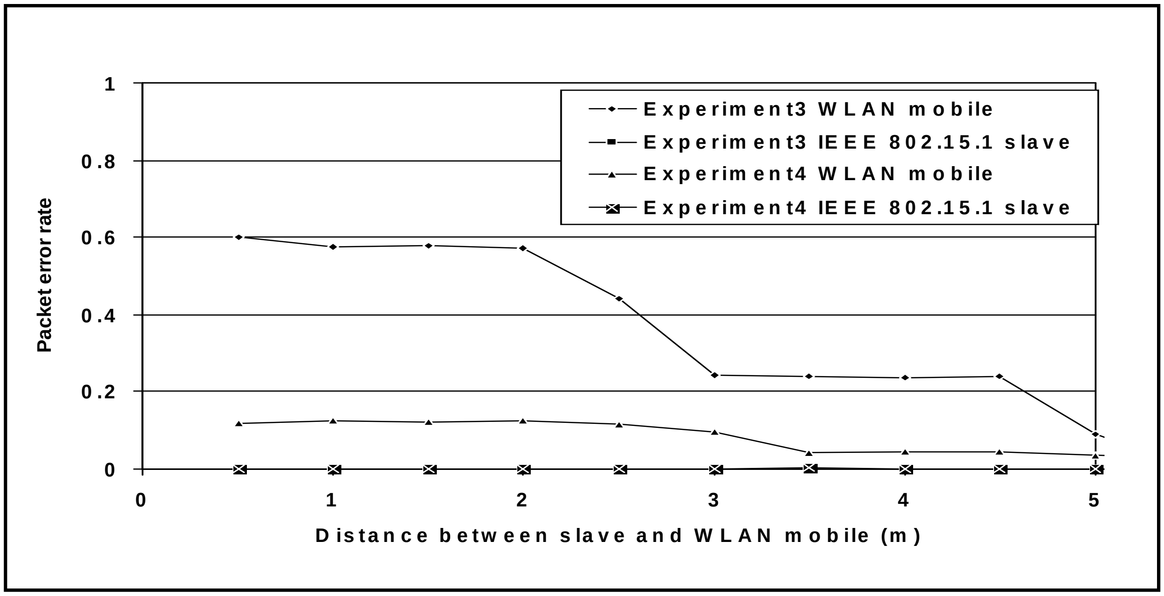

Figure H.12—PER for experiments 3 and 4—1 Mbit/s FH

The PER observed in Figure H.12 for WLAN (experiment 3) leads to extremely high delays, 230 ms, as shown in Figure H.13.

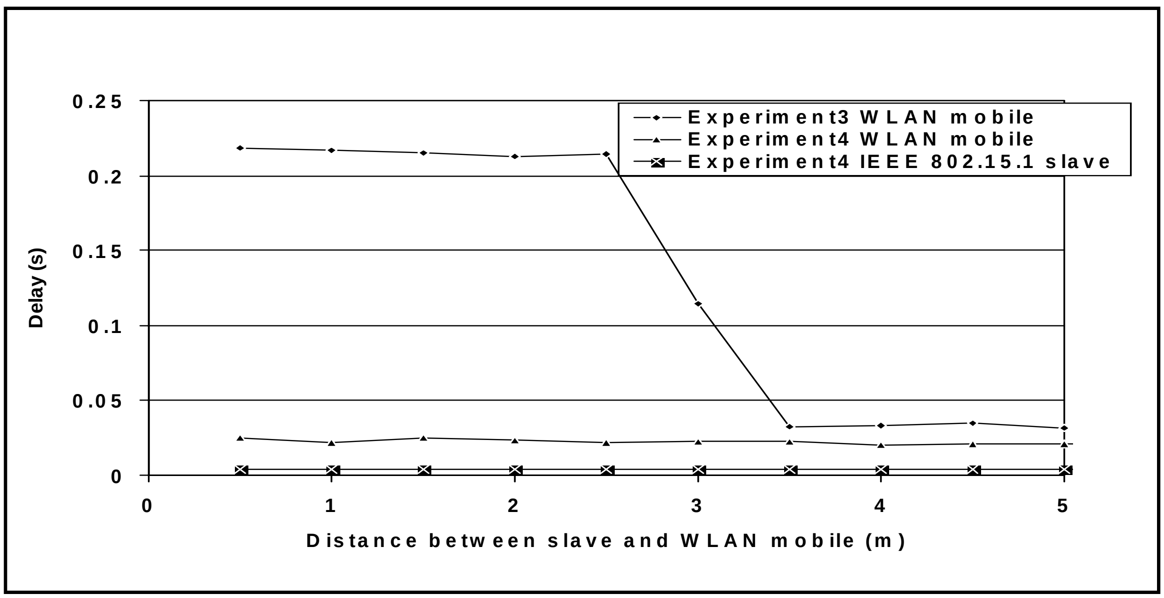

Figure H.13—Access delay for experiments 3 and 4—1 Mbit/s FH

Annex I

(informative)

Performance of WLAN and WPAN utilizing AWMA

Without the use of AWMA the performance of collocated WLAN/WPAN radios is unpredictable due to interference. The performance of the WLAN and WPAN networks utilizing AWMA is predictable.

First define several variables related to the AWMA timing parameters,

$$p = T_{WLAN}/T_B$$

and

$$q = 1 - p = T_{WPAN}/T_B$$

Let β_0 be the throughput of the WLAN with no WPAN present. The WLAN throughput with AWMA enabled is given by

$$\beta_a = p\beta_0$$

Similarly, let γ_0 be the throughput of the WLAN with no WPAN present. The WLAN throughput with AWM enabled is given by

$$\gamma_a = q\gamma_0$$

The AWMA coexistence mechanism will also increase the latency of each packet sent over the WLAN and WPAN networks. Let τ_0 be the average latency of a packet over the WLAN network with no WPAN present. The average increase in latency can be calculated using the total probability formula from probability theory. Let D be the average increase in latency for a given packet. It is necessary to define two events. Let S_1 be the event that the device is ready to transmit the packet during the WLAN interval. Let S_2 be the event that the device is ready to transmit the packet during the WPAN interval. Using only probability, this is represented by

$$E[\delta] = E[\delta|S_1]P[S_1] + E[\delta|S_2]P[S_2] = 0p + (T_{WPAN}/2)q = (T_{WPAN}/2)q$$

Then the average latency over the WLAN with AWMA enabled is given by

$$\tau_a = \tau_0 + E[\delta] = \tau_0 + (q/2)T_{WPAN}$$

Let τ_1 be the average latency over the WPAN network with no WLAN present. Then the average latency over the WPAN with AWMA enabled is given by

$$\tau_b = \tau_1 + (p/2)T_{WLAN}$$

Annex J

(informative)

PTA 802.11b performance results

This annex provides two figures that depict PTA 802.11b performance results. The first indicates the validity of the simulation model by comparison of simulated and measured IEEE 802.11b throughput in the presence of interference. The second shows simulation results for throughput in the presence of interference when PTA is operational.

Figure J.1 shows predicted and measured IEEE 802.11b throughput versus IEEE 802.11b received signal strength at a STA under IEEE 802.15.1 interference. The conditions were:

a) All devices set to 20 dBm transmit power,
b) IEEE 802.11 STA plus collocated IEEE 802.15.1 master moving together with varying distance from an IEEE 802.11 AP,
c) IEEE 802.15.1 Slave positioned 1 meter from the STA,
d) Throughput measured on top of Transport Control Protocol /Internet Protocol (TCP/IP) with saturated offered load from AP to STA device (1500 byte packets from AP and 40 byte packets from STA), and
e) IEEE 802.15.1 link is saturated with DM1 packets.

Simulation uses the analytical model of the PHY BER performance presented in C.3.

There is good visual correspondence between measurement and simulation results. This indicates that the analytical model is capable of predicting performance well enough for the purpose of this subclause.

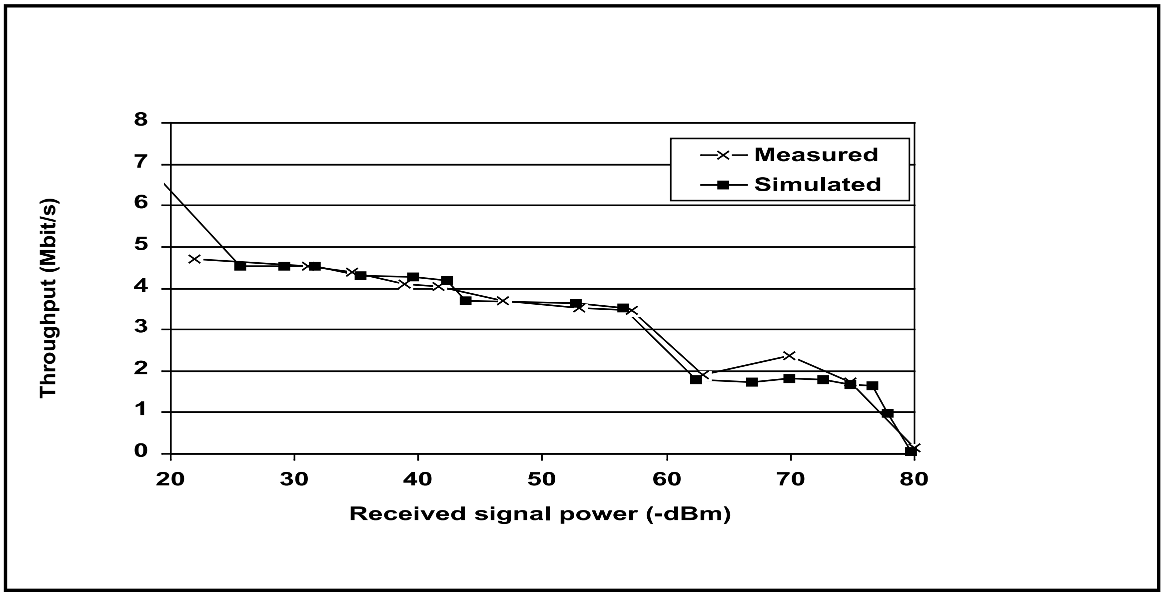

Figure J.1—Throughput of IEEE 802.11b in the presence of IEEE 802.15.1 interference

Figure J.2 shows the result of a simulation that includes the PTA mechanism. There are three curves: IEEE 802.11b throughput with no interference, throughput with IEEE 802.15.1 interference, and throughput with IEEE 802.15.1 interference and the PTA mechanism operating at the collocated IEEE 802.11b STA and 802.15.1 master. The conditions otherwise are the same as before.

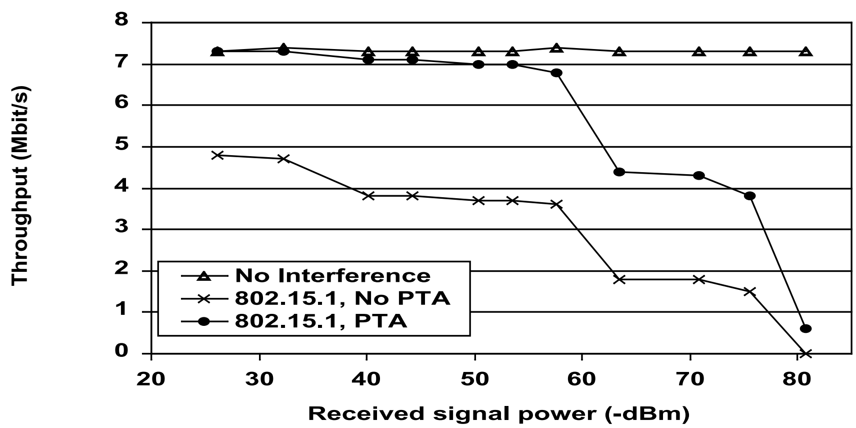

Figure J.2—Effect of PTA mechanism on throughput of IEEE 802.11b

It can be seen that the effect of the PTA mechanism substantially improves the performance of the IEEE 802.11b link in the presence of IEEE 802.15.1 interference. The PTA curve shows essentially full IEEE 802.11b throughput until a received signal power of –53 dBm. It then degrades to about 60% of full throughput up to –75 dBm. Beyond that, throughput decreases rapidly.

Annex K

(informative)

Simulation results for deterministic interference suppression

In Annex C, Figure C.9 shows the BER performance of the original 1-Mbit/s IEEE 802.11b system for an AWGN channel with IEEE 802.15.1 interference. The SIR and SNR are measured at the input to the chip matched filter. Without any type of interference suppression, a SIR value of –5 dB is needed for acceptable performance at all frequency offsets. If the offset is at least 5 MHz, then a value of approximately –11 dB is acceptable. Figure K.1 shows the performance when the adjustable transversal filter, with $N = 3$, is used (where the adjustable transversal filter has $2N$ taps). When using Equation (2) from Clause 7, it is assumed that the SIR was –20 dB. Even when there is a mismatch between the assumed SIR and the actual SIR, the performance is greatly improved. For the worst cases of 2- and 3-MHz offsets, a SIR of –34 dB gives a BER below 10^{-2}.

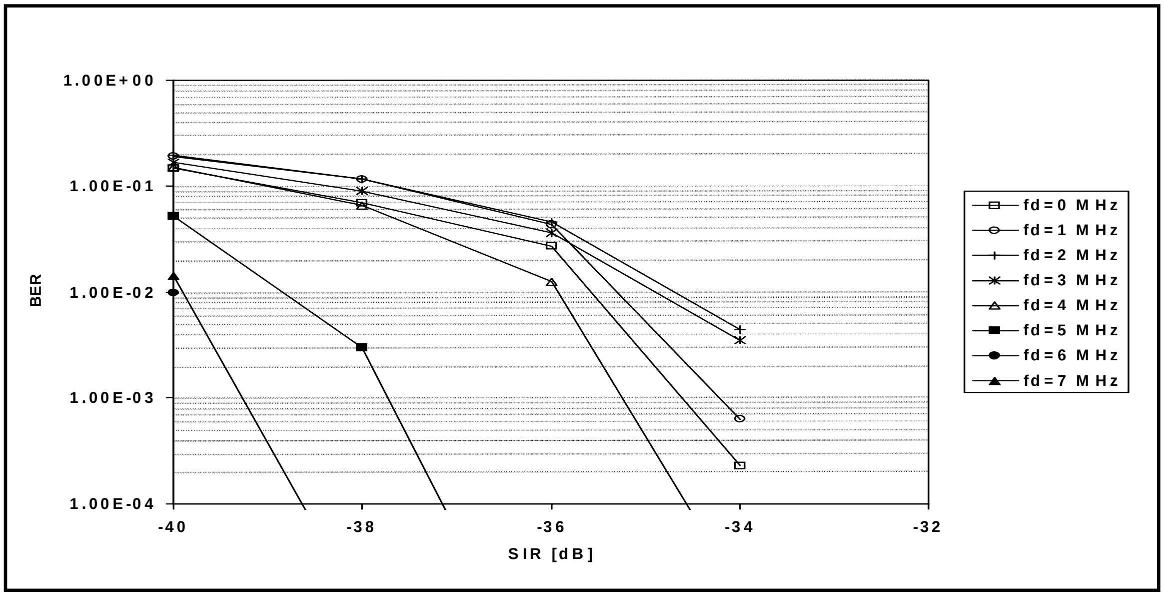

Figure K.1—BER performance of 1-Mbit/s 802.11 receiver with IEEE 802.15.1 interference and with adjustable transversal filtering. AWGN channel. High SNR case.

Annex L

(informative)

Simulation results for adaptive interference suppression

Figure L.1 presents the BER results for 1 Mbit/s with an IEEE 802.15.1 interference for different frequency offsets and SIRs. As mentioned before, $M = 3$ is used for the measurement. The simulation is conducted for a high value of SNR. Here, the IEEE 802.15.1 interference is a short packet generated in the middle of a long WLAN packet. The adaptive filter sees a transition in the input from no interference to interference and vice versa. For this range of SIR, the generic receiver with no interference suppression filter breaks down. On the other hand, the adaptive filter adds a notch at the frequency offset of IEEE 802.15.1. When f_d=0 MHz, this notch will be added to the already existing notch of the Barker code (for the IEEE 802.11b 1 Mbit/s DSSS), so the results are better than for other carrier offsets.

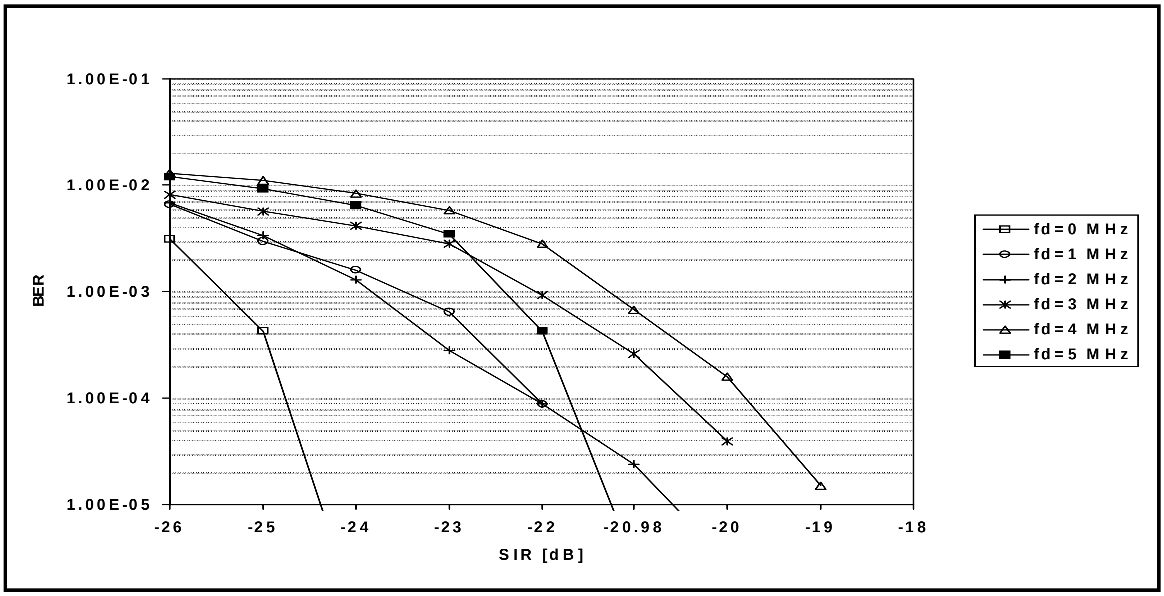

Figure L.1—1 Mbit/s 802.11 receiver with adaptive interference rejection filter

Figure L.2 shows the results for the 11-Mbit/s WLAN with IEEE 802.15.1 interference. Here, $M = 4$ is used, because lower values of M showed very poor results. Comparing Figure L.2 with the generic receiver performance, one can see that the RLSL filter is still capable of rejecting the interference. However, the degree of enhancement in the performance is not as high as the 1-Mbit/s rate. This comes from the inherent processing gain in the 1-Mbit/s WLAN DSSS waveform, which is higher than that in the 11-Mbit/s waveform.

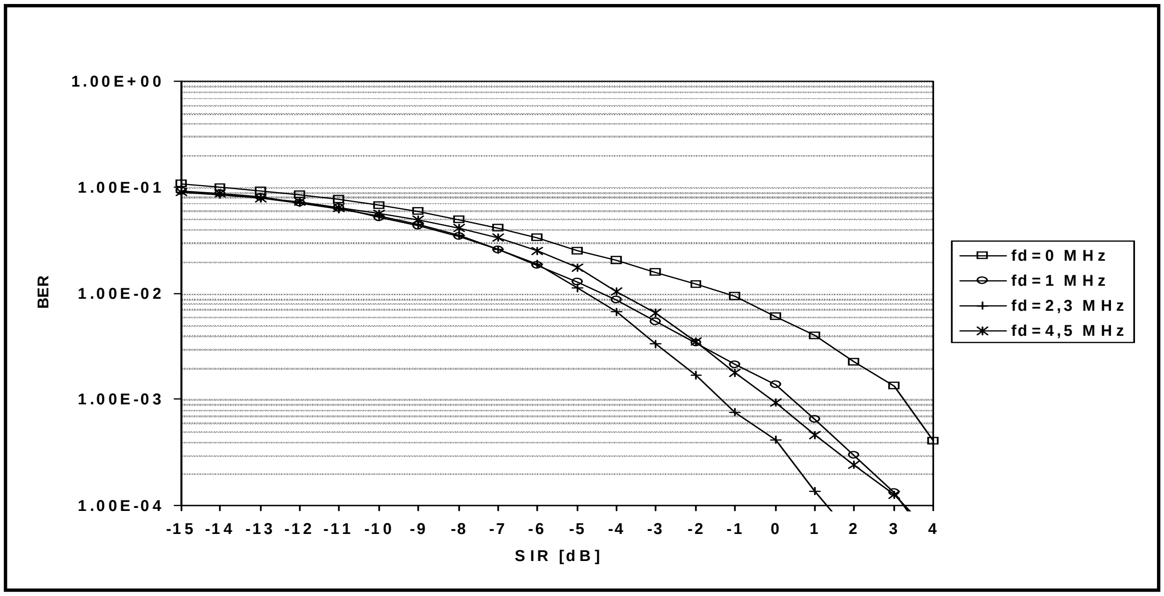

Figure L.2—11 Mbit/s 802.11 receiver with adaptive interference rejection filter

The measurements in Figure L.2 were also carried out for low SNR values and the results are still satisfying Soltanian, et al.[B21].

So far, it has been assumed that the power change of the interference is similar to a step function. In real systems, a transmitter usually has a sequential power up/down procedure in order to decrease the amount of spurious emission into the environment. For 802.15.1, although it is not explicitly defined in the standard, it is assumed that the transmitter reaches its maximum power in a two-bit time interval (2µs) like a ramp function, and similarly for the power down. For this case, the mean-squared-error is measured in addition to the BER results. The approximate mean-squared-error is obtained by averaging the instantaneous squared error, $e^2(n)$, versus n curve, over 200 independent trials of simulation.

Figure L.3 and Figure L.4 (see Soltanian, et al. [B21]) show the mean-squared-error output for the two cases, when there are two overlapping 802.15.1 interferers at different frequency offsets and different SIRs. The spikes in the figure represent a change in the input statistics. For the step power up/down case, these spikes are very high because the interference is added instantaneously. On the other hand, the ramping interference helps the suppression filter to smoothly adapt to the changes, and the spikes at the output of the filter are dramatically decreased. Consequently, the measured BER, which arises from the transient errors, is decreased. From this experiment, the previous BER results for a hopping jammer could be considered to be somewhat pessimistic. Moreover, these results suggest that the adaptive filter is stable, even when there are multiple overlapping interferers.

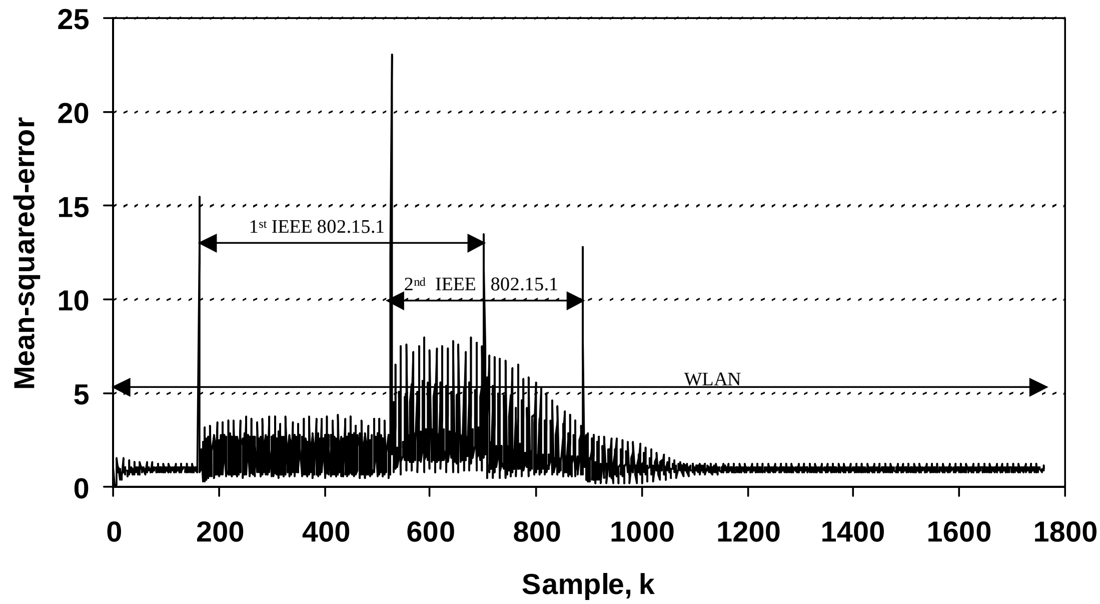

Figure L.3—Mean-squared-error. SIR1,2 = (–18, –16) dB, $f_{d1,2}$ = (–2, 2) MHz. Step BER = 3 x 10^{-2}

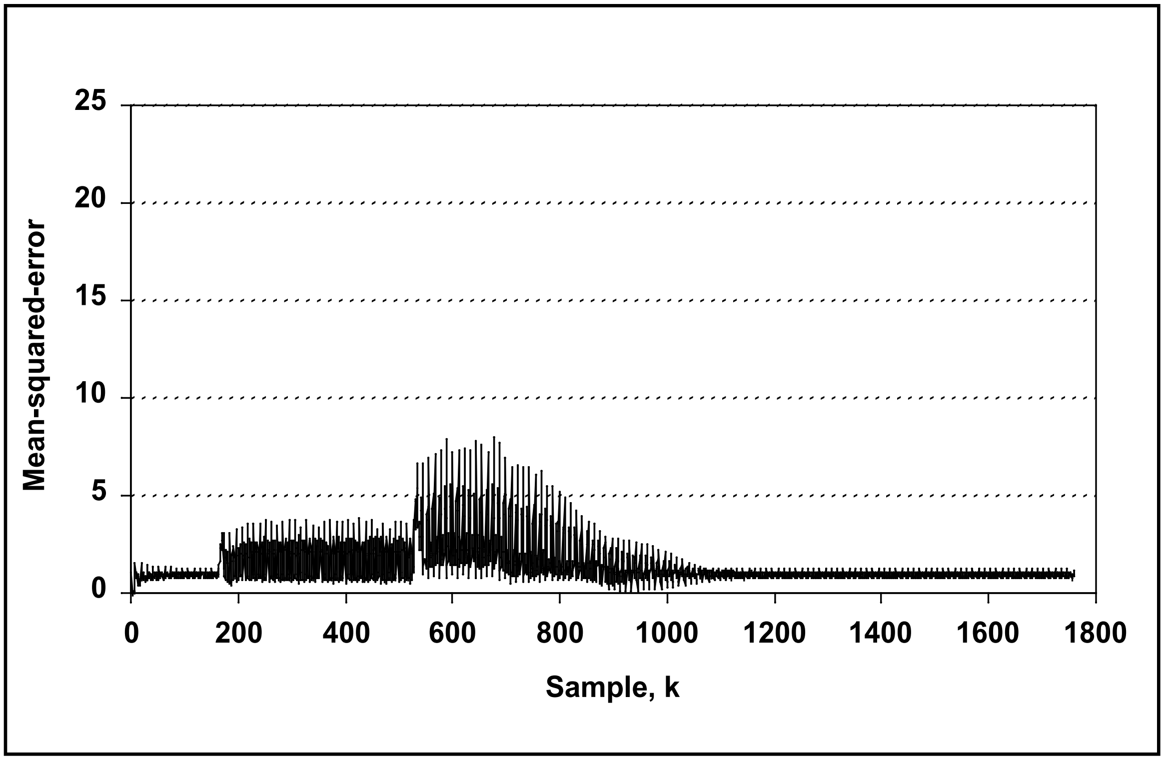

Figure L.4—Mean-squared-error. SIR1,2 = (–18, –16) dB, $f_{d1,2}$ = (–2, 2) MHz. Ramp BER = 7 x 10^{-3}.

Annex M

(informative)

Numerical results for packet scheduling for ACL links

This annex presents the simuations of the scheduling policy proposed in Clause 10 is then simulated. A 4-node topology consisting of two IEEE 802.15.1 nodes (1 master and 1 slave) and two WLAN devices (1 AP and 1 mobile device) is used. The IEEE 802.15.1 devices are located at (0,0) meters for the slave device and (1,0) meters for the master device. The WLAN devices are located at (0,15) meters for the AP and (0,d) for the mobile device. An assumption that WLAN devices implement the IEEE 802.11b specifications at 11 Mbit/s is made. The WLAN mobile is assumed to be transmitting data to the AP which responds with ACK messages. The WLAN offered load is assumed to be 50% of the channel capacity, the data packet size is set to 8000 bits (including the MAC header) and the packet inter arrival time is assumed to be exponential with a mean equal to 1.86 ms.

Three types of IEEE 802.15.1 packet encapsulations, namely, DM1, DM3, and DM5 that occupy 1, 3, and 5 slots, respectively are used. The offered load for IEEE 802.15.1 is set to 30% of the channel capacity, which corresponds to a packet inter arrival of 2.91 ms, 8.75 ms, and 14.58 ms for DM1, DM3 and DM5 packets, respectively.

The transmitted power for IEEE 802.15.1 and WLAN is fixed at 1 mW and 25 mW respectively.

Figure M.1 and Figure M.2 give the PER and the mean access delay respectively measured at the IEEE 802.15.1 slave for varying distances of the interference source from the IEEE 802.15.1 receiver.

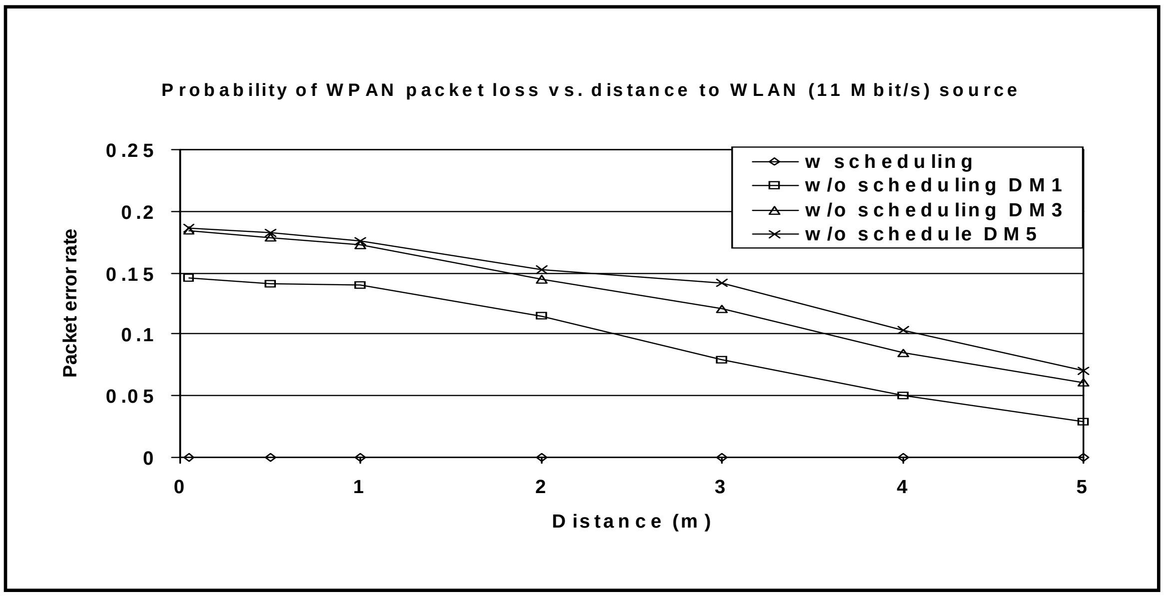

Figure M.1—Effect of scheduling on IEEE 802.15.1—PER

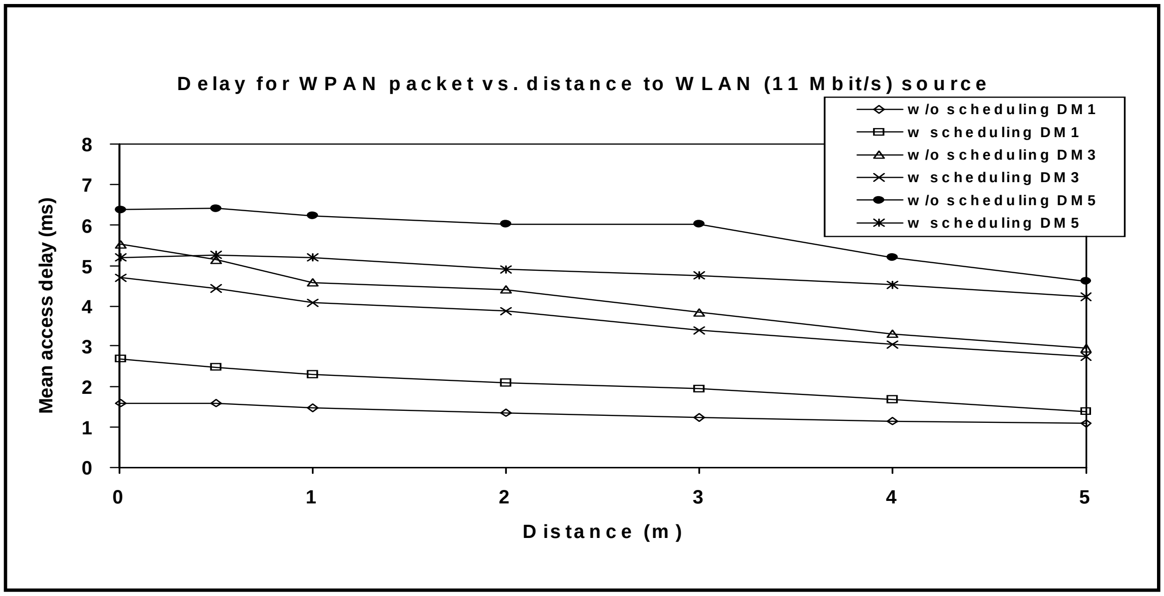

Figure M.2—Effect of scheduling on IEEE 802.15.1—mean access delay

From Figure M.1 shows that using the scheduling policy, leads to a PER of zero, thus basically able to avoid the channels occupied by the interfering system. When no scheduling policy is used the PER is ~ 20% for DM5 and DM3, and 15% for and DM1 packets, respectively, when the IEEE 802.15.1 receiver is at a distance of 0.005 meter from the interference source. As the distance from the interference source is increased the PER drops to around 2.7% for DM1 packets. It is still around 6.7% for DM3 and DM5 packets.

For DM1, an increase in delay from 1.6 ms to 2.6 ms is observed when the scheduling policy is applied. On average the scheduling policy leads to a delay increase of 1 ms (~1.6 IEEE 802.15.1 slots). On the other hand, the scheduling policy reduces the delays by 0.8 ms and 2.6 ms for DM3 and DM5, respectively. Thus, delaying transmission to avoid “bad” channels pays off for packets occupying more than one slot. When “bad” channels are used, dropped packets are retransmitted, yielding large delays. This effect does not apply to DM1 packets because they occupy only one slot.

In summary, the scheduling policy is effective in reducing PER and delay (especially for multi-slot IEEE 802.15.1 packets). Another advantage worth mentioning, are the additional savings in the transmitted power because packets are not transmitted when the channel is “bad.” Moreover, by avoiding channels occupied by other devices, interference on the other system sharing the same spectrum band may be eliminated. Figure M.3 shows the PER for the WLAN Mobile device (receiving ACKs). Scheduling reduces the ACK PER to zero. Therefore, scheduling may be considered as a neighbor friendly policy. Note that the PER at the WLAN AP located at (0,15) meter is negligible in this case because the IEEE 802.15.1 signal is too weak.

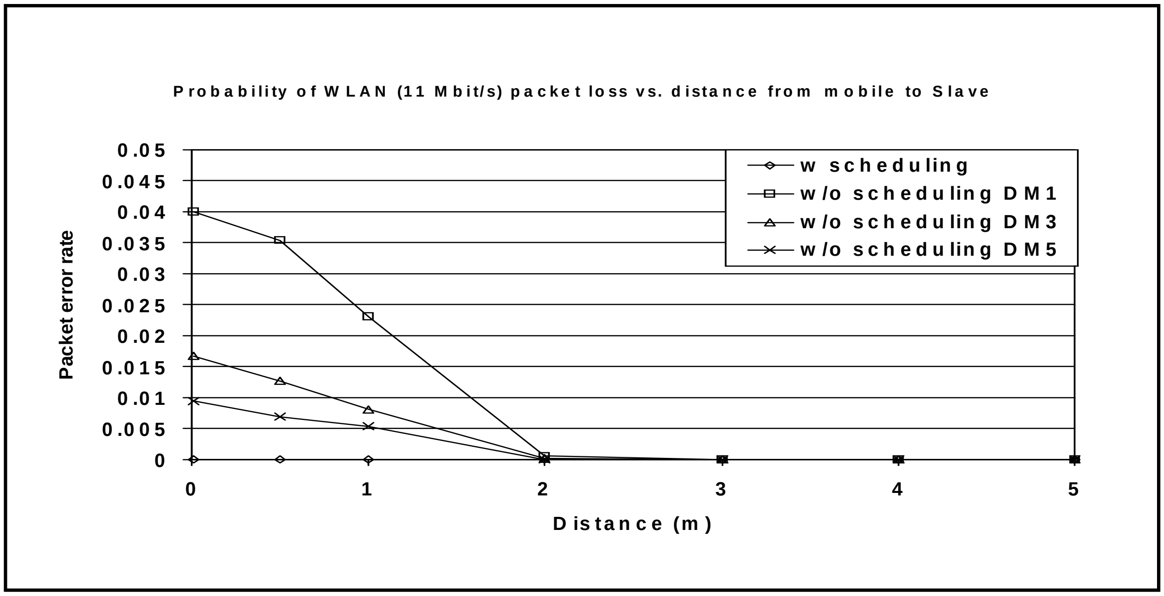

Figure M.3—Impact of MAC scheduling on the WLAN mobile device

Finally, the scheduling policy proposed here works only with data traffic because voice packets need to be sent at fixed intervals. However, if the delay variance is constant and the delay may be limited to a slot (as was shown here), it may be worthwhile to use DM packets for voice using the same scheduling mechanism proposed here.

Annex N

(informative)

Bibliography

N.1 Articles in periodicals

[B1] Aulin, T., and Sundberg, C. E., "Continuous phase modulation—Part 2: Partial response signaling," *IEEE Transactions on Communications*, Vol. 29, pp. 210–225, March 1981.

[B2] Golay, M. J. E., "Complementary series," *IRE Transactions Information Theory*, Vol. IT-7, pp. 82–87, April 1961.

[B3] Iltis, R., "A GLRT-based Spread Spectrum receiver for joint channel estimation and interference suppression," *IEEE Transactions on Communications*, Vol. 37, issue 3, pp. 2174–2183, March 1989.

[B4] Ketchum, J. W., and Proakis, J. G., "Adaptive algorithms for estimating and suppressing narrow-band interference in PN spread-spectrum systems," *IEEE Transactions on Communications*, vol. COM-30, pp. 913–924, May 1982.

[B5] Laster, J. D., and Reed, J. H., "Interference rejection in digital wireless communications," *IEEE Signal Processing Magazine*, Vol 14, issue 3, pp. 37–62, May 1997.

[B6] Milstein, L. B., and Li, L. M., "Rejection of narrow-band interference in PN spread-spectrum systems using transversal filters," *IEEE Transactions on Communications*, vol. COM-30, pp. 925–928, May 1982.

[B7] Milstein, L. B., Davidovici, S., and Schilling, D. L., "The effect of multiple-tone interfering signals on a direct sequence spread spectrum communication system," *IEEE Transactions on Communications*, vol. 30, pp. 436–446, March 1982.

[B8] Murota, K., and Hirade, K., "GMSK modulation for digital mobile radio telephony," *IEEE Transactions on Communications*, Vol. 29, pp. 1044–1050, July 1981.

[B9] Schilling, D. L., Milstein, L. B., Pickholtz, R. L., and Brown, R. W., "Optimization of the processing gain of an M-ary direct sequence spread spectrum communication system," *IEEE Transaction on Communications*, Vol. 28, pp. 1389–1398, August 1980.

[B10] Simon, M. K., and Wang, C. C., "Differential detection of Gaussian MSK in a mobile radio environment," *IEEE Transactions Vehicular Technologies*, Vol. VT-33, pp. 307–320, November 1984.

N.2 Books

[B11] Haykin, S., *Adaptive Filter Theory*, Third Edition, Upper Saddle River, NJ, Prentice Hall, 1996, pp. 630–696.

[B12] IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition.

[B13] Lee, J. S., and Miller, L. E., *CDMA Engineering Handbook*, Boston, MA, Artech House, 1998, pp. 1–1228.

[B14] Proakis, J. G., *Digital Communications*, New York, NY, McGraw-Hill, 1995, pp. 1–928.

[B15] Simon, M. K., Omura, J. K., Scholtz, R. A., and Levitt, B. K., *Spread Spectrum Communications*, vol. 1, Rockville, MD, Computer Science Press, 1985, pp. 135–139.

[B16] Steele, R., *Mobile Radio Communications*, Chichester, UK, John Wiley & Sons, Inc., 1996, pp. 1–779.

[B17] Weik, M., *Communications Standard Dictionary 2ed.*, New York, NY, Kluwer Academic (Van Nostrand Reinhold), 1989, pp. 199–1082.

N.3 Other types of bibliographies

N.3.1 Annotated bibliography

[B18] Marquess, K., "Physical Model Sub-Group Discussion and Questions," IEEE 802.15/138R0 November 1999.

This contribution discusses the physical layer model and cites other related papers.

N.3.2 Articles presented at conferences

[B19] Halford, S., Halford, K., and Webster, M., "Complementary code keying for RAKE-based indoor wireless communications," *Proceedings IEEE International Conference on Circuits and Systems*, Orlando, FL, pp. 427–430, May 1999.

[B20] Soltanian, A., and Van Dyck, R. E., "Physical layer performance for coexistence of Bluetooth® and IEEE 802.11b," *Proceedings 11th Virginia Tech/MPRG Symposium on Wireless Personal Communications*, Blacksburg, VA, pp. 31–41, June 6–8, 2001.

[B21] Soltanian, A., Van Dyck, R. E., and Rebala, O., "Rejection of Bluetooth® interference in 802.11 WLANs," *Proceedings IEEE Vehicular Technology Conference*, Vancouver, Canada, Vol. 2, pp. 932–936, September, 2002.

[B22] Van Nee, R. D. J., "OFDM codes for peak-to-average power reduction and error correction," *Proceedings IEEE Global Telecommunications Conference*, London, UK, Vol. 1, pp. 740–744, November 1996.