

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

T-MOBILE US, INC. AND T-MOBILE USA, INC.
Petitioner

v.

BARKAN WIRELESS ACCESS TECHNOLOGIES, L.P.
Patent Owner

U.S. Patent Nos. 8,559,369 and 9,042,306

Title: Wireless Internet System and Method

DECLARATION OF TAL LAVIAN, PH.D.

Table of Contents

Page

Contents

I.	INTRODUCTION AND QUALIFICATIONS.....	1
B.	Introduction	1
C.	Summary of My Opinions.....	<u>3</u>
D.	Qualifications and Experience	<u>5</u>
E.	Materials Considered.....	<u>9</u>
II.	BASIS FOR MY OPINION AND STATEMENT OF LEGAL PRINCIPLES.....	<u>11</u>
A.	Claim Construction.....	<u>11</u>
B.	Anticipation	<u>11</u>
C.	Obviousness.....	<u>12</u>
1.	Motivation to Combine	<u>15</u>
III.	PERSON OF ORDINARY SKILL IN THE ART	<u>17</u>
IV.	CLAIM CONSTRUCTION	<u>19</u>
A.	“access point” (“AP”) (’306 Patent claims 1, 16, 43 and ’369 Patent claims 1, 8, 13)	<u>19</u>
B.	“AP Identification” (“APID”) (’306 Patent claims 1, 16, 43 and ’369 Patent claims 1, 8, 13).....	<u>23</u>
C.	“proxy server” (’306 Patent claims 1, 16, 43 and ’369 Patent claims 1, 8)	<u>24</u>
D.	“communication module” (’306 Patent claims 1, 16, 43 and ’369 Patent claims 1, 8).....	<u>25</u>
E.	“AP module” (’306 Patent claims 1, 16, 43 and ’369 Patent claim 1).....	<u>26</u>
F.	“connects to the IP based network . . . concurrently with connecting to the IP based network through the second AP” (’306 Patent claims 20 and 57).....	<u>27</u>

Table of Contents
(continued)

	Page
G. The following table summarizes the above claim construction analysis.	<u>28</u>
V. OVERVIEW OF THE CHALLENGED PATENTS AND PRIOR ART	<u>30</u>
A. Brief Description and Summary of the State of the Art at the Time of the Alleged Invention	<u>33</u>
a) Essentially claiming the fundamentals of 802.11 wireless network standard architecture	<u>33</u>
b) Claiming the fundamental of networking layer architecture and the separation between layers (layer 2 and layer 3).....	<u>34</u>
c) Claiming the fundamentals of networking standards	<u>35</u>
d) Products available in the market.....	<u>36</u>
1. 802.11 Wireless Local Area Networks	<u>36</u>
e) 802.11 Standard	<u>38</u>
f) Microsoft’s Feature description and implementation in standard Windows 2000 and Windows XP’s Internet Connection Sharing	<u>41</u>
g) FreeBSD’s Feature Detailed Description and Implementations	<u>42</u>
h) Feature description in the 802.11-1999 standard, and in tutorial of the feature in an 802.11 tutorial	<u>44</u>
i) 802.11-1999 Standard - WDS 4-Address Format	<u>46</u>
j) 802.11-1999 Tutorial – Using WDS 4-Address Formal.....	<u>49</u>
k) Basic implementation of Bridging/Repeating	<u>52</u>
l) Product implementations in the market.....	<u>52</u>
m) D-Link Air-Plus Enhanced 2.4Ghz Wireless Access Point (DWL 90AP+)	<u>53</u>

Table of Contents
(continued)

	Page
n) 3Com’s Office Connect Wireless 11a/b/g Access Point	<u>54</u>
2. Mobile IP	<u>58</u>
3. Security: IPsec, SSH, Packet filters, VPNs, Proxy Servers.....	<u>66</u>
a) Thirty-eight IETF RFCs that specify aspects of IPsec, encryption and tunneling, all of which were well known before 2006	<u>70</u>
b) Packet encapsulation (tunneling).....	<u>70</u>
4. DHCP and NAT part of TCP/IP Suit.....	<u>73</u>
B. Summary of the Challenged Patents	<u>74</u>
1. Specification of the Challenged Patents.	<u>74</u>
2. ’306 Patent Claims.	<u>89</u>
3. ’369 Patent Claims.	<u>90</u>
C. Summary of the Prior Art References	<u>91</u>
1. Summary of Buddhikot.....	<u>91</u>
a) Another embodiment: the gateway also has a “wireless backhaul link”	<u>98</u>
2. Summary of Lord US 6,763,012.....	<u>102</u>
3. Summary of Kim U.S. Pub. No. 2004/0218611	<u>105</u>
4. Summary of Abhishek U.S. Pub. No. 2004/0103278.....	<u>106</u>
5. Summary of Moran PCT App. No. WO2005/069577	<u>106</u>
6. Summary of Fajardo U.S. Pub. No. 2007/0014259.....	<u>107</u>
7. Summary of Vucina U.S. Pub. No. 2005/0261970.....	<u>109</u>
8. Summary of Aarnio U.S. Patent No. 7,606,559	<u>109</u>
VI. PATENT OWNER DID NOT INVENT THE HOT-SPOT.....	<u>110</u>
VII. UNPATENTABILITY OF THE ’306 PATENT	<u>114</u>

Table of Contents
(continued)

	Page
A. Buddhikot And Lord Render Claims 1-12, 14-22, 26-31, 34-37, 39-40, 43, 47-53, 56-63, 65-66 Obvious	<u>114</u>
1. Element-by-Element Analysis of the Challenged Claims	<u>117</u>
a) “at least one communication module adapted to:”	<u>120</u>
(1) “(1) wirelessly connect said computing device to an IP based network via a first wireless access point (AP) having a first AP Identification (APID); and”	<u>121</u>
(2) “(2) wirelessly communicate with other wireless enabled computing devices;”	<u>126</u>
b) “a user interface and display adapted to allow a user of said computing device to interact with destinations over the IP based network, through the first wireless AP, using a first public IP address associated with the computing device; and”	<u>127</u>
c) “an AP module adapted to:”	<u>132</u>
(1) (1) provide a given device of the other wireless enabled computing devices with access to the IP based network by causing said computing device to serve the given device as a second AP having a second APID, distinct from the first APID, and provide the given device access to the network via the first AP; and”	<u>133</u>
(2) “(2) tunnel data traffic from the given device, through said computing device, through the first AP, through the IP network, to a proxy server, such that the proxy server acts as a proxy of the given device and the data traffic is secure from said computing device and first AP”	<u>135</u>

Table of Contents
(continued)

	Page
(3) “and the given device operates on the network using a second public IP address distinct from the first public IP address, with the second public IP address associated with the given device.”	<u>138</u>
2. Dependent claims 2-12, 14, and 15	<u>142</u>
3. Independent claim 16	<u>152</u>
4. Dependent claims of Claim 16.....	<u>154</u>
5. Claim 43	<u>162</u>
6. Dependent Claims of Claim 43	<u>164</u>
B. Buddhikot, Lord, and Moran Render Claims 23-25, 41-42, 44-46, and 67-68	<u>169</u>
1. Motivation to Combine Moran With Buddhikot and Lord....	<u>169</u>
2. Element-by-Element Analysis of the Challenged Claims	<u>170</u>
C. Buddhikot, Lord, and Abhishek Render Claims 32-33 and 54-55 Obvious	<u>178</u>
1. Motivation to Combine Abhishek With Buddhikot and Lord	<u>178</u>
2. Element-by-Element Analysis of the Challenged Claims	<u>179</u>
D. Buddhikot, Lord, and Kim Render Claims 13, 38 and 64 Obvious.....	<u>182</u>
1. Motivation to Combine Kim With Buddhikot and Lord	<u>182</u>
2. Element-by-Element Analysis of the Challenged Claims	<u>183</u>
VIII. GROUNDS FOR UNPATENTABILITY OF THE '369 PATENT	<u>185</u>
A. Buddhikot and Lord Render Claims 1-7 Obvious	<u>185</u>
1. Motivation to Combine Buddhikot With Lord	<u>185</u>
2. Element-by-Element Analysis of the Challenged Claims	<u>185</u>
B. Buddhikot, Lord, and Fajardo Render Claims 8-11 Obvious	<u>191</u>

Table of Contents
(continued)

	Page
1. Motivation to Combine Fajardo with Buddhikot and Lord ...	<u>191</u>
2. Element-by-Element Analysis of the Challenged Claims	<u>193</u>
C. Buddhikot, Lord, Fajardo, and Aarnio Render Claim 12 Obvious.....	<u>199</u>
1. Motivation to Combine Aarnio with Buddhikot, Lord, and Fajardo.....	<u>199</u>
2. Element-by-Element Analysis of the Challenged Claims	<u>200</u>
D. Claim 13 is Obvious Over Buddhikot.....	<u>202</u>
1. Element-by-Element Analysis of the Challenged Claims	<u>202</u>
E. Claim 13 is Anticipated By Vucina	<u>205</u>
1. Element-by-Element Analysis of the Challenged Claims	<u>205</u>
IX. CONCLUSION.....	<u>209</u>

EXHIBIT LIST SUBMITTED WITH PETITIONS

- EX1001 U.S. Pat. No. 8,559,369 B2 to Barkan (“the ’369 Patent”)
- EX1002 U.S. Patent No. 9,042,306 B2 to Barkan (“the ’306 Patent”)
- EX1003 Declaration of Dr. Tal Lavian (“Lavian Decl.”)
- EX1004 Prosecution History of the ’369 Patent
- EX1005 Prosecution History of the ’306 Patent
- EX1006 U.S. Pat. No. 7,562,393 (“Buddhikot”)
- EX1007 U.S. Pat. No. 6,763,012 (“Lord”)
- EX1008 U.S. Pub. No. 2004/0218611 (“Kim”)
- EX1009 U.S. Pub. No. 2004/0103278 (“Abhishek”)
- EX1010 Int’l Pub. No. WO2005/069577 (“Moran”)
- EX1011 U.S. Pub. No. 2007/0014259 (“Fajardo”)
- EX1012 U.S. Pub. No. 2005/0261970 (“Vucina”)
- EX1013 U.S. Pat. No. 7,606,559 (“Aarnio”)
- EX1014 Perkins, Mobile IP, Design Principles and Practices ISBN 0-201-63469-4, (1998)
- EX1015 RFC3344, IP Mobility Support for IPv4 (2002) (subsequent version of RFC3220, which is a subsequent version of RFC2002)
- EX1016 U.S. Pat. No. 7,389,412 (“Sharma”)
- EX1017 Frankel et al., Guide to IPsec VPNs, NIST Special Publication 800-77, (2005)
- EX1018 Lucent IPsec Client Datasheet (2005)
- EX1019 RFC4301 Security Architecture for the Internet Protocol (2005)
- EX1020 RFC2131 Dynamic Host Configuration Protocol (1997)

TABLE OF CONTENTS

(continued)

- EX1021 RFC2663 IP Network Address Translator (NAT) Terminology and Considerations
- EX1022 RFC2002 IP Mobility Support (1996)
- EX1023 Internet Pull of RFC3220, IP Mobility Support for IPv4 (2002) illustrating that it obsoletes RFC2002 and is obsoleted by RFC3344
- EX1024 Patent Owner's Claim Construction Statement filed in case against Verizon
- EX1025 Patent Owner's Infringement Contentions against T-Mobile, '306 patent
- EX1026 U.S. Pat. No. 7,660,584 ("Maxwell")
- EX1027 U.S. Pat. No. 6,636,491 ("Kari")
- EX1028 Excerpts from Newton's Telecom Dictionary, 22nd Edition (2006), ISBN 1-57820-319-8
- EX1029 Excerpts from IEEE100, The Authoritative Dictionary, 7th Edition (2000), ISBN 0-7381-2601-2
- EX1030 U.S. Pat. No. 6,795,689 ("Ogren")
- EX1031 U.S. Pub. No. 2006/0098592 ("Proctor")
- EX1032 IEEE DISTRIBUTED SYSTEMS ONLINE 1541-4922 (2005)
- EX1033 U.S. Pat. No. 6,950,628 ("Meier")
- EX1034 U.S. Pat. No. 8,411,650 ("Iyer")
- EX1035 U.S. Pub. No. 2005/0054343 ("Nykanen")
- EX1036 Patent Owner's Infringement Contentions against T-Mobile, '396 patent
- EX1037 Patent Owner's Technology Tutorial filed in case against Verizon

TABLE OF CONTENTS
(continued)

Table of Contents

Appendix	Title
APP. A	CV of Dr. Lavian
APP. B	ANSI/IEEE Std 802.11, 1999 Edition
APP. C	Not provided due to size but referenced: IEEE 802.11 1997 – 1 Mbit/s and 2Mbit/s
APP. D	Not provided due to size but referenced: IEEE 802.11a 1999 – 54Mbit/s, 5GHz standard
APP. E	Not provided due to size but referenced: IEEE 802.11b 1999 - Higher speed extension in the 2.4 GHz band
APP. F	Not provided due to size but referenced: IEEE 802.11c 1999 – Media Access Control (MAC) Bridges
APP. G	Not provided due to size but referenced: IEEE 802.11d 2001 – Additional regulatory compliance
APP. H	Not provided due to size but referenced: IEEE 802.11e 2005 – Quality of service (QoS)
APP. I	Not provided due to size but referenced: IEEE 802.11f 2003 – Inter access point protocol across distributed systems.
APP. J	Not provided due to size but referenced: IEEE 802.11g 2003 – Further higher data rate extension
APP. K	Not provided due to size but referenced: IEEE 802.11h 2004 – European spectrum and transmit power management extensions in the 5GHz band in Europe
APP. L	Not provided due to size but referenced: IEEE 802.11i 2004 – Enhanced security

TABLE OF CONTENTS
(continued)

Appendix	Title
APP. M	Not provided due to size but referenced: IEEE 802.11j 2004 – Extensions for Japan
APP. N	IEEE 802.11 WDS Clarification IEEE 802.11-05/0710r0 July 2005
APP. O	Submitted by T-Mobile as EX1022
APP. P	Submitted by T-Mobile as EX1023
APP. Q	Submitted by T-Mobile as EX1015
APP. R	Not provided by Appendix. Referenced as: DHCP Related IETF RFCs not including Novell and NetWare: 1534 Oct 1993, 1542 Oct 1993, 2131 May 1997, 2132 May 1997, 2485 Jan 1999, 2563 May 1999, 2937 Sep 2000, 3004 Nov 2000, 3001 Nov 2000, 3046 Jan 2001, 3074 Feb 2001, 3118 June 2001, 3203 Dec 2001, 3256 Apr 2002, 3315 Jul 2003, 3396 Nov 2002, 3397 Nov 2002, 3342 Dec 2002, 3495 Mar 2003, 3527 Apr 2003, 3633 Dec 2003, 3634 Dec 2003, 3646 Dec 2003, 3679 Jan 2004, 3736 Apr 2004, 3825 Jul 2004, 3898 Oct 2004, 3925 Oct 2004, 3942 Nov 2004, 3993 Mar 2005, 4014 Feb 2005, 4030 Mar 2005, 4039 Mar 2005, 4075 May 2005, 4076 May 2005, 4174 Sep 2005, 4242 Nov 2005, 4280 Nov 2005.
APP. S	Not provided due to size but referenced: Amjad Umar, Ph.D. Mobile Computing and Wireless Communications. ISBN 0-9759182-0-6. Publisher NGE Solutions. July 2004.
APP. T	NetGear, Wireless Networking Basics. October 2005.

TABLE OF CONTENTS
(continued)

Appendix	Title
APP. U	Perkins, C., “IP Encapsulation within IP”, IETF RFC 2003, October 1996.
APP. V	Hanks, S., Li, R., Farinacci, D., and P. Traina, “Generic Routing Encapsulation (GRE)”, IETF RFC 1701, October 1994
APP. W	Perkins, C., “Minimal Encapsulation within IP”, IETF RFC 2004, October 1996.
APP. X	K. Ingham, S. Forest. A History and Survey of Network Firewalls. ACM Journal. Original paper TR-CS-2002-37 University of New Mexico, Albuquerque, NM (2002)
APP. Y	Cisco Systems, IP Access List Entry Sequence Numbering, 2003
APP. Z	Levkowetz, H., S. Vaarala, “Mobile IP Traversal of Network Address Translation (NAT) Devices”, IETF RFC 3519, April 2003.
APP. AA	U.S. Pub. No. 2007/0206527
APP. AB	U.S. Pub. 2004/0214572
APP. AC	U.S. Patent No. 7,860,978
APP. AD	S. Kent, R. Atkinson. IETF Network Working Group, IETF RFC 2401 Security Architecture of the Internet Protocol. November 1998.
APP. AE	R. Atkinson. IETF Network Working Group, IETF RFC 1825 Security Architecture of the Internet Protocol. August 1995.
APP. AF	Provided by T-Mobile as EX1019

TABLE OF CONTENTS
(continued)

Appendix	Title
APP. AG	3Com, Configuring a Wireless Distribution System (WDS)

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

I, Tal Lavian, Ph.D., declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

B. Introduction

1. My name is Tal Lavian, and I am a lecturer and Industry Fellow in the Center of Entrepreneurship and Technology (“CET”) at the UC Berkeley College of Engineering.

2. I have been engaged by counsel for Petitioner T-Mobile US, Inc. and T-Mobile USA, Inc. (“Petitioner”) to investigate and opine on certain issues related to U.S. Patent Nos. 8,559,369 (“the ’369 Patent”) and 9,042,306 (“the ’306 Patent”), both entitled “Wireless Internet System and Method” (collectively, the “Challenged Patents”).

3. Specifically, Petitioner has asked me to provide my opinions related to all claims of the ’369 Patent and all claims of the ’306 Patent (the “Challenged Claims”). Because of their similarity, I have offered a single declaration for both patents.

4. I understand that the Challenged Patents are assigned to Barkan Wireless Access Technologies, L.P. (“Patent Owner”).

5. In this declaration, I will discuss the technology related to the Challenged Patents, including an overview of that technology as it was known at

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

the time of the earliest priority date of the Challenged Patents, which is February 22, 2006, according to Petitioner's counsel. This overview of the relevant technology provides some of the bases for my opinions with respect to the Challenged Patents.

6. This declaration is based on the information currently available to me. To the extent that additional information becomes available, I reserve the right to continue my investigation and study, which may include a review of documents and information that may be produced, as well as testimony from depositions that may not yet be taken.

7. In forming my opinions, I have relied on information and evidence identified in this declaration, including but not limited to the Challenged Patents, the prosecution history of the Challenged Patents, and the documents identified in this declaration and in the exhibit list provided in the petitions against the Challenged patents.

8. I have also relied on my own education and experience in the field of Networks and Telecommunications technologies and systems that were already in use prior to, and within the timeframe of the earliest priority date of the claimed subject matter in the Challenged Patents (February 22, 2006).

C. Summary of My Opinions

9. The Challenged Patents purport to describe a method for providing wireless Internet connection to WiFi-enabled devices by enabling mobile computing devices to become access points by running specific software. The two Challenged Patents are related in that the '306 Patent is a continuation of the '369 Patent, so that both Challenged Patents share the same specification, and have similar claims, especially for the independent claims. I typically reference the '306 Patent specification when making specific citations to the specification, but since the Challenged Patents have identical specifications, any discussion of a reference to either patent's specification should be presumed to apply equally to both patents.

10. As I will explain below, the claims of both Challenged Patents do not recite any feature that would have been regarded as novel or non-obvious to a person of ordinary skill in the art. By February 22, 2006 (the earliest priority date of the Challenged Patents), communication over computer networks was well-known, including wireless local area and cellular networks. One of these references, U.S. Patent No. 7,756,393 to Milind M. Buddhikot et. al. ("Buddhikot") was filed two and a half years before the earliest priority date of for

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

the Challenged Patents. Buddhikot discloses a gateway for mobile access that discloses the supposedly inventive features of the Challenged Patents.

11. As I will explain below, the claims recite functionalities which were standard features of IEEE 802.11 standards as well as IETF standards including Mobile IP, IPsec, DHCP and network address translation (NAT) which were well known and widely adopted and implemented years before the Challenged Patents.

12. It is my opinion that claims 1-13 of the '369 Patent and claims 1-68 of the '306 Patent are invalid as obvious. Additionally, as explained below, it is my opinion that claim 13 of the '369 Patent is invalid as anticipated under the patentability standard of 35 U.S.C. § 102.

13. For the purpose of my analysis in this declaration only and based on the disclosure and file history of the Challenged Patents, I will provide my proposed construction of certain terms in claims 1-13 of the '369 Patent and claims 1-68 of the '306 Patent in detail in a later part of this declaration.

14. The subsequent sections of this declaration will first provide my qualifications and experience and will then detail my analysis and observations. My analysis in this declaration is intended to supplement Petitioner's petitions for *inter partes review* of the Challenged Patents. In addition to the analysis and citations contained in this declaration, I have reviewed Petitioner's petitions for

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

inter partes review of the Challenged Patents. I agree with the analysis contained in the petitions. The citations to prior art references contained therein, whether or not also cited in this declaration, form the basis for my opinion that the Challenged Claims of the Challenged Patents are invalid.

D. Qualifications and Experience

15. I have more than 25 years of experience in the networking, telecommunications, Internet, and software fields. I received a Ph.D. in Computer specializing in networking and communications, Science from the University of California at Berkeley in 2006 and obtained a Master's of Science ("M.Sc.") degree in Electrical Engineering from Tel Aviv University, Israel, in 1996. In 1987, I obtained a Bachelor of Science ("B.Sc.") in Mathematics and Computer Science, also from Tel Aviv University.

16. I am currently employed by the University of California at Berkeley and was appointed as a lecturer and Industry Fellow in the Center of Entrepreneurship and Technology ("CET") as part of UC Berkeley College of Engineering. I have been with the University of California at Berkeley since 2000 where I served as Berkeley Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, and Nortel's Scientist Liaison, where some positions and projects were done concurrently, others sequentially.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

17. I have more than 25 years of experience as a scientist, educator and technologist, and much of my experience relates to telecommunications, network communicators, computer networking, mobile and wireless technologies. For eleven years from 1996 to 2007, I worked for Bay Networks and Nortel Networks. Bay Networks was in the business of making and selling computer network hardware and software. Nortel Networks acquired Bay Networks in 1998, and I continued to work at Nortel after the acquisition. Throughout my tenure at Bay and Nortel, I held positions including Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer, and led the development and research involving a number of networking technologies. I led the efforts of Java technologies at Bay Networks and Nortel Networks. In addition, during 1999-2001, I served as the President of the Silicon Valley Java User Group with over 800 active members from many companies in the Silicon Valley.

18. Prior to that, from 1994 to 1995, I worked as a software engineer and team leader for Aptel Communications, designing and developing mobile wireless technologies, mobile wireless devices and network software products. I worked on development of two-way wireless OFDM technology, in the 915 MHz band, under the FCC part 15. The technology was a continuation of military research for low power, wideband OFDM to reduce wireless transmission detectability. From

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

1990 to 1993, I worked as a software engineer and team leader at Scitex Ltd., where I developed system and network communications tools (mostly in C and C++).

19. I have extensive experience in communications technologies including wireless technologies, routing and switching architectures and protocols, including Multi-Protocol Label Switching Networks, Layer 2 and Layer 3 Virtual Private Networks, and Pseudowire technologies. Much of my work for Nortel Networks (mentioned above) involved the research and development of these technologies. For example, I wrote software for Bay Networks and Nortel Networks switches and routers, developed network technologies for the Accelar 8600 family of switches and routers, the OPTera 3500 SONET switches, the OPTera 5000 DWDM family, and the Alteon L4-7 switching product family. I wrote software for Java based device management including software interface to the device management and network management for the Accelar routing switch family network management system.

20. I am named as a co-inventor on more than 100 issued patents and I co-authored more than 25 scientific publications, journal articles, and peer-reviewed papers. Furthermore, I am a Senior Member of the Institute of Electrical and Electronics Engineers (“IEEE”). I am also certified under the IEEE WCET

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

(Wireless Communications Engineering Technologies) Program, which was specifically designed by the IEEE Communications Society (ComSoc) to address the worldwide wireless industry's growing and ever-evolving need for qualified communications professionals.

21. Since 2007, I have been serving as a Principal Scientist at my company Telecomm Net Consulting Inc., where I develop network communication technologies and provide research and consulting in advanced technologies, mainly in computer networking and Internet technologies. In addition, I serve as a Co-Founder and Chief Technology Officer (CTO) of VisuMenu, Inc., since 2010 and to the present, where I design, and develop architecture of visual IVR technologies for smartphones and wireless mobile devices in the field of network communications. I have worked on wireless and cellular systems using a variety of modulation technologies including time-division multiple-access (TDMA), code-division multiple-access (CDMA), and orthogonal frequency-division multiplexing (OFDM). I have additionally worked on various projects involving the transmission and streaming of digital media content.

22. The above outline of my experience with communications systems is not comprehensive of all of my experience over my years of technical experience. Additional details of my background are set forth in my curriculum vitae, attached

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

as **APP. A** to this Declaration, which provides a more complete description of my educational background and work experience. I am being compensated for the time I have spent on this matter. My compensation does not depend in any way upon the outcome of this proceeding. I hold no interest in the Petitioner (T-Mobile US, Inc. and T-Mobile USA, Inc.) or the Patent Owner (Barkan Wireless Access Technologies, L.P.).

E. Materials Considered

23. The analysis that I provide in this Declaration is based on my education and experience in the fields of telecommunications, network communications, and wireless networking, as well as the documents I have considered including the Challenged Patents, which both claim priority to back to a provisional application filed on February 22, 2006. I understand from counsel that there is a dispute about the Challenged Patents' priority date which has no impact on the invalidity arguments presented by the Petitions, so for purposes of this Declaration, I have assumed February 22, 2006 as the relevant priority date. My opinion does not change if a different time of invention is claimed.

24. I reviewed various documents including patents and publications dated prior to February 22, 2006 describing the state of the art at the time of the alleged invention of the Challenged Patents. I have reviewed the documents

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

provided in the exhibit list to the petition and the documents cited in my declaration, including those cited in the appendix list. Documents I have reviewed—including prior art documents relied upon as actually disclosing the limitations of the claims—including those in the exhibit list of the petitions against the '306 and '369 patents, also included with my declaration.

25. Other documents I have reviewed not listed in the table above include certain websites that are publicly available on the Internet, and when they are referenced in this report, their web address is cited.

II. BASIS FOR MY OPINION AND STATEMENT OF LEGAL PRINCIPLES

26. My opinions and views set forth in this declaration are based on my education, training, and experience in the relevant field, as well as the materials I have reviewed for this matter, and the scientific knowledge regarding the subject matter that existed prior to February 22, 2006.

A. Claim Construction

27. It is my understanding that, when construing claim terms, a claim subject to *inter partes* review receives the “broadest reasonable construction in light of the specification of the patent in which it appears.”

B. Anticipation

28. It is my understanding that in order for a patent claim to be valid, the claimed invention must be novel. It is my understanding that if each and every element of a claim is disclosed in a single prior art reference, then the claimed invention is anticipated, and the invention is not patentable according to pre-AIA 35 U.S.C. § 102 effective before March 16, 2013. In order for the invention to be anticipated, each element of the claimed invention must be described or embodied, either expressly or inherently, in the single prior art reference. In order for a reference to inherently disclose a claim limitation, that claim limitation must necessarily be present in the reference.

C. Obviousness

29. Counsel has advised me that obviousness under pre-AIA 35 U.S.C. § 103 effective before March 16, 2013 is the basis for invalidity in the Petitions. Counsel has advised me that a patent claim may be found invalid as obvious if, at the time when the invention was made, the subject matter of the claim, considered as a whole, would have been obvious to a person having ordinary skill in the field of the technology (the “art”) to which the claimed subject matter belongs. I understand that the following factors should be considered in analyzing obviousness: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims; and (3) the level of ordinary skill in the pertinent art. I also understand that certain other factors known as “secondary considerations” such as commercial success, unexpected results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others may be utilized as indicia of nonobviousness. I understand, however, that secondary considerations should be connected, or have a “nexus”, with the invention claimed in the patent at issue. I understand that a person of ordinary skill in the art is assumed to have knowledge of all prior art. I understand that one skilled in the art can combine various prior art references based on the teachings of those prior art references, the general

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

knowledge present in the art, or common sense. I understand that a motivation to combine references may be implicit in the prior art, and there is no requirement that there be an actual or explicit teaching to combine two references. Thus, one may take into account the inferences and creative steps that a person of ordinary skill in the art would employ to combine the known elements in the prior art in the manner claimed by the patent at issue. I understand that one should avoid “hindsight bias” and ex post reasoning in performing an obviousness analysis. But this does not mean that a person of ordinary skill in the art for purposes of the obviousness inquiry does not have recourse to common sense. I understand that when determining whether a patent claim is obvious in light of the prior art, neither the particular motivation for the patent nor the stated purpose of the patentee is controlling. The primary inquiry has to do with the objective reach of the claims, and that if those claims extend to something that is obvious, then the entire patent claim is invalid. I understand one way that a patent can be found obvious is if there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent’s claims. I understand that a motivation to combine various prior art references to solve a particular problem may come from a variety of sources, including market demand or scientific literature. I understand that a need or problem known in the field at the

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

time of the invention can also provide a reason to combine prior art references and render a patent claim invalid for obviousness. I understand that familiar items may have obvious uses beyond their primary purpose, and that a person of ordinary skill in the art will be able to fit the teachings of multiple prior art references together “like the pieces of a puzzle.” I understand that a person of ordinary skill is also a person of at least ordinary creativity. I understand when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If these finite number of predictable solutions lead to the anticipated success, I understand that the invention is likely the product of ordinary skill and common sense, and not of any sort of innovation. I understand that the fact that a combination was obvious to try might also show that it was obvious, and hence invalid, under the patent laws. I understand that if a patent claims a combination of familiar elements according to known methods, the combination is likely to be obvious when it does no more than yield predictable results. Thus, if a person of ordinary skill in the art can implement a predictable variation, an invention is likely obvious. I understand that combining embodiments disclosed near each other in a prior art reference would not ordinarily require a leap of inventiveness.

1. Motivation to Combine

30. I have been advised by counsel that obviousness may be shown by demonstrating that it would have been obvious to modify what is taught in a single piece of prior art to create the patented invention. Obviousness may also be shown by demonstrating that it would have been obvious to combine the teachings of more than one item of prior art. I have been advised by counsel that a claimed invention may be obvious if some teaching, suggestion, or motivation exists that would have led a person of ordinary skill in the art to combine the invalidating references. Counsel has also advised me that this suggestion or motivation may come from the knowledge of a person having ordinary skill in the art, or from sources such as explicit statements in the prior art. Alternatively, any need or problem known in the field at the time and addressed by the patent may provide a reason for combining elements of the prior art. Counsel has advised me that when there is a design need or market pressure, and there are a finite number of predictable solutions, a person of ordinary skill may be motivated to apply common sense and his skill to combine the known options in order to solve the problem. The following are examples of approaches and rationales that may be considered in determining whether a piece of prior art could have been combined

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

with other prior art or with other information within the knowledge of a person having ordinary skill in the art:

- (1) Some teaching, motivation, or suggestion in the prior art that would have led a person of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- (2) Known work in one field of endeavor may prompt variations of it for use in the same field or a different field based on design incentives or other market forces if the variations would have been predictable to a person of ordinary skill in the art;
- (3) Combining prior art elements according to known methods to yield predictable results;
- (4) Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;
- (5) Applying a technique or approach that would have been “obvious to try” (choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success);
- (6) Simple substitution of one known element for another to obtain predictable results;
- or (7) Use of a known technique to improve similar products, devices, or methods in the same way;

III. PERSON OF ORDINARY SKILL IN THE ART

31. I understand that an assessment of claims of the Challenged Patents should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date, which I understand is February 22, 2006. I have also been advised that to determine the appropriate level of a person having ordinary skill in the art, the following factors may be considered: (1) the types of problems encountered by those working in the field and prior art solutions thereto; (2) the sophistication of the technology in question, and the rapidity with which innovations occur in the field; (3) the educational level of active workers in the field; and (4) the educational level of the inventor.

32. The perceived problem and the purported solution identified in the Challenged Patents are generally related to the fields of mobile devices and wireless network communications, and specifically to the field of wireless mobile device communications. Therefore, a person of ordinary skill in the art would need education and work experience in these fields.

33. In my opinion, a person of ordinary skill in the art (“POSITA”) as of February 22, 2006, would possess at least a bachelor’s degree in computer science and/or electrical engineering or comparable experience, plus at least four years of practical experience in the design or development of telecommunications, network

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

communications systems, and wireless network-based communication between computer systems. A master's degree with at least two years of practical experience in the same fields would also be acceptable.

34. Although my qualifications and experience exceed those of the hypothetical person having ordinary skill in the art defined above, my analysis and opinions regarding the Challenged Patents have been based on the perspective of a person of ordinary skill in the art as of February 22, 2006,. I have familiarity with the above described POSITA as I frequently deal with individuals of the above-described experience level, by reviewing their work.

35. The Challenged Patents deal with core concepts and architecture of wireless network communication devices. In short, the Challenged Patents describe a Wi-Fi hotspot where a computing device provides an access point for other Wi-Fi enabled devices to access the Internet. *E.g.*, '306 patent at 11:33-41. The computing device provides standardized networking services—a NAT, and a DHCP to provide IP addresses to the Wi-Fi enabled devices. *Id.* at 11:63-12:2.

36. The ideas contained therein were mainstream at the time of the invention. Indeed, the concepts including wireless access points communicating with each other and with stations, secure traffic, tunneling, proxy servers, mobile IP, dynamic host configuration protocol (DHCP), network address translation

(NAT), and basic failover/redundancy functionality were well-known and incorporated into a large portion of network architectures and devices at the time of the Challenged Patents. Because the technology involved in the Challenged Patents involves largely off-the-shelf components and functionality, and engineer with about 4 years of experience would be well-versed in the concepts discussed in the Challenged Patents.

IV. CLAIM CONSTRUCTION

37. I understand that a claim in an unexpired patent subject to *inter partes* review must be given its “broadest reasonable construction in light of the specification of the patent in which it appears,” which is different from the manner in which the scope of a claim is determined in litigation. I apply the “broadest reasonable construction” standard in my analysis below. For those claims not explicitly listed, I have applied a plain and ordinary meaning.

A. “access point” (“AP”) (’306 Patent claims 1, 16, 43 and ’369 Patent claims 1, 8, 13)

38. In my opinion, the broadest reasonable construction of “**access point**” is a “**a device that connects a computer to a network.**”

39. The Challenged Patents’ specification frequently describe the access point as a device that connects a computer to a network. *See e.g.*, ’306 patent at 2:35-61, 3:3-16, 3:36-43, 4:7-54, 5:20-51, 6:26-34, 11:42-46, Figs. 1-2. The

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

specification provides 802.11 (WiFi) as an example access point, but a POSITA would understand that the claimed access point refers to other network access points as well.

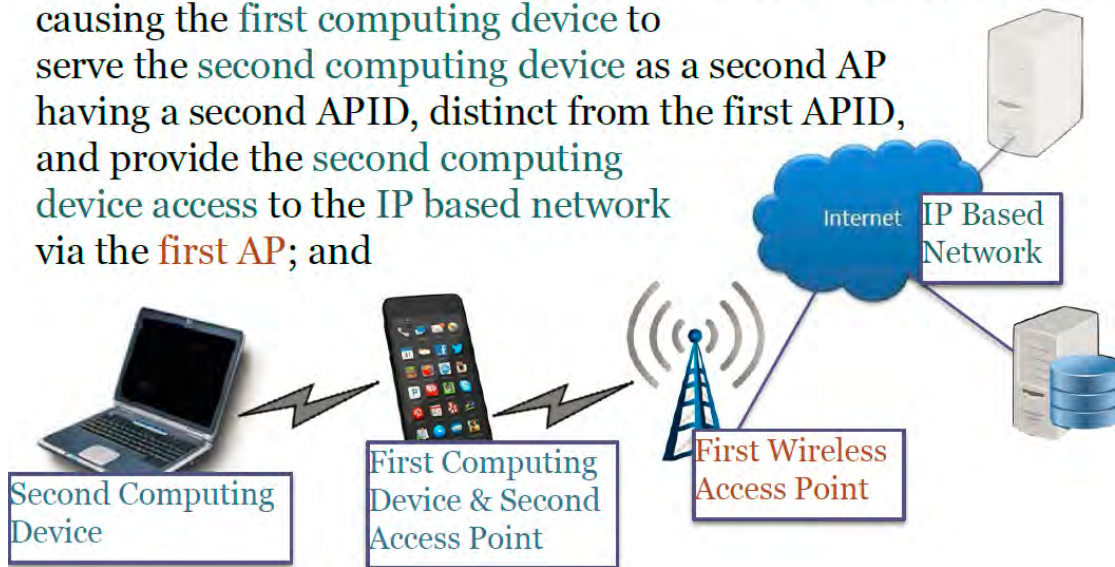
40. The dependent claims of both Challenged Patents also make clear that other network access points are claimed. Claims 5, 28, and 50 of the '306 Patent state that the “**AP is a cellular cell.**” Claims 7, 30, and 52 of the '306 Patent require the first and second APs to be two different wireless protocols (i.e., they cannot both be 802.11). Claim sets 4/27/49; 6/29/51; and 8/31/530 of the '306 Patent recite that access points are included in a cellular network, controlled by the cellular network, and included in a terrestrial wireless network, respectively. Claims 5 and 10 of the '369 Patent state that the “computing device” accessing the first AP “is a cellular phone.” A POSITA would understand that a cellular phone would be used to access cellular access points. In light of the dependent claims of both Challenged Patents, a POSITA would understand the BRI of the claimed “**access point**” cannot be exclusive to a single protocol and instead must be “**a device that connects a computer to a network.**”

41. Further, in my opinion, Patent Owner’s technology tutorial makes clear that they believe that the “access point” to encompass all types of networks.

As shown in a technology tutorial that they submitted to the court, they've identified a cell station as an access point. EX1037 at 16.

Claim 16 - Tethering Connection

provide a **second computing device** of the other wireless enabled computing devices with access to the **IP based network** by causing the **first computing device** to serve the **second computing device** as a second AP having a second APID, distinct from the first APID, and provide the **second computing device** access to the **IP based network** via the **first AP**; and



42. As shown below, from the same document, the tethering device that Patent Owner believes they invented connects Wi-Fi connected devices to the Internet via a 3G wireless link. As shown below, in my opinion, it is clear that Patent Owner believes that the access point can be either a Wi-Fi access point or a 3G base station access point.

Tethering Illustration

- Tethering enables, for example, a smartphone connected to the Internet via a 3G cellular data connection to act as a mobile WiFi hotspot for other wireless devices, such as a tablet computer, laptop computer, or personal data assistant (PDA).



43. Moreover, in my opinion, the Patent Owner believes that the construction of the claimed access point is “**a device that connects a computer to a network**” because the Patent Owner submitted this construction in litigation. (EX1024 at 3-4.) Patent Owner also stated that the **claimed access point encompasses both a cellular network access point** (EX1036 at 1, EX1024 at 3) and a Bluetooth access point (EX1036 at 7, EX1025 at 1) in its infringement contentions. This is also consistent with how the term “access point” is used in the art, which frequently refers to the access point being a device that connects a computer to a network. *See, e.g.*, EX1027, U.S. Patent No. 7,660,584 at 3:38-40 (Nortel patent stating that a 3GPP base station is also referred to as an access

point); EX1026, U.S. Patent No. 6,636,491 at Abstract (Nokia patent referring to a GGSN access point), *see also* EX1029 Newton's Telecom Dictionary at 80; EX1028 IEEE Dictionary at 7.

44. In my opinion, this construction is within the BRI, especially so because Patent Owner believes it to be.

B. “AP Identification” (“APID”) (’306 Patent claims 1, 16, 43 and ’369 Patent claims 1, 8, 13)

45. As described above, the broadest reasonable construction for “**access point**” is “**a device that connects a computer to a network.**” Consistent with that construction, in my opinion, the broadest reasonable construction for “**AP Identification**” is “**identifier for the access point.**”

46. As explained above, “access point” is not restricted to any particular network or protocol. Likewise, APID is not restricted to any network or protocol. The language of APID is not restricted anywhere in the Challenged Patents to a particular network such as 802.11, which utilizes a SSID. This is consistent with Patent Owner's belief—Patent Owner stated that an APID can be either base station's cell ID (EX1036 at 1, EX1025 at 1), a Bluetooth ID, or a WiFi ID (EX1036 at 7, EX1025 at 7) in its infringement contentions.

47. In my opinion, this construction is within the BRI, especially so because Patent Owner believes it to be.

C. “proxy server” (’306 Patent claims 1, 16, 43 and ’369 Patent claims 1, 8)

48. In my opinion, the broadest reasonable construction of “**proxy server**” is “**a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers.**”

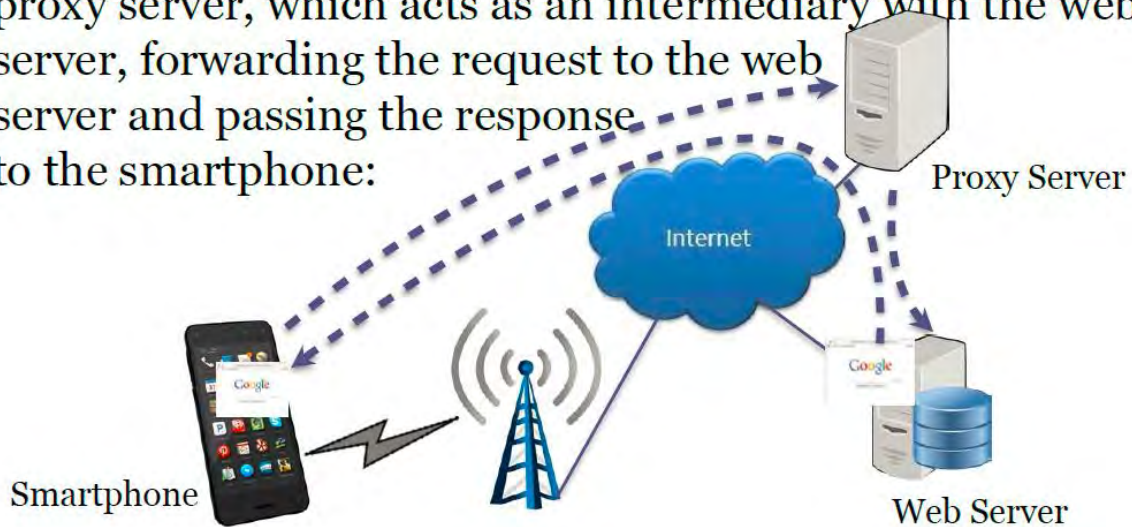
49. This is supported by the Challenged Patents’ specification, which repeatedly refers to a “**proxy server**” as a “**trusted network site**” at the end of the secure tunnel. *E.g.*, ’306 patent at 6:35-53, 11:28-12:2, 14:35-15:19, 16:26-27, 20:49-21:17, 29:27-46, 29:57-65, Figs. 1-2. Moreover, dependent claims like claims 41-42 of the ’306 Patent specify the proxy server can perform additional services, like DHCP and NAT.

50. Patent Owner agrees with this construction—it was provided by Patent Owner in litigation. EX1024 at 1. Further illustrating the breadth of Patent Owner’s belief, Patent Owner’s Infringement Contentions identify a “**packet data network server**” at the end of the secure tunnel as a proxy server. EX1036 at 10, EX1025 at 12. Patent Owner believes that a proxy server is merely an intermediary, *e.g.*, a server that routes traffic to Google.com, for instance. Patent Owner provided a claim construction tutorial to the District Court (below) explaining that the proxy server forwards data traffic to Google.com.

51. In my opinion, this construction is within the BRI, especially so because Patent Owner believes it to be.

Retrieving Web Pages Via a Proxy Server

- Instead of having its own unique IP address for communicating directly with the web server, the smartphone is able to send a request for a web page to a proxy server, which acts as an intermediary with the web server, forwarding the request to the web server and passing the response to the smartphone:



D. “communication module” (’306 Patent claims 1, 16, 43 and ’369 Patent claims 1, 8)

52. In my opinion, no construction is necessary. However, if the Board deems this phrase a “*means plus function*” term under §112, where I believe a construction may be required to be presented, in my opinion the disclosed structure is “**software in a computing device**” and the corresponding function is “**wireless communication.**”

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

53. The Challenged Patents' specification is replete with discussion of software that utilizes the computer's wireless interface to wirelessly communicate with other devices and the Internet. *E.g.*, '306 patent at Fig. 1, 6:1-13, 9:42-10:45, 13:45-14:3, 15:20-40, 16:15-35, 17:34-45. Furthermore, during prosecution, the examiner indicated that this term should be governed by §112, sixth paragraph, requiring identification of sufficient structure. EX1004 at 478-479. The examiner identified the corresponding structure as a "[s]oftware module in a STA or software module in a laptop" and invited Patent Owner to modify or amend if it disagreed. *Id.* Patent Owner did not, and because of their concession, a POSITA would view the Examiner's structure to be within the BRI.

E. "AP module" ('306 Patent claims 1, 16, 43 and '369 Patent claim 1)

54. In my opinion, no construction is necessary. However, if the Board deems this phrase a "*means plus function*" term under §112, where I believe a construction may be required to be presented, in my opinion the disclosed structure is "**software in a computing device**" and the corresponding function is "**providing an access point.**"

55. As described above, the specification is replete with reference to software that provides an access point. *E.g.*, '306 patent at Fig. 1, 6:1-13, 9:42-10:45, 13:45-14:3, 15:20-40, 16:15-35, 17:34-45. When example software

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

(Vagabee) is run, it interacts with the computer's networking interface to make it an access point. *Id.* at 6:9-13. Further, like “**communication module**,” the Examiner rejected the “AP module” in the parent '369 application, identifying the structure as a “[s]**oftware module in a STA or software module in a laptop**,” and invited Patent Owner to modify or amend if it disagreed. EX1004 at 478-479. Patent Owner did not, and because of their concession, a POSITA would view the Examiner's structure to be within the BRI.

F. “connects to the IP based network . . . concurrently with connecting to the IP based network through the second AP” ('306 Patent claims 20 and 57)

56. In my opinion, the broadest reasonable construction is “**algorithm to seamlessly choose between two separate wireless channels to send and receive IP packets.**”

57. The Challenged Patents' specification describes a device that has the ability to communicate via two separate wireless channels. *E.g.*, '306 patent at 9:28-30. This allows the device to access one channel or the other. *Id.* at 6:14-21. The specification describes particular instances where the software would choose one interface over another (*e.g.*, load balancing (6:17-21), hand-overs (6:22-25), or connection quality (2:1-6)). A POSITA would understand that this phrase

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

requires some algorithm to seamlessly choose between channels, but that the particular algorithm used is not claimed in the Challenged Patents.

58. With respect to “**concurrently**,” in my opinion a POSITA would understand that this does not mean simultaneous—a particular packet is not simultaneously sent on both channels. A POSITA would understand that **one channel would be used to send a particular packet, and later in time, the algorithm would have the ability to send a later packet on either one of the two channels.**

G. The following table summarizes the above claim construction analysis.

Claim Term	Broadest Reasonable Construction
“access point” (“AP”)	“a device that connects a computer to a network.”
“AP Identification” (“APID”)	“a device that connects a computer to a network.”
“proxy server”	“a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers.”
“communication module”	“software in a computing device” and the corresponding function is “wireless communication.”
“AP module”	“software in a computing device” and the corresponding function is “providing an access point.”

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

Claim Term	Broadest Reasonable Construction
“connects to the IP based network . . . concurrently with connecting to the IP based network through the second AP”	“algorithm to seamlessly choose between two separate wireless channels to send and receive IP packets.”

V. OVERVIEW OF THE CHALLENGED PATENTS AND PRIOR ART

59. I have reviewed and analyzed the prior art references and materials listed in above. In my opinion, each and every limitation of claims 1-13 of the '369 Patent are obvious over Buddhkot/Lord (claims 1-7); Buddhikot/Lord/Fajardo (claims 8-11); Buddhikot/Lord/Fajardo/Aarnio (claim 12); Buddhikot (13), and claim 13 is anticipated by Vucina.

60. In my opinion, each and every limitation of claims 1-68 of the '306 Patent are obvious over Buddhikot/Lord (claims 1-12, 14-22, 26-31, 34-37, 39-40, 43, 47-53, 56-63, 65-66) Buddhikot/Lord/Moran (claims 23-25, 41-42, 44-46, 67-68); Buddhikot/Lord/Abhishek (claims 32-33, 54-55); Buddhikot/Lord/Kim (claims 13, 38, 64).

61. Additionally, at the priority date of the Challenged Patents, one of skill was familiar with the then-current ANSI/IEEE Standard 802.11, 1999 Edition [APP. A] and prior iterations of that standard in versions IEEE 802.11[a-j] [APP. B-L]. For example, one of skill would understand the analysis of the 802.11 standard in the IEEE 802.11 WDS Clarification IEEE 802.11-05/0710r0 published in July 2005 [APP. M]. This understanding of the 802.11 wireless network standard—as well as other networking services discussed in the Challenged Patents and prior art like Mobile IP [APP. N-P], IPSec [APP. Q], and DHCP

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

[APP. R]—forms part of the background knowledge that one of skill would have at the priority date of the Challenged Patents.

62. I observe that the claims of the Challenged Patents reveal significant redundancy and duplication—other independent and dependent claims recite substantially the same and in many cases identical language as claims 1-15, 17, 20, 23-25, 32-37, and 41-42 of the '306 Patent. I am aware that the Petitioner may use my analysis of these claims to show the unpatentability of other claims that recite substantially the same limitations. My focus on these claims does not suggest that my opinions do not apply to other claims; to the contrary, to the extent other claims recite the same language or impose the same limitations, my analysis applies.

63. I am informed that Buddhikot qualifies as prior art because it issued from an application filed on October 20, 2003, which is more than a year before the earliest application to which the Challenged Patents can claim priority (February 22, 2006), and it was published May 12, 2005. I am informed that Lord qualifies as prior art because it issued from an application filed July 21, 2000, and it was published on July 13, 2004. I am informed that Kim qualifies as prior art because it was published on November 4, 2004. I am informed that Abhishek qualifies as prior art because it was published on May 27, 2004. I am informed

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

that Moran qualifies as prior art because it was filed in English designating the US on January 17, 2005, claiming priority to January 15, 2004, and published on July 28, 2005. I am informed that Fajardo qualifies as prior art because it was filed August 31, 2005, claiming priority to a July 14, 2005 provisional application. I am informed that Vucina qualifies as prior art because it was filed on May 21, 2004, and published on November 24, 2005. I am informed that Aarnio qualifies as prior art because it was filed December 21, 2004.

64. Before explaining how the prior art applies to the claims, I will briefly summarize the state of the art, the Challenged Patents, and each piece of prior art and provide an overview of how I have applied it.

A. Brief Description and Summary of the State of the Art at the Time of the Alleged Invention

65. The Challenged Patents purport a “**computing device**” that functions as a “**wireless access point (AP)**”. The computing device can “**wirelessly communicate with other wireless enabled computing devices**”. The computing device can connect to “**an IP based network**” via wireless access points using a “**public IP address**”. The computing device can secure the traffic by “**tunnel[ing] data traffic... to a proxy server.**” In addition, devices which communicate with the proxy server through the computing device can have a “**second public IP address**”. The Challenged Patents require that wireless access points have identification (APID).

a) Essentially claiming the fundamentals of 802.11 wireless network standard architecture

66. **The Challenged Patents are essentially claiming the fundamentals of 802.11 wireless network standard architecture with end-to-end connectivity services provided by the standard of the Internet protocol suite, including IP addressing, Dynamic Host Configuration Protocol (DHCP), Network Address Translation (NAT), basic encapsulation inherent to protocol stack layering and tunneling.**

b) Claiming the fundamental of networking layer architecture and the separation between layers (layer 2 and layer 3)

67. The Challenged Patents are also claiming forms of integrating between technologies, when in reality such integrations are inherent to the design of the standards defining the technologies and the overall architecture they were designed to support. For example, layer 2 protocols are unaware of layer 3 encapsulation, hence are unrelated to end-to-end connectivity, routing or tunneling. Any claim which attempts to relate between layer 2 functionality and layer 3 or higher functionality is simply a repetition of standards, well known research, and best design practices which were well known and widely accepted.

68. The existing technologies which I describe in more detail below enabled all the capabilities described in the claims of the Challenged Patents including: wireless connectivity between computing devices, wireless connectivity between computing devices and access points, wireless connectivity between access points and other access points using the same protocols and wireless connectivity between devices or access points which use different wireless protocols (*e.g.* 802.11 to 3G).

c) Claiming the fundamentals of networking standards

69. I also describe how mobile stations are able to maintain connectivity with servers and other computers over the Internet while they move between mobile networks. This problem was understood and solved by the ANSI, IEEE, and IETF years before the Challenged Patents when it defined Mobile IP. Mobile IP is an architecture which allows a mobile station to be known by its ‘home’ IP address and receive data packets using a ‘foreign’ IP address which is tunneled from its home address.

70. The existing technologies (e.g. IPsec, NAT and proxy servers) and widely used best practices involving these technologies addressed the need to secure traffic as well as the need to secure computing devices from unauthorized access. Tunnels use protocol encapsulation which by definition enables a device at the end of a tunnel to be reached through a second “public IP address” which is distinct from its first “public IP address”. This is a standard behavior which can be used with or without a proxy server. The Challenged Patents claim that a proxy server provides a tunnel end-point is simply a private case of many possible uses of proxy servers in the context of securing traffic and devices. I show this in detail below.

d) Products available in the market

71. The teachings of the Challenged Patents were present in several technological standards related to computer networking in general and wireless networking specifically that existed long before the Challenged Patents' priority date of February 22, 2006, and these standards were well known to one of skill.

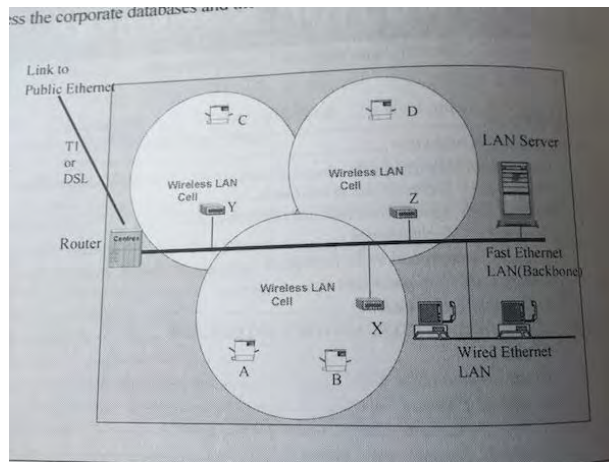
1. 802.11 Wireless Local Area Networks

72. 802.11 is an IEEE standard which before the time of the invention had been adopted worldwide. The IEEE first published the wireless 802.11 standard in 1997 [APP. C]. By the time of the filing of the Challenged Patents, the IEEE had published a revision [APP. D] as well as ten updates [APP. D-M].

73. At the time of the Challenged Patents wireless LAN technology was widely used in home, corporate and public locations. The Challenged Patents themselves are based on the fact that 802.11 was widely used. As the Challenged Patents acknowledge: "*Currently, there is a growing number of WiFi public hot-spots (or access points). These access points allow WiFi enabled devices (which we refer to as STA) that are in their coverage area to connect to the internet.*" '306 Patent at 1:30-33. The Challenged Patents describe standard 802.11 process including 'discovery', 'authentication' and 'association.' *Id.* at 3:24:35.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

74. A wireless network uses radio signals instead of cables. A device which supports wireless communications uses an antenna instead of a physical port. A small office wireless network is depicted in the figure below¹:



75. “The network consists of three cells, handled by access points X, Y, Z. This type of wireless LAN allows users to access the Internet from laptops equipped with wireless cards.” **APP. S** at 6-2. The wireless cards have the antennas as well as the circuits required to transform radio signals to and from digital signals.

76. IEEE 802.11 wireless nodes communicate with each other using radio frequency signals in the ISM (Industrial, Scientific, and Medical) band. For example, 802.11 g/b use bands between 2.4 GHz and 2.5 GHz. Neighboring

¹ **APP. S**, page 6-2

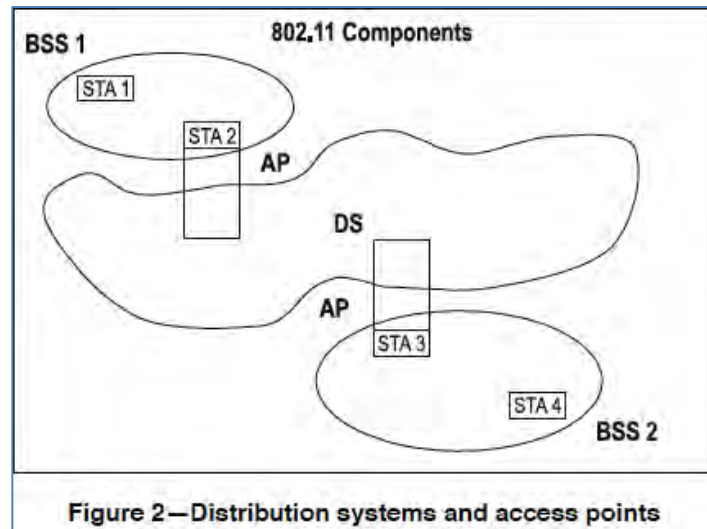
channels are 5 MHz apart. The preferred channel separation between the channels in neighboring wireless networks is 25 MHz (five channels). This means that you can apply up to three different channels within your wireless network. In the United States, only 11 usable wireless channels are available, so it is recommended to start using channel 1, grow to use channel 6, and add channel 11 when necessary, because these three channels do not overlap.²

e) 802.11 Standard

77. The following are from the 802.11 wireless standard at the time of the Challenged Patents, **App. B**, section 3. They describe key 802.11 wireless elements and services.

- **Access point (AP):** Any entity that has station functionality and provides access to the via the wireless medium (WM) for associated stations to distribution services. This is depicted in the figure below (**APP. B** page 11). Each access point is itself a station (STA – see below) that also provides access to a distribution system (DS – see below).

² **APP. T** at 2-2, 2-3.



- **Ad hoc network:** A network composed solely of stations within mutual communication range of each other via the wireless medium (WM). An ad hoc network is typically created in a spontaneous manner. The principal distinguishing characteristic of an ad hoc network is its limited temporal and spatial extent. These limitations allow the act of creating and dissolving the ad hoc network to be sufficiently straightforward and convenient so as to be achievable by nontechnical users of the network facilities; i.e., no specialized “technical skills” are required and little or no investment of time or additional resources is required beyond the stations that are to participate in the ad hoc network. The term ad hoc is often used as slang to refer to an independent basic service set (IBSS).
- **Association:** The service used to establish access point/station (AP/STA) mapping and enable STA invocation of the distribution system services (DSSs).
- **Authentication:** The service used to establish the identity of one station as a member of the set of stations authorized to associate with another station.
- **Basic service set (BSS):** A set of stations controlled by a single coordination function.
- **Distribution:** The service that, by using association information, delivers medium access control (MAC) service data units (MSDUs) within the

- distribution system (DS).
- **Distribution system (DS):** A system used to interconnect a set of basic service sets (BSSs) and integrated local area networks (LANs) to create an extended service set (ESS).
 - **Distributed system medium (DSM):** The medium or set of media used by a distribution system (DS) for communications between access points (APs) and portals of an extended service set (ESS).
 - **Extended service set (ESS):** A set of one or more interconnected basic service sets (BSSs) and integrated local area networks (LANs) that appears as a single BSS to the logical link control layer at any station associated with one of those BSSs.
 - **Infrastructure:** The infrastructure includes the distribution system medium (DSM), access point (AP), and portal entities. It is also the logical location of distribution and integration service functions of an extended service set (ESS). An infrastructure contains one or more APs and zero or more portals in addition to the distribution system (DS).
 - **Independent basic service set (IBSS):** A BSS that forms a self-contained network, and in which no access to a distribution system (DS) is available.
 - **Mobile station:** a type of station that uses network communications while in motion.
 - **Portal:** The logical point at which media access control (MAC) service data units (MSDUs) from a non-IEEE 802.11 local area network enter the distribution system (DS) of an extended service set (ESS).
 - **Station (STA):** Any device that contains an IEEE 802.11 conformant medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM).
 - **Wireless Distribution System (WDS): the IEEE published clarifications of this term, see APP. N.**
 - **Wireless medium:** The medium used to implement the transfer of protocol data units (PDUs) between peer physical (PHY) entities of a wireless local area network (LAN).

**f) Microsoft's Feature description and implementation
in standard Windows 2000 and Windows XP's Internet
Connection Sharing**

78. Microsoft supported "Internet Connection Sharing" at least since Windows 2000 which was released in February 2000. This capability was supported by Windows XP (2003), and all other versions of windows that followed. This feature enables a Windows machine to create an ad-hoc wireless network and route Internet traffic through an Internet connection.



79. This is a regular Windows user or server machine, with a wireless network card and all the features of such a machine including a monitor (as well as a keyboard and mouse). The machine uses a second network interface to connect to the Internet.

g) FreeBSD's Feature Detailed Description and Implementations

80. The FreeBSD distribution of UNIX enabled configuring a computer with a wireless card as an access point. For example, the reference cited is for FreeBSD version 5.3 which was released on November 6, 2004.³ The configuration is done by software which is part of the operating system or loaded by the operating system.⁴ A FreeBSD computer with bridging software could configure one or two wireless cards to act as an AP and an STA, simultaneously or separately. This is an earlier implementation of software and hardware which implements the capability described in the Challenged Patents.

³ <https://en.wikipedia.org/wiki/FreeBSD>

⁴ <https://docs.freebsd.org/doc/5.4-RELEASE/usr/share/doc/handbook/network-wireless.html>

Declaration of Tal Lavian, Ph.D. in Support of Petitions for *Inter Partes* Review of U.S. Patent Nos. 8,559,369 and 9,042,306

25.3.3.2 Building a FreeBSD Access Point

25.3.3.2.1 Requirements

In order to set up a wireless access point with FreeBSD, you need to have a compatible wireless card. Currently, only cards with the Prism chipset are supported. You will also need a wired network card that is supported by FreeBSD (this should not be difficult to find, FreeBSD supports a lot of different devices). For this guide, we will assume you want to [bridge\(4\)](#) all traffic between the wireless device and the network attached to the wired network card.

The hostap functionality that FreeBSD uses to implement the access point works best with certain versions of firmware. Prism 2 cards should use firmware version 1.3.4 or newer. Prism 2.5 and Prism 3 cards should use firmware 1.4.9. Older versions of the firmware may or may not function correctly. At this time, the only way to update cards is with Windows® firmware update utilities available from your card's manufacturer.

25.3.3.2.2 Setting It Up

First, make sure your system can see the wireless card:

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
    inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
    ether 00:09:2d:2d:c9:50
    media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
    status: no carrier
    ssid ""
    stationname "FreeBSD Wireless node"
    channel 10 authmode OPEN powersavemode OFF powersavesleep 100
    wepmode OFF weptxkey 1
```

Next, you will need to load a module in order to get the bridging part of FreeBSD ready for the access point. To load the [bridge\(4\)](#) module, simply run the following command:

```
# kldload bridge
```

It should not have produced any errors when loading the module. If it did, you may need to compile the [bridge\(4\)](#) code into your kernel. The [Bridging](#) section of this handbook should be able to help you accomplish that task.

Now that you have the bridging stuff done, we need to tell the FreeBSD kernel which interfaces to bridge together. We do that by using [sysctl\(8\)](#):

```
# sysctl net.link.ether.bridge.enable=1
# sysctl net.link.ether.bridge.config="wi0,xl0"
# sysctl net.inet.ip.forwarding=1
```

On FreeBSD versions earlier than 5.2, you need to use the following options instead:

```
# sysctl net.link.ether.bridge=1
# sysctl net.link.ether.bridge_cfg="wi0,xl0"
# sysctl net.inet.ip.forwarding=1
```

Now it is time for the wireless card setup. The following command will set the card into an access point:

```
# ifconfig wi0 ssid my_net channel 11 media DS/11Mbps mediaopt hostap up stationname "FreeBSD AP"
```

The [ifconfig\(8\)](#) line brings the `wi0` interface up, sets its SSID to `my_net`, and sets the station name to `FreeBSD AP`. The `media DS/11Mbps` sets the card into 11Mbps mode and is needed for any `mediaopt` to take effect. The `mediaopt hostap` option places the interface into access point mode. The `channel 11` option sets the 802.11b channel to use. The [wicontrol\(8\)](#) manual page has valid channel options for your regulatory domain.

Now you should have a complete functioning access point up and running. You are encouraged to read [wicontrol\(8\)](#), [ifconfig\(8\)](#), and [wi\(4\)](#) for further information.

**h) Feature description in the 802.11-1999 standard, and
in tutorial of the feature in an 802.11 tutorial**

81. Access points add a level of management and security which can segment the network and prevent nodes which cannot authenticate from connecting. This is described in detail in the standard **APP. B** page 26 section 5.7.5, page 29 section 6.1.2, and page 60-64 section 8.1.2-8.1.4.

82. **APP. N** explains specifically that any STA and AP can connect based on the standard. It is important to note that this is a clarification of the standard, specifically emphasizing that access points can communicate directly based on the protocol as it was defined by the standard in 1999. Based on the standard, it would be clear to a POSITA that an access point could connect to another access point as a station, since an access point is first and foremost a station (STA).

83. Moreover, the specific link architecture between access points (AP to AP) is also supported by the standard. This form of connectivity was based on the address fields which the 802.11 had defined precisely for that purpose. The following are some key excerpts from **APP. N** which highlight WDS (Wireless Distribution System) features which are part of the 802.11 standard. **APP. N** states: This document clarifies the term ‘WDS’ as defined and described in IEEE 802.11-1999, noting that the term is actually NOT defined and described and there is much confusion on this point. To prevent further confusion, the author

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

recommends reducing the use of or deprecating the term in favor of the underlying mechanism”. *Id.* at 1.

84. The 802.11-1999 standard (**APP. B**) defines the four address fields of an 802.11 packet in using two tables:

Existing references to “WDS” in IEEE 802.11-1999:

4. Abbreviations and acronyms

“WDS wireless distribution system”

7.1.3.1 Frame Control field

Table 2—To/From DS combinations in data type frames

To/From DS values	Meaning
To DS = 0, From DS = 0	A data frame direct from one STA to another STA within the same IBSS, as well as all management and control type frames.
To DS = 0, From DS = 1	Data frame exiting the DS.
To DS = 1, From DS = 0	Data frame destined for the DS.
To DS = 1, From DS = 1	Wireless distribution system (WDS) frame being distributed from one AP to another AP.

Table 2 indirectly references clause 7.2.2 and Table 4:

7.2.2 Data frames

The content of the Address fields of the data frame is dependent upon the values of the To DS and From DS bits and is defined in Table 4. Where the content of a field is shown as not applicable (N/A), the field is omitted. Note that Address 1 always holds the receiver address of the intended receiver (or, in the case of multicast frames, receivers), and that Address 2 always holds the address of the station that is transmitting the frame.

Table 4 – Address field contents

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	RA = DA	TA = SA	BSSID	N/A
0	1	RA = DA	TA = BSSID	SA	N/A
1	0	RA = BSSID	TA = SA	DA	N/A
1	1	RA	TA	DA	SA

APP. N at 2 (summarizing the standard, **APP. B**).

i) 802.11-1999 Standard - WDS 4-Address Format

85. **APP. B** (and in fact the first version of the standard published in 1997), clearly mentioned “Wireless distribution system (WDS) frame being distributed from AP to another AP.” **APP. B** at 7.1.3.1 (underline added).

86. **APP. B** further clarifies what was stated by the standard from its inception. According to **APP. B**, “WDS is simply a **mechanism** [emphasis in original] for constructing 802.11 frames using the 4-address format, as shown in the last row of Table 4 (shown above) and subsequently the last row of Table 5/710-T01 (shown below). The application of the four-address addressing type and procedures for operating STAs that uses that type of addressing are not defined or described by IEEE 802.11-1999. The 4-address format enables the implementation of devices that move MPDUs between various points, i.e. a relay type device. For example, the 4-address mechanism can be used to build some special products, like:

- a) A point-to-point (or bldg-to-bldg) bridge
- b) A wireless Access Unit (AU), aka repeater AU (i.e. an AU, which includes an AP, with a wireless backhaul connection to a real AU which in turn has a direct connection to the uplink network)
- c) An infrastructure mode STA with bridging capabilities,
- d) An 802.11 DS where the DSM is constructed from 802.11 STAs (the elusive, true, ‘wireless distribution system’).” **APP. N** at 3.

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

87. **APP. N** combines the original tables mentioned in **APP. B** while maintaining four possible 802.11 links ‘A’, ‘B’, ‘C’, ‘D’. The table specifically elaborate the details from Clause #5.6 of the 802.11-1999 Standard (see right column of the table)

July 2005

doc.: IEEE 802.11-05/0710r0

Explanation

Here is a new, extended table that shows all those aspects together:

Table 5/710-T01: ToDs/ FromDS, Address Fields and Entities

ID	Mode	ToDS	FromDS	# of Address Fields	Ext. Entity	802.11 Entity 1	Flow	802.11 Entity 2	Ext. Entity	802.11-1999 clauses
A	Ad hoc	0	0	3		ad hoc STA	↔	ad hoc STA		5.6
B	LAN access	0	1	3		infra mode STA	←	ACM_STA in an AP	xx	5.6
C	LAN access	1	0	3		infra mode STA	→	ACM_STA in an AP	xx	5.6
D	4A	1	1	4	yy	??	↔	??	zz	None

In case A both Entity 1 and Entity 2 are STAs operating in IBSS (aka ad hoc) mode.

Cases B and C describe an infrastructure mode STA accessing an external entity on the integrated non-IEEE 802.11 LAN through an AP and DS. The external entity is referred to as “xx”. Yes, an infrastructure mode STA could be transferring data to another infrastructure mode STA, but note that such traffic (from xx1 to xx2) transits the AP and also conceptually transits the DS.

Case D describes the 4 address scenario, referred to here as 4 Address (4A) mode, where the ultimate source/ destination addresses can be something other than the Entity 1/ Entity 2 TA/RA or RA/TA addresses.

4A mode can be used to construct a number of different types of devices, products and/ or systems.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

Analysis

Hence, “WDS” is simply a **mechanism** for constructing 802.11 frames using the **4-address format**, as shown in the last row of Table 4 and subsequently the last row of Table 5/710-T01. The application of the four-address addressing type and procedures for operating STAs that uses that type of addressing are not defined or described by IEEE 802.11-1999.

The 4-address format enables the implementation of devices that move MPDUs between various points, i.e. a relay type devices.

For example, the 4-address mechanism can be used to build some special products, like:

- a) a point-to-point (or bldg-to-bldg) bridge,
- b) a wireless Access Unit (AU), aka repeater AU (i.e. an AU, which includes an AP, with a wireless backhaul connection to a real AU which in turn has a direct connection to the uplink network),
- c) an infrastructure mode STA with bridging capabilities,
- d) an 802.11 DS where the DSM is constructed from 802.11 STAs (the elusive, true, “wireless distribution system”),
- e) and so on.

The IEEE 802.11-1999 standard does not define how to construct any such implementations or how stations interact to arrange for exchanging frames of this format, it merely defines the 4-address frame format that makes it possible. Neither does the standard define all the other supporting operations or protocols that would be required to build such devices.

APP. N at 3 (yellow highlighting added).

88. The table above discloses that the 4A Mode can be used to construct a number of different types of devices, products and/or systems. Row ‘D’ address structuring allows packets to be sent from source to destination through multiple access points along the route.

89. A POSITA would know that BSSs are uniquely identified by the MAC address of their access points, and that MAC addresses are globally unique. This satisfies the requirement for a unique “identification (APID)” mentioned in the Challenged Patents.

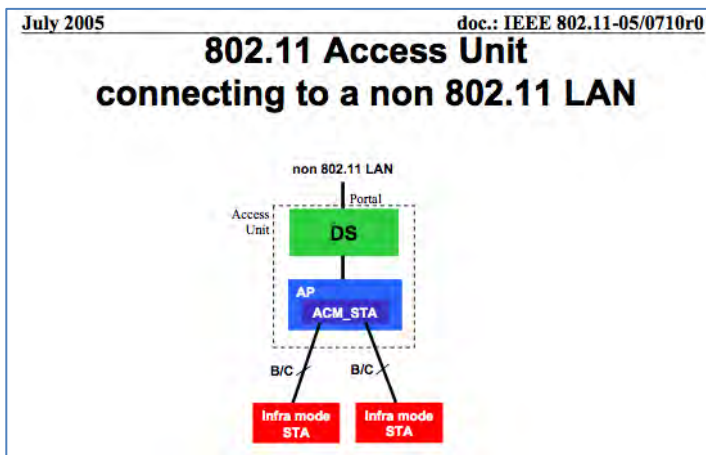
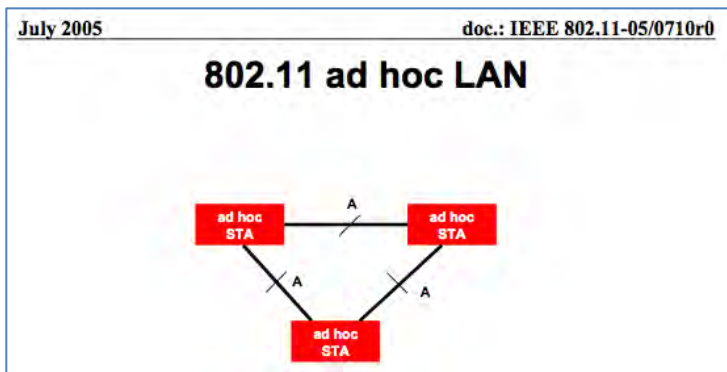
7.1.3.3.3 BSSID field

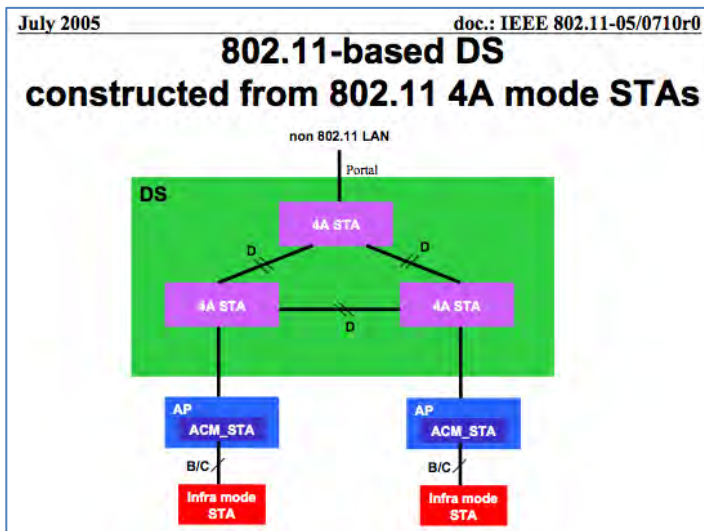
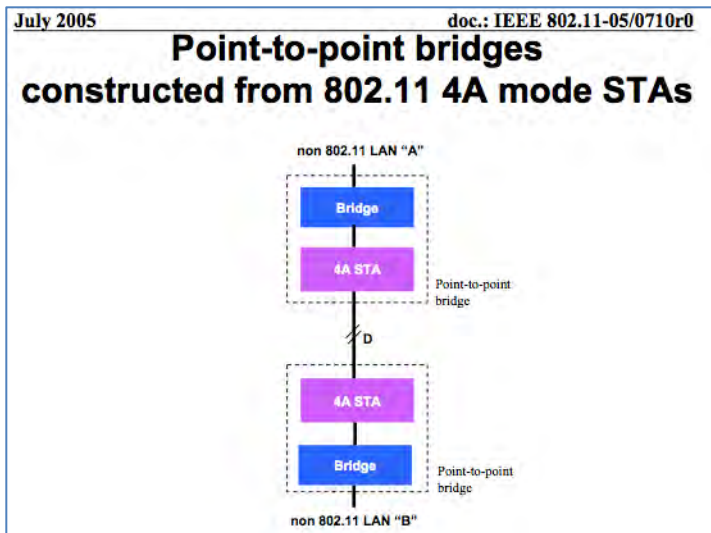
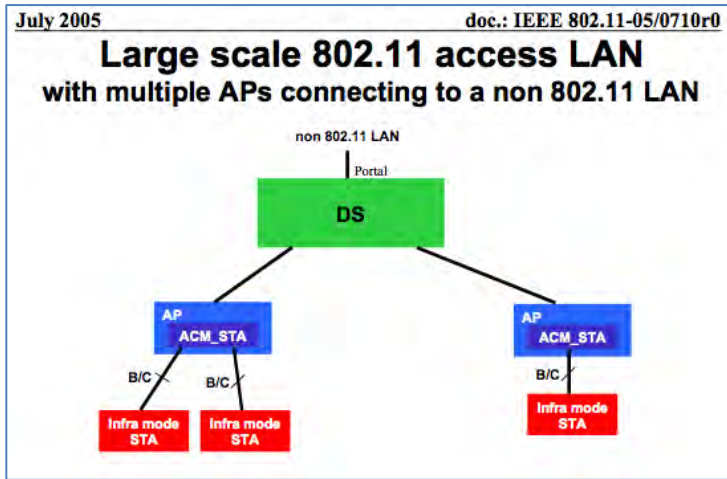
The BSSID field is a 48-bit field of the same format as an IEEE 802 MAC address. This field uniquely identifies each BSS. The value of this field, in an infrastructure BSS, is the MAC address currently in use by the STA in the AP of the BSS.

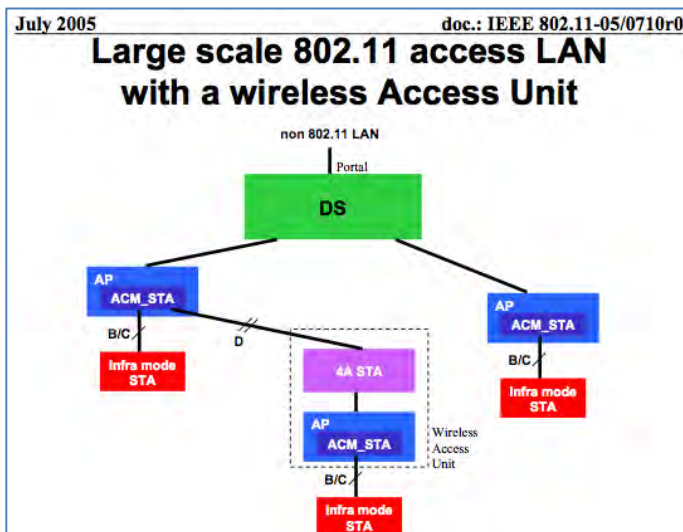
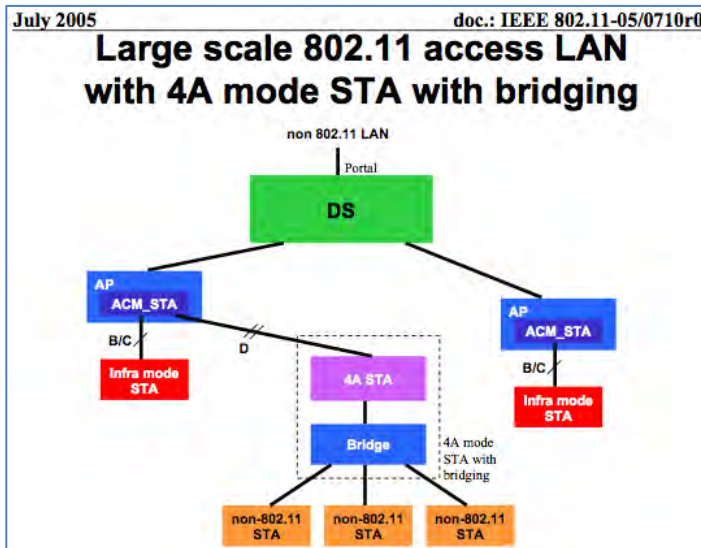
Source: **APP. B** page 39.

j) 802.11-1999 Tutorial – Using WDS 4-Address Format

90. **APP. N** shows seven sample implementations based on the four types of WDS address field uses. Note that the diagrams indicate the type of address field configuration with labels assigned to the links.







91. A POSITA would understand that the “Wireless Access Unit” in the figure above is a combination of an access point AP and a station STA.

92. A POSITA would understand that it is also possible to combine such topologies.

93. Based on the existing technologies it is clear that neither the connections between stations and computers functioning as access points, nor computers functioning as access points and other computers functioning as access points, that are described in the Challenged Patents, are either novel or inventive. These capabilities were foreseen by the standard years before the patents.

k) Basic implementation of Bridging/Repeating

94. Moreover, it is technically feasible to build the system described in the Challenged Patents without requiring WDS at all. In this case the system is a direct implementation of basic 802.11 capabilities where each computer acts as a local AP while sending packets as a station on interfaces which are not in the BSS from which a packet was received. A POSITA would know that this was and still is a basic bridging/repeating (and routing) logic used to forward packets to destination which are not in the local physical network segment.

l) Product implementations in the market

95. It was clear to manufacturers that these architectures could be supported by upgrading firmware and software of wireless devices. These mechanisms had already been used in products which supported wireless bridging and repeating were available on the market before the invention.

96. For example, D-Link AirPlus Enhanced Wireless Access Point⁵ with a wireless repeating function which was added in October 2002.

m) D-Link Air-Plus Enhanced 2.4Ghz Wireless Access Point (DWL 90AP+)



D-Link AirPlus Enhanced 2.4GHz Wireless Access Point (DWL-900AP+)	
Summary	TI-based 802.11b access point with '22Mbps' mode when used with other AirPlus products. Has bridging and wireless repeating features & can act as an Ethernet client adapter. No WEP enabled throughput penalty, best case speed only about 7Mbps.
Update	10/11/2002 Wireless Repeating function added in 2.20 firmware. See the related NTK.
Pros	<ul style="list-style-type: none">• Web browser administration with lots of features• Multiple bridging modes• Good performance with no WEP enabled throughput degradation• Can use other antennas• 2.20 firmware adds wireless Repeating!
Cons	<ul style="list-style-type: none">• You won't get 22Mbps

⁵ <https://www.smallnetbuilder.com/wireless/wireless-reviews/24589-dlinkdw1900apreview>

97. D-Links's AirPlus Enhanced Wireless Access Point is a "computing device" which also an "access point" which can connect to an "IP based network" as well as communicate wirelessly with other computing devices.

n) 3Com's Office Connect Wireless 11a/b/g Access Point

98. Another example is 3Com's OfficeConnect® Wireless 11a/b/g Access Point 3CRWE454A72⁶ which enabled building complete wireless networks based on WDS. *See APP. AG.* The figures below depict some bridging configurations of the 3Com's OfficeConnect®. The "WDS link" uses the 802.11 four-address format frame discussed in section V.A.1.e.

⁶ <http://www.endoftether.net/docsPages/manuals/3comWirelessAccessPoint-ConfiguringWDS.pdf>

TECHNICAL BRIEF

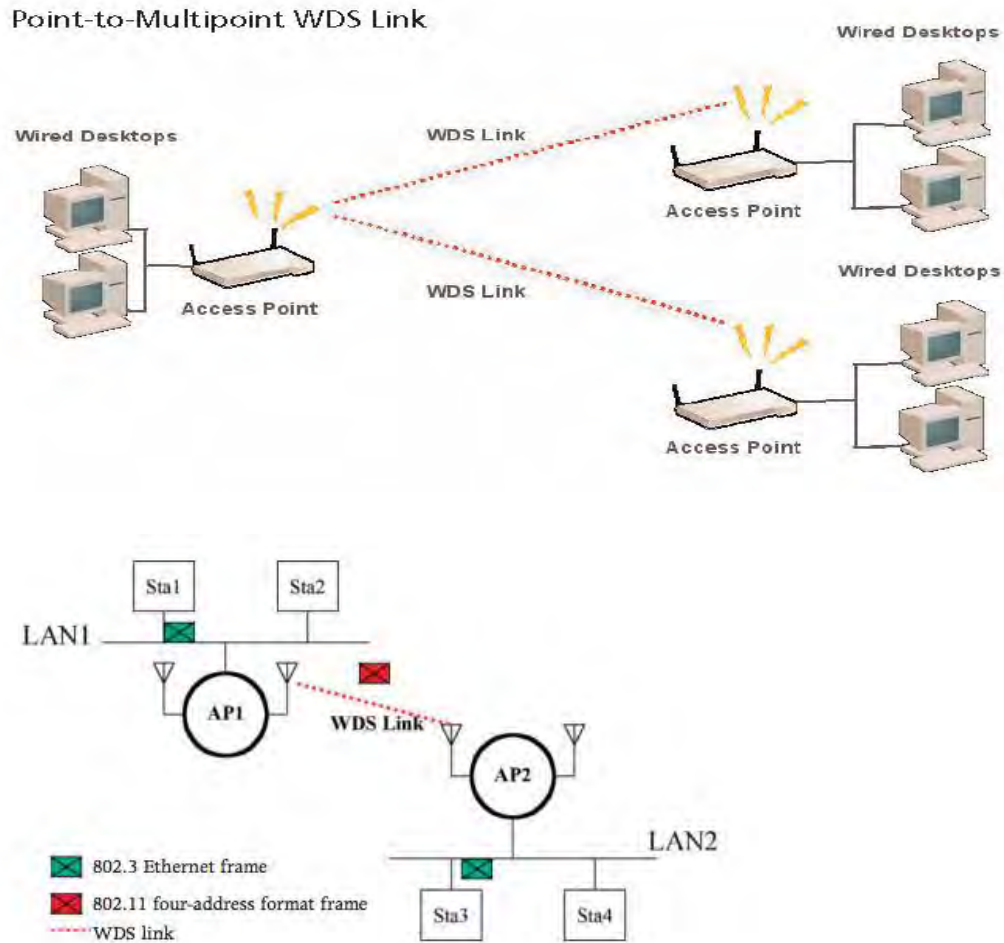
Configuring a Wireless Distribution System (WDS) with the 3Com® OfficeConnect® Wireless 11a/b/g Access Point

This document explains the WDS (Wireless Distribution System) features provided by the 3Com® OfficeConnect® Wireless 11a/b/g Access Point (3CRWE454A72). These features allow you to build a completely wireless infrastructure because the network equipment no

longer has to be connected to a wired LAN. Also, WDS features allow you to create large wireless networks by linking several wireless access points with WDS links. WDS is normally used in large, open areas where pulling wires is cost prohibitive, restricted or physically impossible.

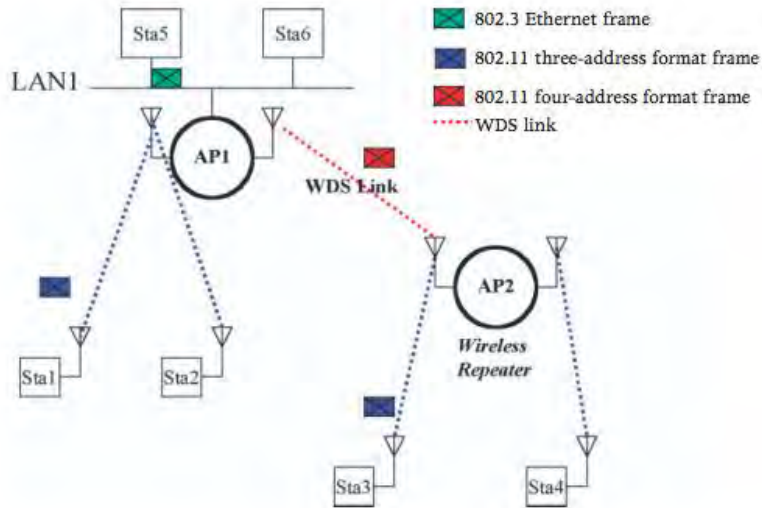
Point-to-Point WDS Link



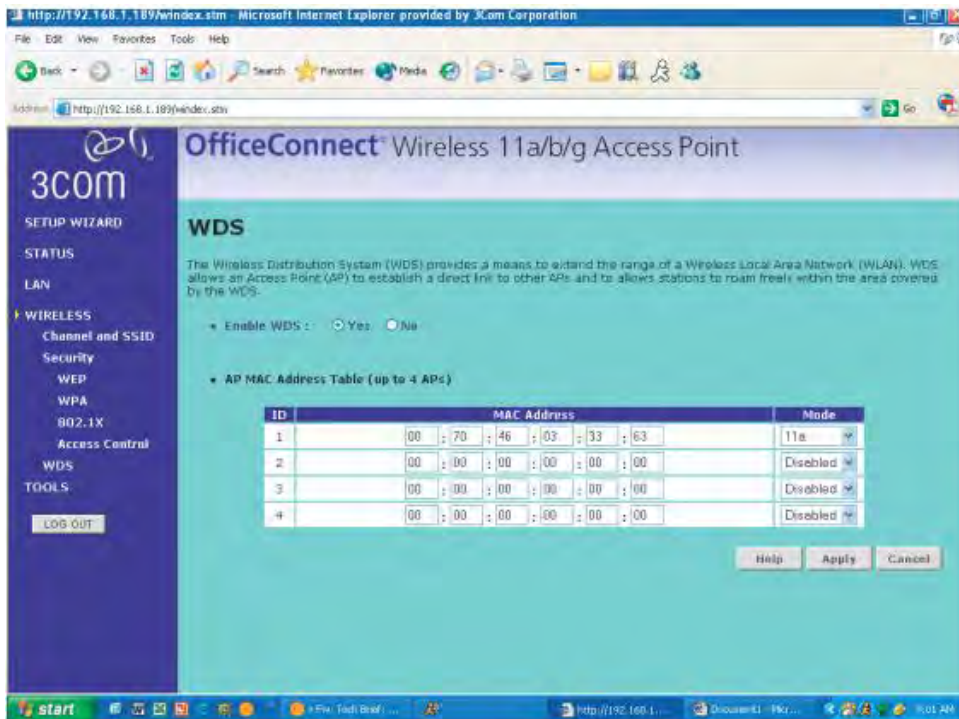


99. The following figure depicts a repeating configuration, in which 3CRWE454A72 connects wirelessly to an “**IP based network**” (STA1 and STA2) on one side and to another 3CRWE454A72 on another wireless link. The “WDS link” uses the 802.11 four-address format frame discussed in section V.A.1.e.

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306



100. The 3CRWE454A72 runs a web server which allows users to configure WDS links through a browser interface:



Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

101. 802.11 provides standard mechanisms for creating ad hoc and managed connections between clients and access points.

102. A computer with a wireless card and appropriate software can function as a station as well as an access point.

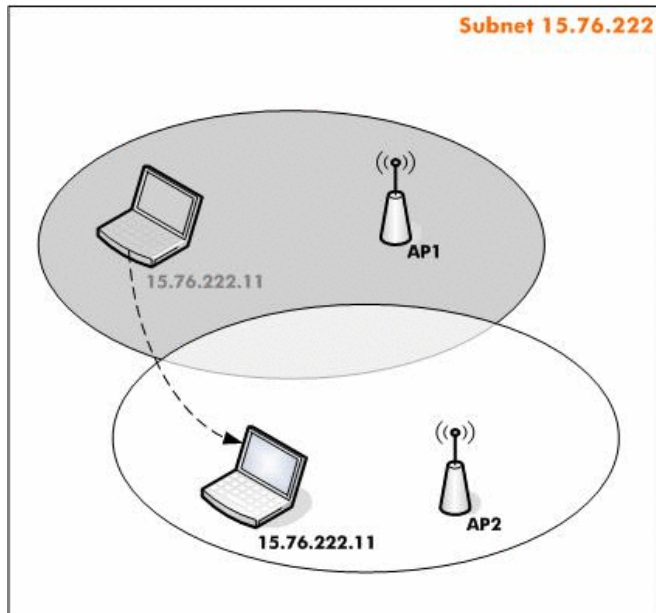
2. *Mobile IP*

103. Mobile IP is an IETF standard, and was well-known and often used architecture at the time of the Challenged Patents.

104. A common analogy to explain Mobile IP is when someone moves his residence from one location to another; say, from Boston to New York. The person drops off the new mailing address at the New York post office, which notifies the Boston post office of the new mailing address. When the Boston post office receives mail for a person, it knows to forward the mail to that person's New York address.

105. For instance, the figure below shows two access points—AP1 and AP2—within the same subnet 15.76.222.

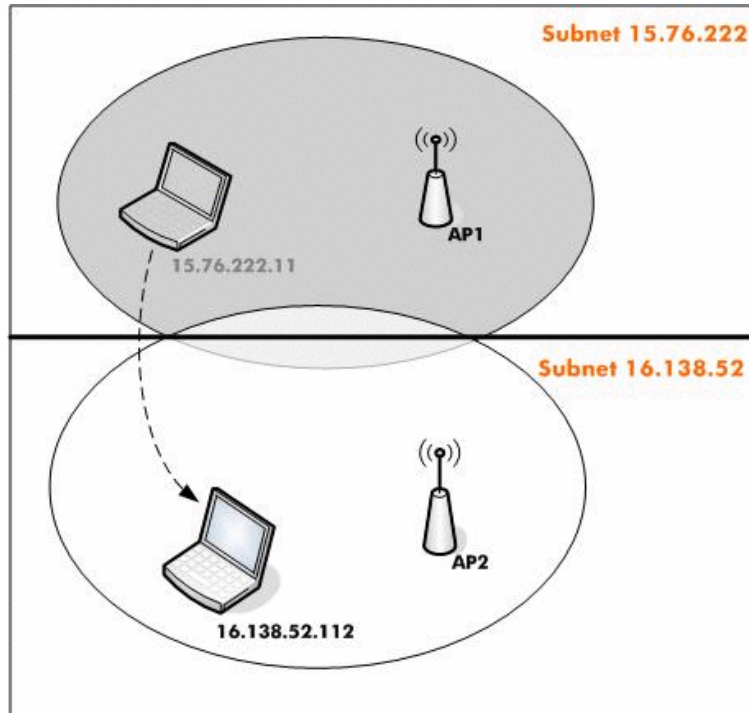
Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306



106. The WLAN client is initially associated with AP1 and assigned the IP address 15.76.222.11. The user issues a huge HTTP download and walks out of the coverage area of AP1 and is now associated to the new access point AP2. Because both the access points are serving the same subnet, the client's IP address is retained. The download continues without any hiccups. In this case, the roaming occurs at Layer 2. Layer 2 roaming is very fast and reliable across similar network media within the same subnet or network topology.

107. Take a case where AP1 is in subnet 15.76.222/24 and AP2 is in subnet 16.138.52/24, as illustrated in the next figure. The client is assigned 15.76.222.11 when connected to AP1.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

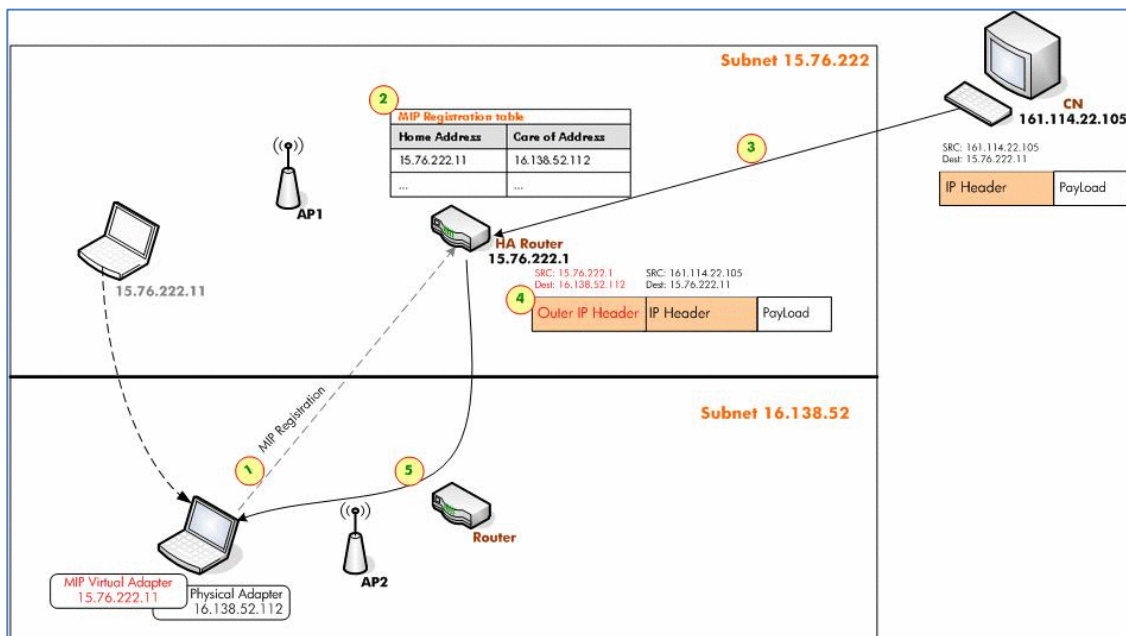


When it associates with AP2, it cannot retain the same address; hence, it gets a new address 16.138.52.112 from the 16.138.52 subnet's DHCP server. The HTTP download is a TCP session breaks because the source address changed from 15.76.222.11 to 16.138.52.112. The question is how to retain the address 15.76.222.11 when a mobile node moves to the new subnet 16.138.52 so that the existing application sessions are not disrupted. Mobile IP solves this problem by introducing two new network entities called “**Home Agent**” (HA) and “**Foreign Agent**” (FA). The HA is typically in a gateway router at the home network, and a FA is typically a gateway router in the foreign network.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

108. The mobile node is any host (e.g. laptop) running the Mobile IP client. The mobile node is always assigned one permanent IP address; in this case, 15.76.222.11. In Mobile IP terminology, the permanent address is known as the “**home address**” and the subnet 15.76.222 is the “**home network.**” The mobile node implements a virtual network adapter that is assigned to the home address. All application layer packets are routed via this adapter. These packets are then sent over the physical WLAN interface, which is assigned 16.138.52.112. The WLAN adapter's address changes depending on the network it is connected to and this temporary address is called the “**care-of**” address.

109. The following figure explains how Mobile IP works:



The default gateway acts as the HA. The mobile node notifies its **new care-of address** by sending a Mobile IP registration packet. The registration packet is sent over UDP at port 434 of the HA.

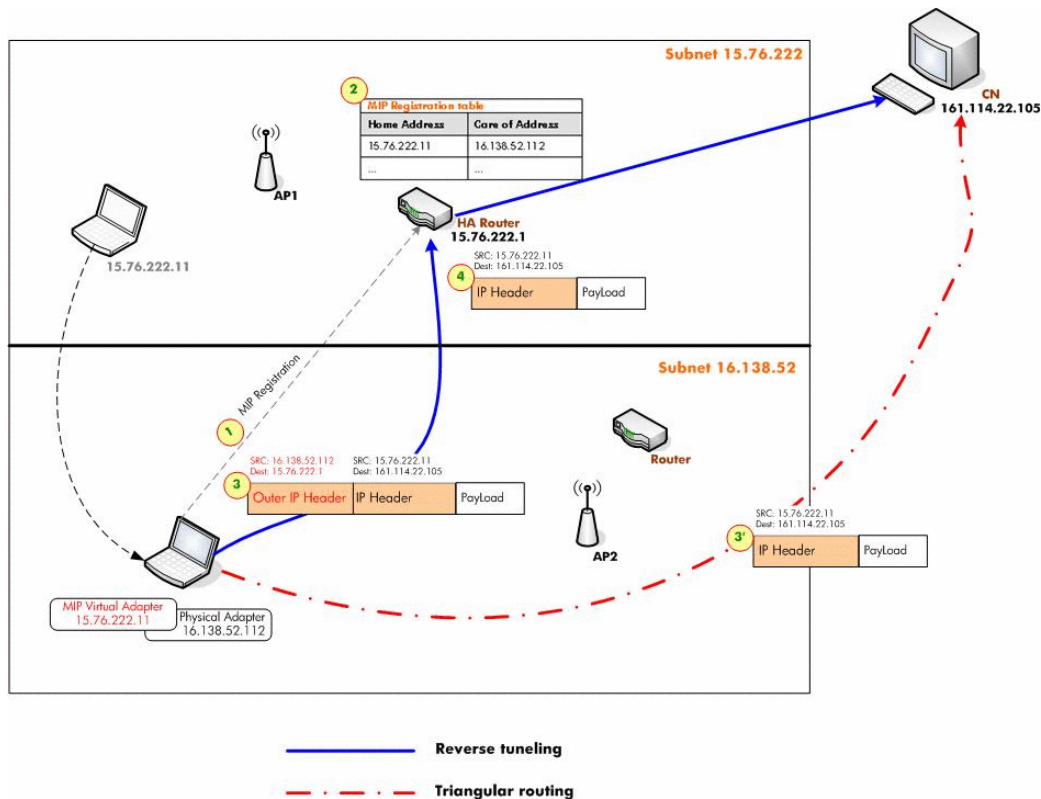
110. The HA sends back the reply after updating its registration table. The registration table consists of home address and care-of address mapping. The packets sent by the host running the web server known as the “**correspondent node**” (CN) are intercepted by the HA. These packets are then sent to the care-of address 16.138.52.112.

111. The HA encapsulates the original IP packet sent by the CN in another IP packet with the destination address as the care-of address 16.138.52.112 and the source address as the HA address 15.76.222.1. The process is known as “**tunneling**” (see APPS. U-W).

112. The mobile node's WLAN adapter receives the tunneled packets, extracts the original packets, and passes them to the virtual adapter. Because the host application interfaces only with the virtual adapter, it is transparent to changes in the care-of address. When the mobile node goes back to the 15.76.222 network, it deregisters by sending a registration packet with a lifetime equal to 0. If it moves to another subnet, it registers its **new care-of address** and deregisters the previous one.

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

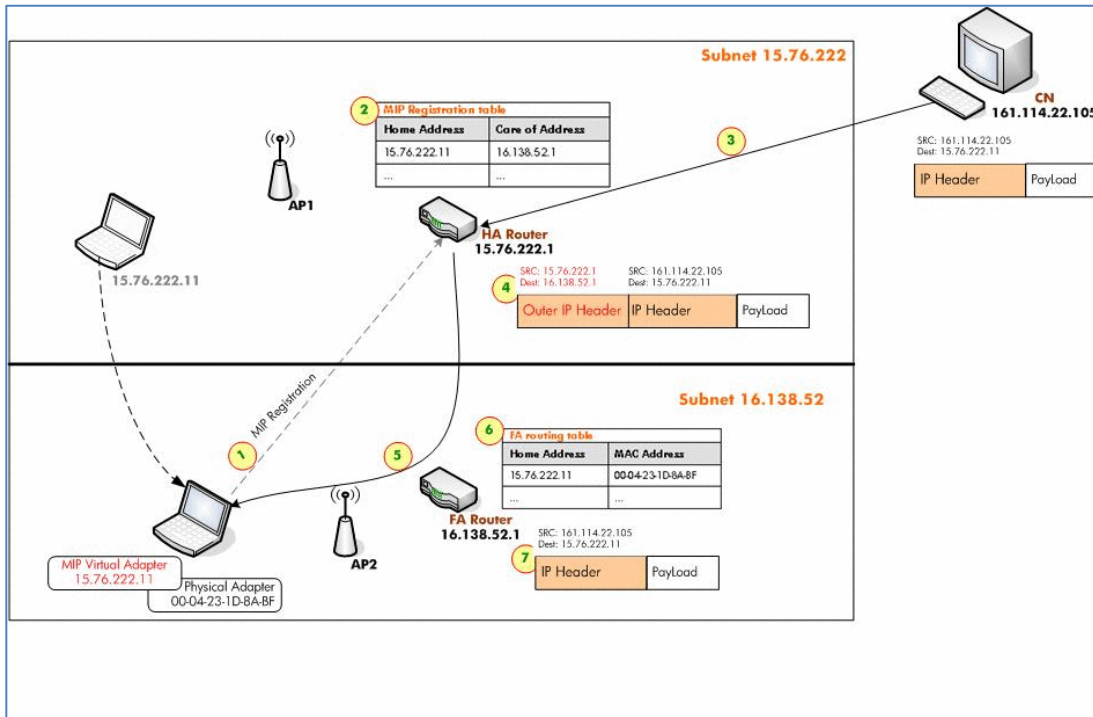
113. The mobile node (can send the packets directly to the CN without tunneling as shown below:



114. In the presence of an FA, each mobile node is not assigned a care-of address. Instead, the FA's address is used as the care-of address in the registration table and the mobile node registers with the HA via an FA. The packets sent from the CN are tunneled via the HA and sent to the FA. The FA decapsulates them and sends the original packets directly to the mobile node's WLAN adapter MAC address. FA maintains a routing table with the mobile node's home address and

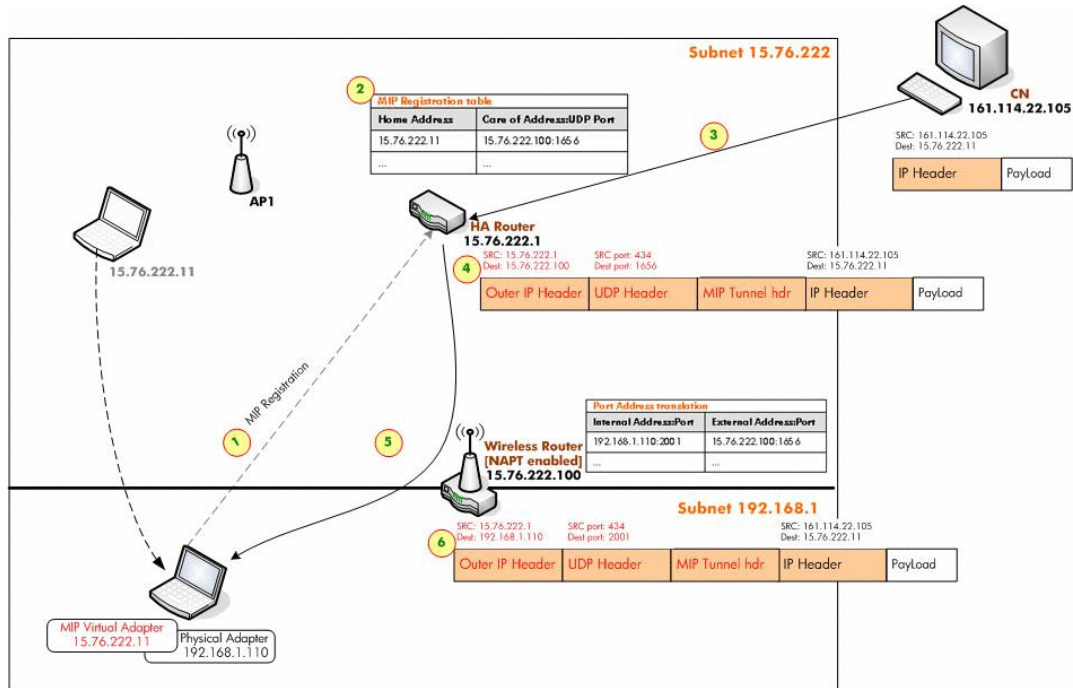
Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

the WLAN adapter's MAC address as in the figure below. The mobile node detects an FA by listening to agent advertisements broadcast by the FA.



115. Mobile IP can work together NAT (network address translation) as described in **APP. Z**. An example is provided below.

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306



- The mobile node requests for IP-in-UDP tunneling when it sends the registration packet. The registration packet contains the tuple {192.168.1.110:2001, 15.76.222.1:434}.
- The router uses port address translation to map many internal addresses and TCP/UDP ports in the subnet 192.168.1 to a single external address 15.76.222.100. For the registration UDP packet, the router is mapped from 192.168.1.110:2001 to 15.76.222.100:1656.
- The HA receives the registration request with source IP 15.76.222.100 and a UDP port 1656. The HA maintains the registration table mapping home address 15.76.222.11 to the care-of address and UDP port 15.76.222.100:1656.
- When a packet addressed to the mobile node reaches the HA, it encapsulates the packet in another UDP packet and sends it to the port 1656 that was used for registration.
- Once the packet reaches the NAPT-enabled wireless router, it is forwarded to 192.168.1.110:2001.
- The mobile node decapsulates and extracts the original packet.

116. Mobile IP and its use were well known to the Challenged Patents, and discussed in the specification. *E.g.*, '306 Patent at 4:13-33.

117. The '306 patent describes the “care of” address as the fixed address⁷ at 4:19-20.

118. Mobile IP is also mentioned in Buddhikot and discussed in detail in Sharma.

3. Security: IPsec, SSH, Packet filters, VPNs, Proxy Servers

119. Security products have been well known since the early 1990s. By the time of the invention it is hard to envision an organization large or small that did not deploy such technologies to shield internal networks from malicious access from external networks.

120. Packet filtering was available as part of FreeBSD from version 4 (March 2000) onward.

⁷ In my opinion, the '306 patent contains an error at 4:19-20 where it describes the “care of” address as the fixed address. The fixed address referenced is supposed to be the Mobile IP home address.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

FreeBSD Handbook
Chapter 26 Firewalls

26.4 The OpenBSD Packet Filter (PF) and ALTQ

As of July 2003 the OpenBSD firewall software application known as PF was ported to FreeBSD and was made available in the FreeBSD Ports Collection; the first release that contained PF as an integrated part of the base system was FreeBSD 5.3 in November 2004. PF is a complete, fully featured firewall that has optional support for ALTQ (Alternate Queuing). ALTQ provides Quality of Service (QoS) bandwidth shaping that allows guaranteeing bandwidth to different services based on filtering rules. The OpenBSD Project does an outstanding job of maintaining the PF User's Guide that it will not be made part of this handbook firewall section as that would just be duplicated effort.

The availability of PF for the various FreeBSD releases and versions is summarized below:

FreeBSD Version	PF Availability
Pre-4.X versions	PF is not available for any release of FreeBSD older than the 4.X branch.
All versions of the 4.X branch	PF is available as part of KAME.
5.X releases before 5.3-RELEASE	The <code>security/pf</code> port can be used to install PF on these versions of FreeBSD. These releases were targeted to developers and people who wanted a preview of early 5.X versions. Upgrading to 5.3-RELEASE or newer versions of FreeBSD is strongly recommended.
5.3-RELEASE and later versions	PF is part of the base system; Do <i>not</i> use the <code>security/pf</code> port on these versions of FreeBSD. It will not work. Use the <code>pf(4)</code> support of the base system instead.

More info can be found at the PF for FreeBSD web site: <http://pf4freebsd.luv2party.net/>.

121. Checkpoint pioneered firewall technology, releasing its first commercial product Firewall-1 in 1993. Checkpoint later released VPN-1 which included stateful inspection (U.S. Patent 5,606,668). By the time of the invention Firewall-VPN technology from Checkpoint was available on Windows, Solaris, and Red Hat Linux.⁸

122. Cisco introduced its first PIX firewall in 1994. PIX products supported network address translation NAT and IPsec virtual private network VPN functions years before the invention. For example, PIX 506 was released in 2000 and supported among others (red highlighting added to the figure below):⁹

⁸ https://en.wikipedia.org/wiki/Check_Point_VPN-1

⁹ <http://www.cisco.com/c/en/us/support/security/pix-506-firewall/model.html>

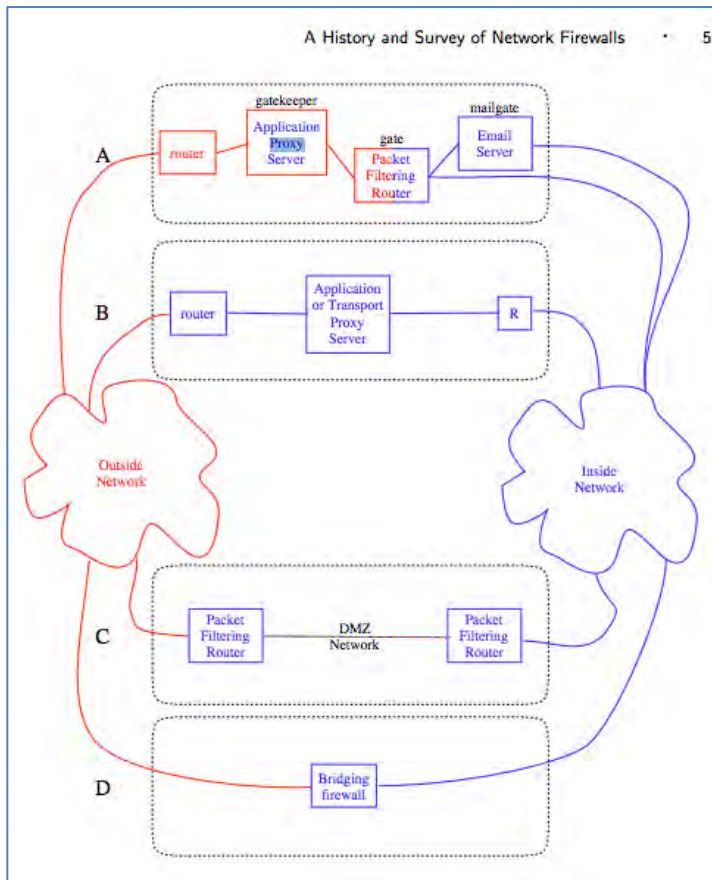
Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

Configuring a GRE Tunnel over IPsec with OSPF
Configuring a IPsec Tunnel - Cisco Secure PIX Firewall to Checkpoint 4.1 Firewall
Configuring a IPsec Tunnel Between a Cisco Secure PIX Firewall and a Checkpoint NG Firewall
Configuring the Cisco VPN Client to Tunnel to Two Remote Sites Through One Hub PIX
Configuring the PIX Firewall and VPN Clients Using PPTP, MPPE and IPsec
Configuring the PIX Firewall to Send Authenticated Usernames to a Websense Server

123. Proxy servers have been part of secure architectures for as long as security products have been available. A proxy is a program that receives the traffic destined for another computer. Proxies sometimes require user authentication; they then verify that the user is allowed to connect to the destination, and then they connect to the destination service on behalf of the user.

124. Firewall architectures are depicted in the figure below.

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306



125. Source: **APP. X**, original colors. Red indicates ‘outside networks.’ Blue indicates internal networks.

126. **APP. X** references many more sources which were published prior to the invention (see **APP. X** References) including firewalls, securing network perimeters, packet classifiers, protection from denial of service attacks, firewall friendly FTP, security in the Internet architecture, IP proxies, Cisco PIX, packet filtering in IP routers, ICMP tunneling, IPsec, NAT, transport level proxies, enhanced network security with packet filters, software bridges with packet

filtering, SOCKS protocol, toolkit for internet firewalls and intrusion detection and prevention.

a) Thirty-eight IETF RFCs that specify aspects of IPsec, encryption and tunneling, all of which were well known before 2006

127. The Internet Protocol Security (IPsec) defines the security services at the network layer by enabling a system to select required security protocols, determine the algorithms to use for the services, and put in place the cryptographic keys required to provide the requested services. I have provided a references to thirty-eight IETF RFCs that specify aspects of IPsec, encryption and tunneling, all of which were well known before 2006. These are just the tip of the iceberg of the huge body of knowledge and practices regarding IPsec and its included technologies (e.g. IPsec AH, IPsec ESP, IPsec IKE, IPsec ISAKMP – all mentioned in the RFCs). IPsec is a well-known network protocol that is used to establish secure tunnels from a client to an end-point gateway. To tunnel, a client encapsulates a packet and sends it to the end-point gateway. EX1017 at p. 3-5, §3.2.1; EX1018; EX1019 at p. 10, §3.2; EX1016 at 3:54-60, 4:27-21 (stating that IPsec is a standard protocol suite that encapsulates and tunnels traffic).

b) Packet encapsulation (tunneling)

128. Packet encapsulation (tunneling) works by: (1) taking an original outgoing packet having an original header, destination, and source address; and (2)

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

encapsulating the entirety of the original packet as a new packet with a new header. *See also* EX1016 at 4:29-39; EX1014 at 96-108. The header of the encapsulated packet specifies the end-point gateway as the destination address for the packet. *See also* EX1016 at 4:29-39. At the end-point gateway, the packet is de-capsulated and forwarded onto the original destination address (*i.e.*, acts as a proxy of the client). There could be multiple levels of encapsulation.

129. As an example, a packet intended for Google.com (IP address of 216.58.193.206) would be encapsulated as a new packet with a header addressed to a tunnel end-point gateway (Example IP address 201.1.2.3). The packet would be tunneled to the end-point gateway at 201.1.2.3 and the end-point gateway would de-capsulate the packet. De-capsulation removes the proxy's (outer) destination address 201.1.2.3, leaving the final destination address of Google. The proxy then sends the packet to Google.com with the address 216.58.193.206. The packet could be encrypted at the client to be decrypted at the end of the end-point gateway, the intended recipient (Google.com), or encrypted twice requiring a third intermediary to decrypt the data.

130. Sharma also reviews IPsec and SSH: "*The internet Security Protocol (IPsec) is a suite of protocols designed to provide security services for the Internet Protocol (IP).*" EX1016 at 4:42-44. "*The IPsec protocol suite provides an*

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

architecture with the overall pieces.” Id. at 4:54-55. “The particular protocol this invention uses for public key exchange and encrypting channels is the Secure Shell (SSH) protocol now being standardized by the Internet Engineering Task Force (IETF). The protocol is described collectively in the IETF draft Request for Comment documents: draft-ietfsecsh-architecture-12 .txt.” Id. at 6:13-19. “The invention uses the Secure Shell Protocol to effect automatic configuration for both Mobile IP and IPsec.” Id. at 6:41-43.

131. Packet filtering is discussed in Buddhikot: *“This embodiment uses dynamic MAC-address based filtering in the gateway 40.”* Buddhikot at 9:15-16. *“The first HTTP access is intercepted by the packet filter 223.” Id. at 9:45-46.*

132. **APP. Y** provides examples of how packet filters (access lists) are configured on Cisco IOS:

Feature History for the IP Access List Entry Additions Feature	
Release	Modification
12.2(14)S	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
12.3(2T	This feature was integrated into Cisco IOS Release 12.3(2)T.

APP. Y at 1.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

IP Access List Process and Rules

- The software tests the source or destination address or the protocol of each packet being filtered against the conditions in the access list, one condition (**permit** or **deny** statement) at a time.

Cisco IOS Release 12.2(14)S, 12.2(15)T, and 12.3(2)T

2

IP Access List Entry Sequence Numbering

Information About IP Access Lists

- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.
- If a packet and an access list statement match, the rest of the statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.
- If the access list denies the address or protocol, the software discards the packet and returns an ICMP Host Unreachable message.
- If no conditions match, the software drops the packet. This is because each access list ends with an unwritten or implicit **deny** statement. That is, if the packet has not been permitted by the time it was tested against each statement, it is denied.
- The access list must contain at least one **permit** statement or else all packets are denied.
- Because the software stops testing conditions after the first match, the order of the conditions is critical. The same **permit** or **deny** statements specified in a different order could result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by name in a command, but the access list does not exist, all packets pass.
- Only one access list per interface, per protocol, per direction is allowed.
- Inbound access lists process packets arriving at the router. Incoming packets are processed before being routed to an outbound interface. An inbound access list is efficient because it saves the overhead of routing lookups if the packet is to be discarded because it is denied by the filtering tests. If the packet is permitted by the tests, it is then processed for routing. For inbound lists, **permit** means continue to process the packet after receiving it on an inbound interface; **deny** means discard the packet.
- Outbound access lists process packets before they leave the router. Incoming packets are routed to the outbound interface and then processed through the outbound access list. For outbound lists, **permit** means send it to the output buffer; **deny** means discard the packet.

APP. Y at 2-3.

4. DHCP and NAT part of TCP/IP Suit

133. Dynamic Host Configuration Protocol (DHCP) enables network administrators to automate the assignment of IP addresses in a network. DHCP was well known and widely adopted in the industry more than ten years prior to the invention. See APP. R for a list of RFCs. DHCP provides both public and

private IP addresses. NATs can set aside a set of public IP addresses, which they translate from a set of private IP addresses. RFC2663 at p. 11, §4.1.1.

134. NAT was the well-known “network address translation” service that provided mapping for when private IP addresses were allocated to devices that needed to reach an outside network. A NAT can provide a many-to-one mapping can occur where many private IP addresses are mapped to a single public IP address. A NAT can also provide many-to-many mappings where many private IP addresses are mapped to different public IP addresses, where there could be a 1-1 correspondence for each private-public IP address.

B. Summary of the Challenged Patents

1. Specification of the Challenged Patents.

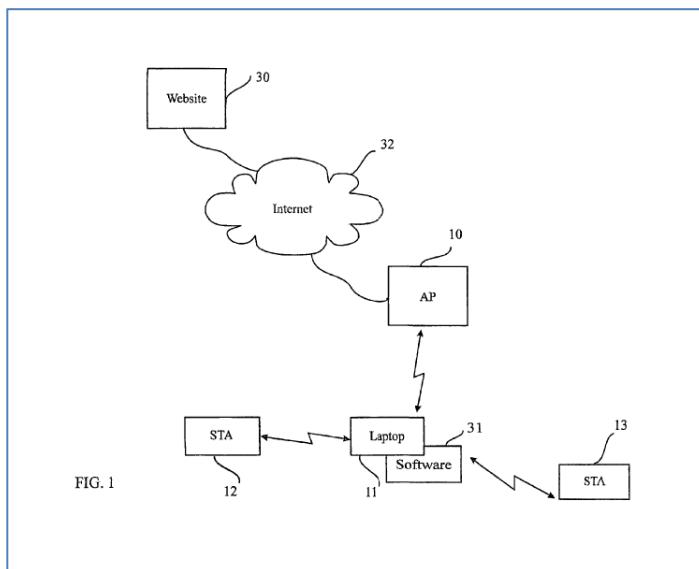
135. The Challenged Patents have identical specifications, so I will address them together. I typically reference the '306 Patent specification for particular citations, but any discussion of either patent's specification should be presumed to apply equally to both patents.

136. The Challenged Patents describe a method for enabling a WiFi-enabled device (STA) to connect to the Internet through a second WiFi-enabled device. The first STA is connected to the Internet through an access point (AP). This access point has a unique service set identified (SSID). The first STA uses

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

dedicated software that the user installs to enable it to function as a second wireless access point. The second access point has a unique SSID. The second STA connects to the AP created on the first STA. This connection enables the second STA to connect to the Internet through the first STA. '306 Patent at Abstract; 5:58:63.

137. The main idea is depicted in Fig. 1:



138. Fig. 1 illustrates “a wireless system for connecting mobile devices to the Internet through an access point. It may use a novel method for performing the deployment of APs, i.e., the method that allows devices to function at the same time as STAs and as APs. For example, a laptop 11 is connected to the Internet through access point AP 10, and at the same time, laptop 11 shares its connection

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

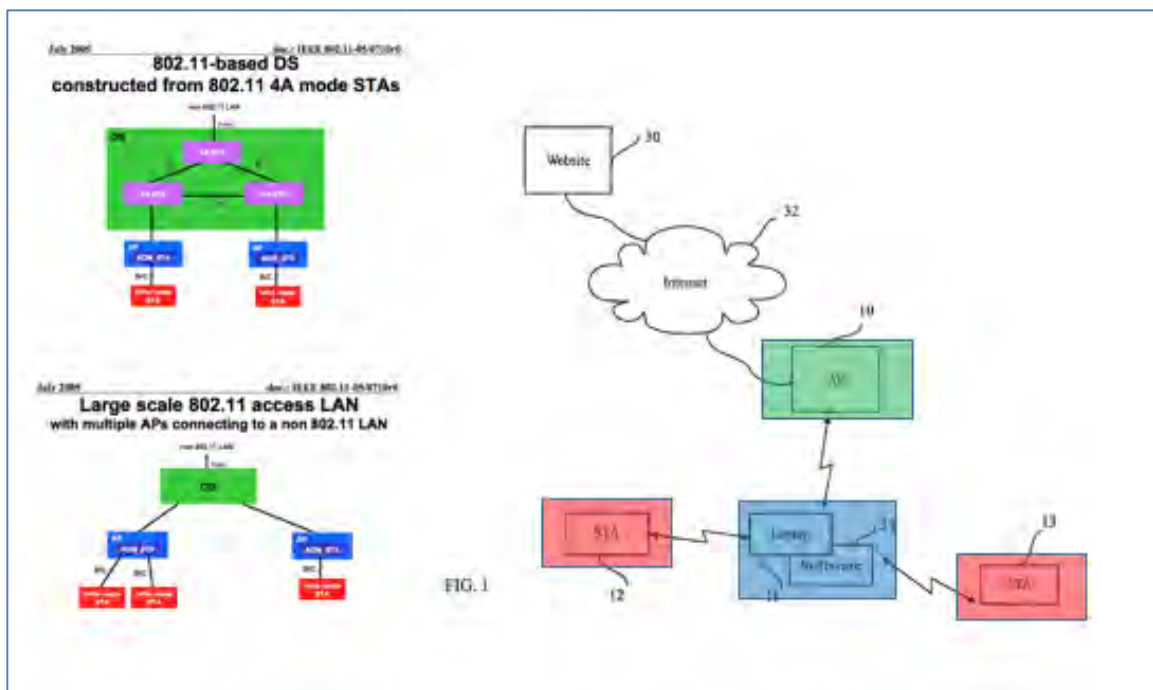
for other STAs by operating as an AP. Thus, other STAs 12 and 13 look at laptop 11 as an AP, and can connect through it to the Internet.” Id. at 11:33-42.

139. Trademarked software called Vagabee is used. “*STAs can download the Vagabee™ software, Vagabee software includes the functionality of the software of the first STA, to open new Aps.*” *Id. at 6:2-4.*

140. Laptops are the “*preferred embodiment*” for running the software. *Id. at 6:28;12:3-4.* However, the Challenged Patents do not specify any hardware requirements for the laptops which implement the invention. A POSITA would understand that the hardware, software and firmware would have had to be compatible with the “Vagabee” software. Moreover, a POSITA would understand that in terms of 802.11 capabilities, this software does not add new functionality. APs could connect to other network elements (switches, routers, gateways) before the invention, and STAs could connect to APs – both are familiar elements of 802.11 standards (see **APP. B, APP. N**).

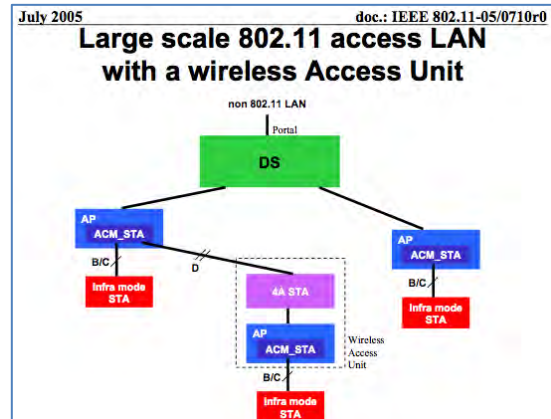
141. Based wireless distribution system (WDS) features it would be possible to represent the topology in Fig. 1 in the following manner: the DS is in bottom left corner of the diagram could be constructed from one or more access points or bridges/repeaters, one of which is connected to a none wireless network including a connection to an Internet link. The ‘laptop’ is a combination of what

APP. N shows as combinations of ‘ACM_STA’ and ‘4A STA’, which is a logical combination of common elements enabled by software running on the laptop using a standard wireless card which is compatible with the operating system. There is nothing which requires a leap of inventiveness in this construction. It has been identified by **APP. B** as what it calls a ‘wireless access unit’ (e.g. slide 10).



142. The diagram shows two topologies from **APP. N** on the left (original colors) vs. FIG. 1 of the Challenged Patents on the right. Highlighting added to Fig.1 to indicate familiar elements found in **APP. N**.

143. The building block of the Challenged Patents is the ‘laptop’ and the ‘software’. This is the combination of access point (ACM-STA) and station (4A STA) which **APP. B** identifies as ‘wireless access unit’. The Challenged Patents yield very predictable results based on the common elements and capabilities defined and recommended by the standard.



144. The figure above depicts a fundamental property of networks which applies to wireless networks described in the Challenged Patents as well. Even the most complex networks are built from a finite number of familiar known elements. The Challenged Patents use familiar elements without applying any changes which require a leap of inventiveness.

145. Fig. 2. of the Challenged Patents would be represented very similarly, only one or more STAs would be replaced by the ‘blue’ access point component which could be a laptop running “Vagabee” software, or other 802.11 bridging/repeating/signaling software. This is very similar to the FreeBSD computer. It would not require a leap of inventiveness to see how the “Vagabee” software which enables many of the functions of the Challenged Patents can be

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

developed using the FreeBSD components described in the technology section
V.A.1.c.

146. Again I emphasize that based on the existing technology a computer with a wireless card could connect a station to a second computer functioning as an access point, while the computer functions as an access point in its own BSS. The existing technology also enabled a computer to connect as an access point to the second computer. This mode is highlighted in the figure below.

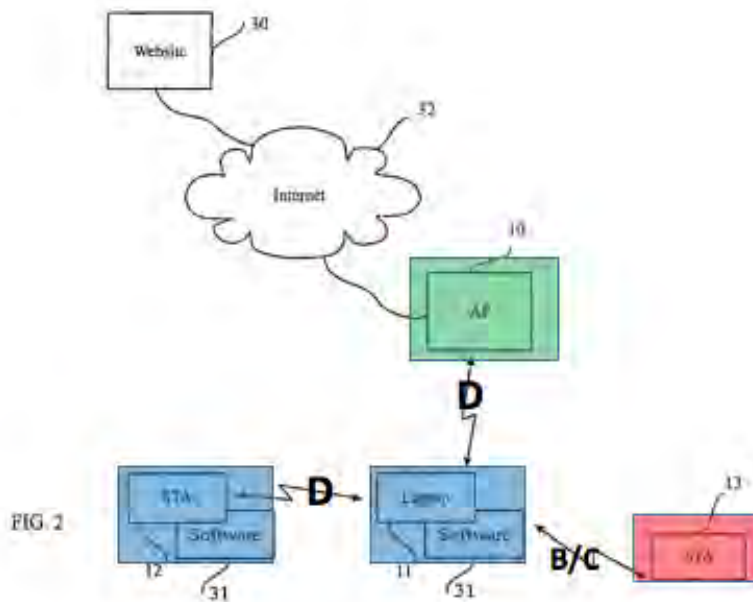


FIG. 2. Highlighting added to indicate functions which are mentioned in the standards: blue indicates an access point AP, red indicates and STA, green indicates a distribution system DS. The letters indicate the types of address field configurations as they are defined by the standard (APP. B, APP. N).

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

147. A POSITA would understand that connecting any number of APs to one another to form a path is a reuse of a familiar element defined by the 802.11 standard which allows APs to be connected to one another. In fact, it is this fundamental linking ability which enables networks to be built from familiar elements of smaller networks and devices.

148. It is also possible to configure “laptop 11 [so that it] will run a Network Address Translation (NAT) and a DHCP server as part of his protocol stack. Running DHCP enables laptop 11 to provide an Internet address to STAs that connect to it. Running a NAT allows laptop 11 to connect other STAs through it, while keeping conformance with regards to AP 10-To AP 10 all the communication appears to be originating from laptop 11.” ’306 Patent at 11:63-12:2.

149. The TCP/IP protocol suit was well known. Given the state of the technology at the time of the Challenged Patents, a POSITA would know that that this is a combination of familiar elements i.e. NAT and DHCP on the laptop.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

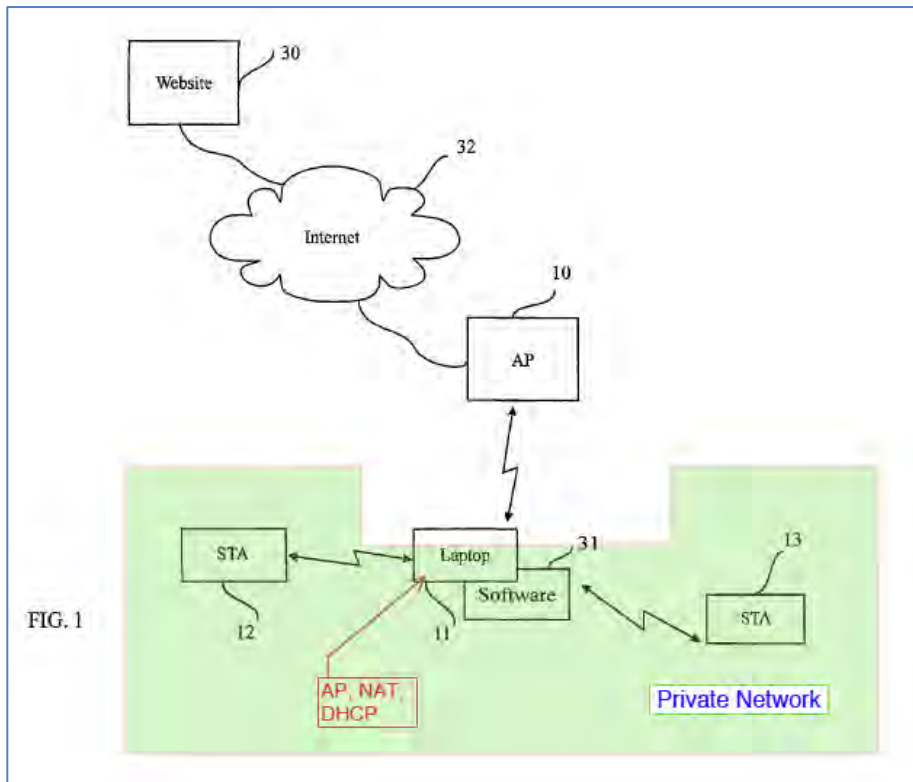


FIG. 1 – highlighting added to indicate the ‘private network’ and the software elements which enable this configuration.

150. This also addresses a security requirement: “To prevent STAs from accessing the inner network, laptop 11 blocks all traffic from the guest STAs to internal addresses 55 (i.e., addresses that appear only in local networks and not in the public internet, such as 192.168.*.*, or 10.*.*.*, and 172.16.0.0-172.31.255.255).” *Id.* at 14:54:57.

151. A POSITA would know that these are commonly used methods of packet filtering which were well known in existing technology.

152. Access within the private network can be restricted as well: “laptop 11 agrees to act as an APs, but it does not agree to allow STA 13 and STA 14 to access his inner network (i.e., it allows STA 13 and STA 14 to access the internet through his network but does not allow them to access computers in his network. For example, a private server 40 should not be accessible to them).” *Id.* at 14:37-42. This is shown in the highlighted and overlays of traffic paths added to Fig. 3 below.

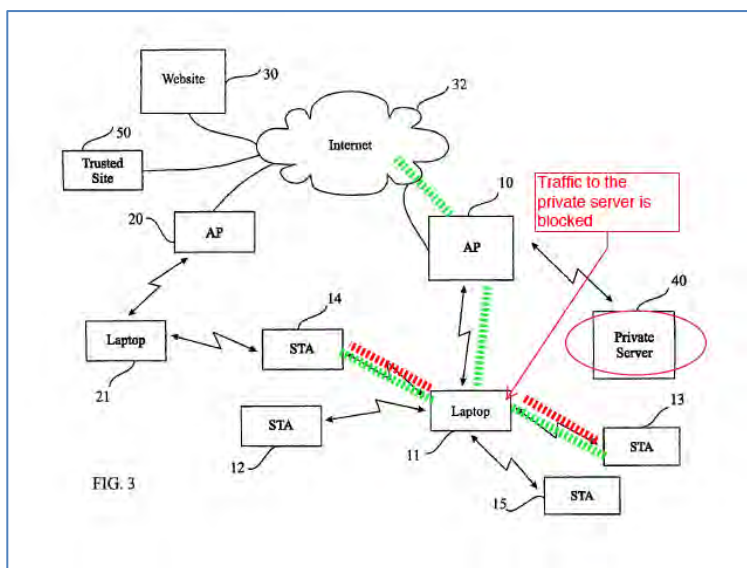


FIG. 3 – colored elements added: traffic is sent through the green paths, while ‘red’ traffic destined to the server is dropped.

153. Packet filtering at all levels of the protocol stack were well known technologies at the time of the invention. It would be obvious to a POSITA that

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

The filtering described in the Challenged Patents is an implementation of standard methods without any inventiveness.

154. “Devices function at the same time as STAs and as APs. This allows a STA to also create a new AP for connecting other STAs to the Internet there through.” *Id.* at 5:59:61. A POSITA would know that this is an implementation of WDS and is not novel. Moreover, it can be implemented with standard off-the-shelf software using a wireless card which is supported by the operating system as shown in the technology section (FreeBSD example).

155. “An STA can connect through two or more APs simultaneously.” *Id.* at 6:15; 12:57-58. The STA can switch connections by “first establish[ing] a new connection and then the old connection is terminated, practically leaving the STA connected.” *Id.* 6:23-25. This is referred to as “soft-handovers.” *Id.* at 9:36:39; 13:1-4. Fig. 3 shows STA 14 connected “through both laptop 11 and laptop 21.” *Id.* at 12:59:62.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

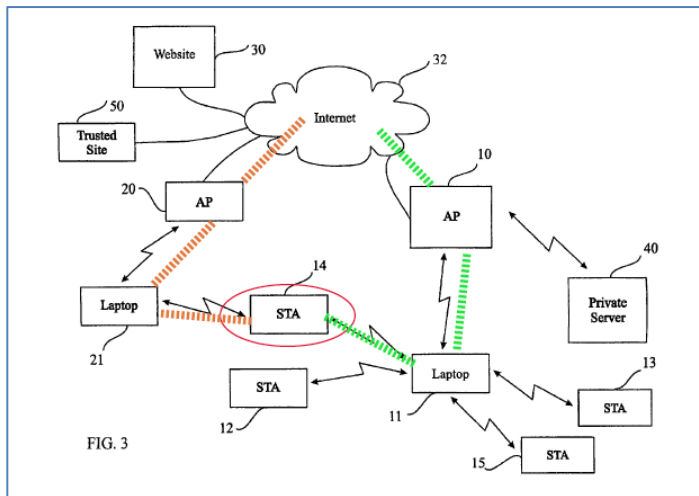


FIG. 3, red oval, green path and brown path highlighting added.

156. Connecting to one or multiple APs is a function of the 802.11 standard described above. There is nothing in the standard which precludes a station from associating with multiple access points. It does not require a leap of creativity to connect one STA to two APs. Moreover, using dual connections for redundancy is a well know best-practice is network design. This allows the network to remain connected if a link is disconnected for whatever the reason. Based on these principles it is common sense to switch connections in an order which prevents all links from being disconnected at any point in time. This is the same logic children apply when swinging between the rungs of monkey bars – letting go of one hand only after the other hand has a firm grip on the next rung.

157. “When laptop 11 and laptop 21 use the same WiFi channel, STA 14 connects to both laptops by creating two protocol stacks on the MAC (Media

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

Access Control) layer. When laptop 11 and laptop 21 operate on different channels, STA 14 agrees with laptop 11 and laptop 21 on period of times in which laptop 11 sends packets to STA 14, and periods of time 10 in which laptop 21 sends packets to STA 14.” *Id.* at 13:5-11.

158. These are standard capabilities of WDS, 802.11 and connectivity between devices in general. The ability to connect on different channels is part of the 802.11 standard. The list of channels available is defined by each country.

159. It is also possible to hide the data sent to and from an STA from the AP which it uses: “On the other hand, STA 13 wishes to use laptop's 11 network, but might not wish laptop 11 to be able to tap into the data that STA 13 exchanges with Internet servers. The current disclosure addresses these two problems using a novel method. First, external STAs are not allowed to access to the inner network by not allowing them to access to local IP addresses. Second, STA 13's privacy is protected by tunneling its sensitive traffic to a trusted network site 50, and STA 13 accesses the internet through its tunnel to the trusted network site 50, which acts as a proxy of STA 13.” *Id.* at 14:42-52; 15:1-14.

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

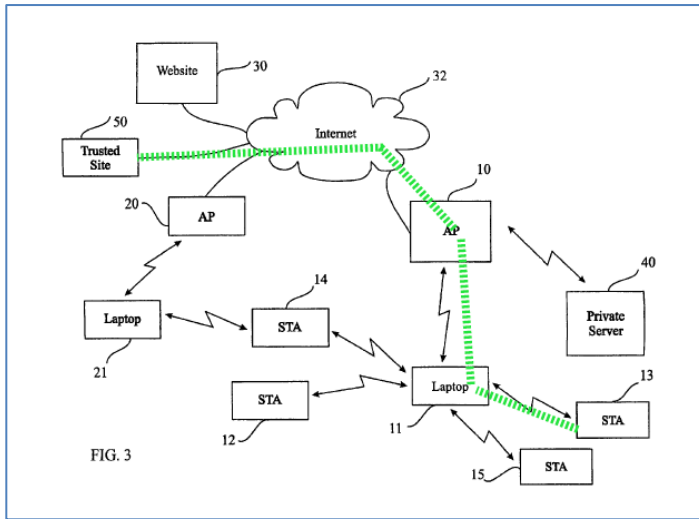
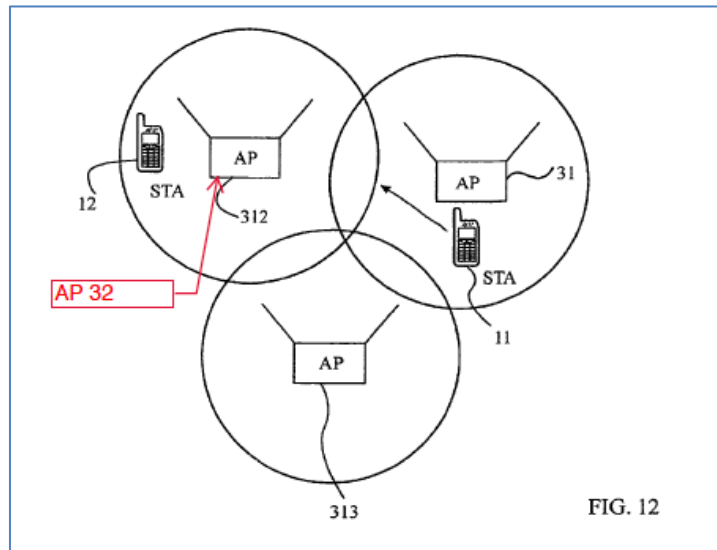


FIG. 3 – green tunnel overlay added.

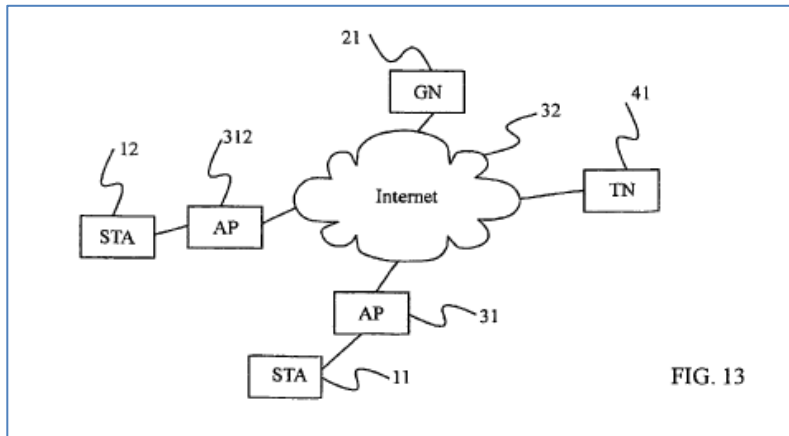
160. There is nothing novel about this feature. A POSITA would know that tunneling was (and still is) a mainstream expected method of sending traffic through devices and networks which should not be able to see the contents of the data. Using proxy servers was also well known and widely practiced years before the Challenged Patents.



161. Handover (switching a connection from one access point to another) can be done very quickly: “to practically almost complete the process of the

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

handover before it even started.” *Id.* at 18:59-60. An example is depicted in FIGs 12 and 13 (red marking added).



162. “STA 11 is in conversation with TN 41 (TN-Termination node, the node with which STA 11 communicates, shown in FIG. 13), and STA 11 is moving from AP 31 towards AP 32. Also assume that a node GN 21 (GN-Governing Node, a node that is non-exclusively responsible for the mobility management in a certain geographic area for a given time, shown in FIG.13) is in contact with STA 11, and it is assisting STA 11 during the hand over process. STA 11 currently has an IP address, which was allocated to it by AP 31. To complete the handover, STA 11 should be associated with AP 32, have an IP address assigned by AP 32, complete any second authentication that is required, and have TN 41 be aware of the new IP address, so it can forward the conversation to the new location.” *Id.* at 18:62-19:8; 19:21-20:61.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

163. This method of assigning addresses is the standard method which is used by Mobile IP to assign ‘care-of-addresses’ to devices as they move from one networks to another.

164. “We assume that the APs are divided into location areas, and for each location area there is a node in the network that is in charge of this location area. For example, assume GN 21 is in charge of a location area composed of AP 31, AP 32, and AP 33.” *Id.* at 26:19:23.

165. This is a standard feature of 802.11 where APs do not overlap in order to minimize interference and maximize the cost-effectiveness of the solution.

166. Handover coordination is done through a Governing Node (GN) which knows about all the APs in its area. A governing node can preset a connection between an AP and an STA which is moving into its area in the following sequence: STA 12 is instructed by the GN to “impersonate STA 11 towards AP 32... and complete a connection process with AP 32 on behalf of STA 11.” *Id.* at 19:25-27. STA 12 “communicates these parameters [IP address] to GN 21.” *Id.* at 19:33. “GN 21 communicates the parameters to STA 11 ... and thus, STA 11 does no longer need to perform the connection process.” *Id.* at 19:35-37.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

167. Indeed, the Challenged Patents make extensive use of a combination of familiar elements without making a leap of innovation and without providing significant increases in efficiencies or alleviating market pain points in a cost effective manner. The Challenged Patents provide negligible, if any, improvement to hand-over time, at the cost of replicating exiting methods and devices with proprietary software. Moreover, the Challenged Patents provide proprietary configuration methods which a POSITA would understand goes against mainstream skillsets related to designing and configuring DHCP, NAT, packet filtering, proxy servers, tunnels, VPNs and access points.

2. '306 Patent Claims.

168. The '306 Patent has 68 claims, of which three (1, 16, and 43) are independent. Other than the fact that Claims 1 and 16 are systems claims and Claim 43 is a method claim, the independent claims are all very similar to Claim 1. Likewise, the dependent claims of the '306 Patent repeat each other two or three times, depending on if they first appear after Claim 1 or Claim 16. As a result, only Claims 1-15, 17, 20, 23-25, 32-37, and 41-42 of the '306 Patent are really “unique,” and even some of these “unique” claims are very similar to one another.

169. Each limitation of the Challenged Patents' claims will be compared in detail to the prior art below, but in short, Claim 1 discloses a wireless hotspot device that acts as a wireless access point for other computing devices, while itself connecting wirelessly to another access point that allows for the hotspot device—which must itself have a user interface and display—and the other connected computing devices to access an IP-based network, like the Internet. The connecting devices use distinct public IP addresses, and their connection must have a secure tunnel to a proxy server in the IP-based network.

170. The focus of the '306 Patent dependent claims vary. Some emphasize that the hotspot device should connect over a cellular network. Others are concerned with security aspects of the wireless connections. Still others are concerned with the services that provide IP address allocation and mapping, like DHCP and NAT.

171. All of the '306 Patent claims will be compared to the prior art in detail below.

3. '369 Patent Claims.

172. The '369 Patent has 13 claims, of which three (1, 8, and 13) are independent. Claim 1 of the '369 Patent is practically identical to Claim 1 of the

'306 Patent, and Claim 1's dependent claims are very similar to dependent claims in the '306 Patent.

173. Independent Claim 8 of the '369 Patent is similar to Claim 1, except that it has additional disclosures about data buffering in the event of a lost connection.

174. Independent Claim 13 of the '369 Patent is concerned with the ability of the hotspot device to serve as multiple access points simultaneously.

175. All of the '369 Patent claims will be compared to the prior art in detail below.

C. Summary of the Prior Art References

1. Summary of Buddhikot

176. Buddhikot discloses "A gateway for mobile access includes a foreign agent that receives user profile data and session state data from a home authentication, authorization and accounting (AAA) system of a mobile node, and a dynamic packet filter that performs multi-layer filtering based on the user profile data. The foreign agent transfers a session from a first network to a second network without session interruption, using the session state data, when the mobile node moves from the first network to the second network. The packet filter permits Internet access by the mobile node without passing Internet data requested

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

by the mobile node through the first network.” Buddhikot at Abstract. This provides “an improved system for integrating 3G and 802.11 access.” *Id.* at 3:1.

177. In certain embodiments, Buddhikot teaches a MobileHotSpot Gateway 1440 that provides Wi-Fi connectivity for computing devices, allowing them to connect to the Internet via a wireless base station.

178. The motivation to connect between the local area wireless 802.11 and wide area wireless 3G is described: “802.11 Access Points can cover areas of only a few thousand square meters, making them suitable for enterprise networks and public hot-spots such as hotels and airports. On the other hand, wireless networks built using the 3G standards require significant capital investments, support limited peak rates that range from 64 Kbps to nearly 2 Mbps as a maximum, but offer a much wider area of coverage that enables ubiquitous connectivity. The deployment of architectures that allow users to seamlessly switch between these two types of network would present several advantages to both service providers and users. ...3G operators and wireless Internet Service Provides (WISP) could ... ultimately facilitate the ubiquitous introduction of high-speed wireless data.” *Id.* at 1:24-40.

179. “When the user travels to a place, such as an airport concourse, where there is such an 802.11 service provider, his machine should be able to

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

transparently switch to the 802.11 access. When the user leaves the coverage of the 802.11 provider, his machine should seamlessly switch to the 3G access.” *Id.* at 4:29-34.

180. Buddhikot responds to market pressures to seamlessly connect mobile users with wireless LAN access to the Internet using wide area 3G networks. The Challenged Patents solve a network configuration and network accessibility problem by combining existing technologies in a non-trivial manner to provide a complete solution which connects between wireless LANs and wireless WANs.

181. “The 802.11 provider desires a secure mechanism through which it can authenticate the user by interacting with the Authentication, Authorization and Accounting (AAA) server of the 3G carrier.” *Id.* at 4:40-43.

182. “When the switching occurs, the user may have several ongoing network sessions (e.g., network radio, voice chat. etc), and these sessions should be transparently maintained.” *Id.* at 4:43:46

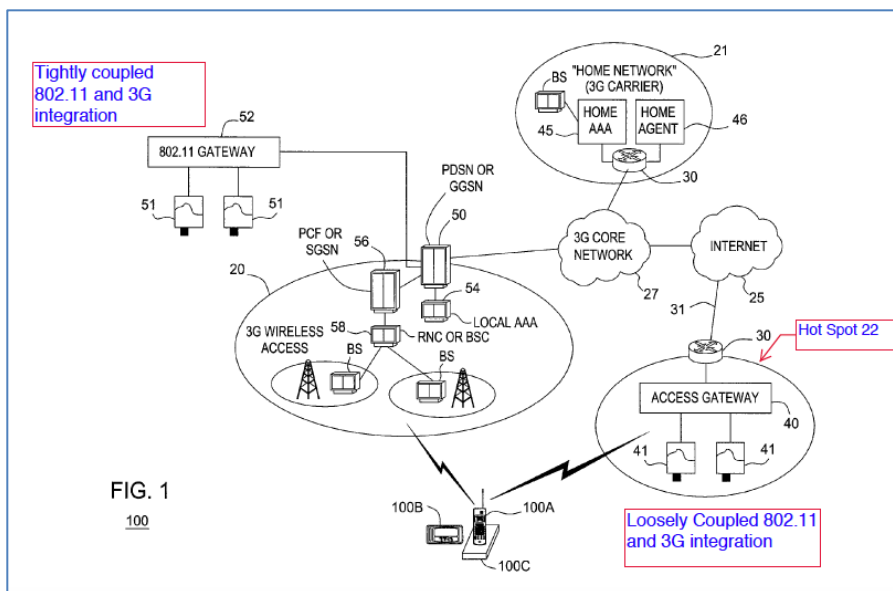
183. “The switching should happen automatically and transparently without the user's intervention.” *Id.* at 4:47-48.

184. “The 802.11 provider should be able to honor the service level, such as QoS (Quality of Service) guarantees, that the carrier has agreed to provide to

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

the user, while enforcing the policies that the user's contract with the 3G carrier foresees.” *Id.* at 4:48-52.

185. Buddhikot describes a “loosely-coupled Internetworking” which is based on an “802.11 gateway” which is added to the 802.11 network. *Id.* at 6:15-17. This is depicted in FIGs 1 and 2 (annotations added below).



186. In a loosely coupled architecture “the gateway 40 connects to the Internet 25 and preferably does not have a direct link to 3G network elements.” *Id.* at 6:18-20. “Each gateway system 40 preferably serves multiple 802.11 access points 41 in a hot-spot, and controls the traffic from these APs 41 before it can reach the back-haul link 31. Although FIG. 1 shows the access points 41 directly connected to the gateway 40, an access point can be indirectly connected to the

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

gateway by way of an Ethernet switch or hub, or other local area network (LAN) switch or hub.” *Id.* at 7:7-14.

187. “A mobile node 100a-100c that roams into a hot-spot 22 preferably obtains 802.11 access under the control of the gateway 40. After successful authentication and Mobile-IP registration, the gateway 40 allows the mobile node 100a-100c to access the network (Internet 25, and possibly, core network 27).” *Id.* at 7:20-25.

188. The functions of the gateway are depicted in FIG 2 (annotations added below):

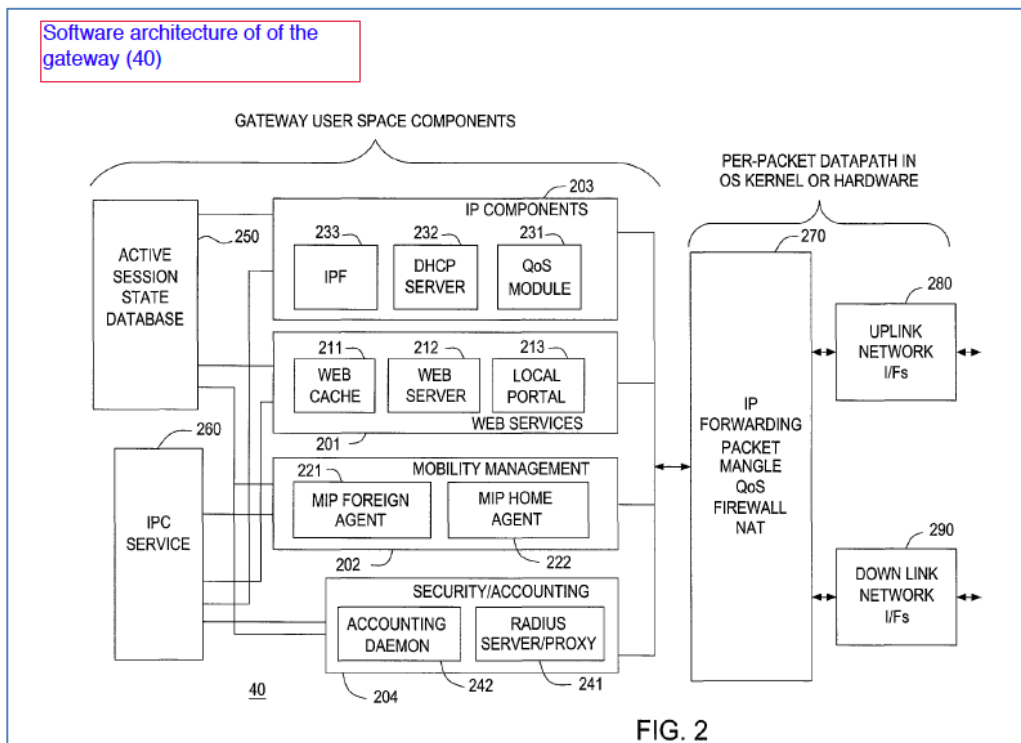


FIG. 2

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

189. “The gateway 40 also preferably provides QoS services and collects accounting data. The gateway 40 also preferably integrates a number of optional sub-systems, as shown in FIG. 2, including: web cache 211, web server 212, local portal 213, Mobile IP foreign agent 221, Mobile-IP home agent 222, QoS module 231, DHCP server 232, Internet Protocol filter 233, RADIUS server 241, accounting daemon 242, and dynamic firewall 270.” *Id.* at 7:25:32.

190. All the Gateway 40 subsystems preferably include a persistent, non-volatile (e.g., on-disk) database 250 to store information about each client's session. Thus, the state of the gateway 40 can be preserved and restored even in the event of a system reboot, making the gateway fault tolerant. The database 250 stores information that has already been processed, such as rules and address information. An IPC service 260 provides inter-process communications among all of the various modules 211, 212, 213, 221,222,231,232,233,241,242.” *Id.* at 7:32-41.

191. The gateway provides “Internet access only to legitimate users. Therefore, the WLAN gateway 40 authenticates the users.” *Id.* at 8:25-27.

192. Buddhikot also teaches tunneling to gateway that acts as a proxy with IPSec/VPN. “The gateway does not provide data-link encryption. For enhanced

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

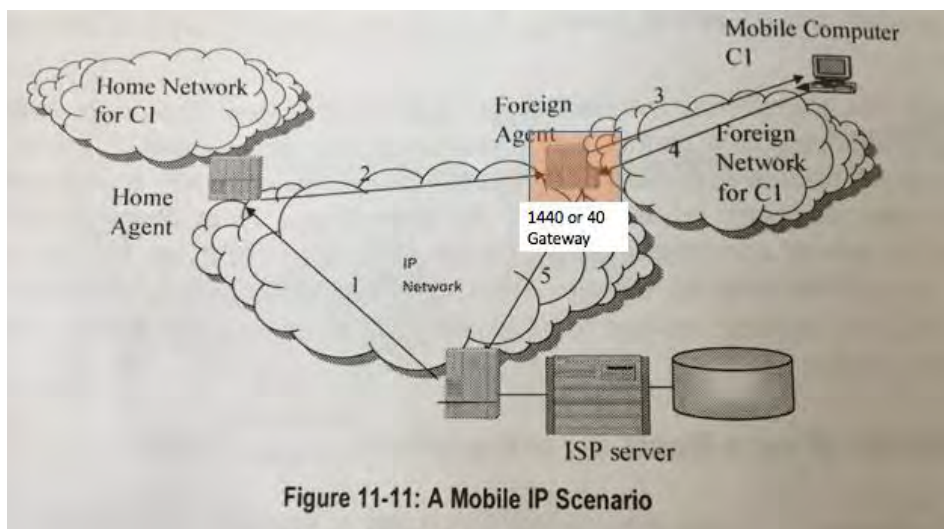
privacy external end-to-end privacy solutions such as IPSec/VPN may be used.” *Id.* at 9:48-58. Buddhikot also teaches encrypting traffic. *Id.*

193. Gateway 40 “preferably implements a very scalable and efficient Mobile-IP agent function 202 which supports the rolls of both Home agent 222 and Foreign agent 221 (HA and FA, respectively).” *Id.* at 10:33-36.

194. The MobileHotSpot Gateway 1440 is capable of running under two public IP address management schemes—Mobile IP and Simple IP, both of which give a client device a unique public IP address. *E.g.*, Buddhikot at 6:34-36, 10:33-43, 16:20-52. Under the Mobile IP framework, a mobile node uses its home address as its public IP address across networks. *Id.* at 10:44-55, 9:36-43, 14:54-56. Under the Simple IP framework, a mobile client obtains its public IP address via DHCP, and performs a browser log-in to access the Internet. *Id.* at 16:21-52

195. A mobile node which supports mobile IP has two IP addresses. “*Some embodiments also preferably support Mobile-IP tunnels.*” *Id.* at 13:37-38. The wording is somewhat redundant because Mobile-IP implies tunnels and tunnel endpoints have different IP addresses than the addresses of their source and destinations. The point is that Buddhikot supports mobile IP technology.

196. As mentioned above the gateway (40) contains a foreign agent (221). The figure below depicts a mobile IP configuration where the gateway (the transparent orange box) is used as such a foreign agent connecting ‘mobile node(s) A’ to the Internet. Each such node would have its fixed IP address on its home network, as well as a ‘**care-of-address**’ which it received from the foreign agent.



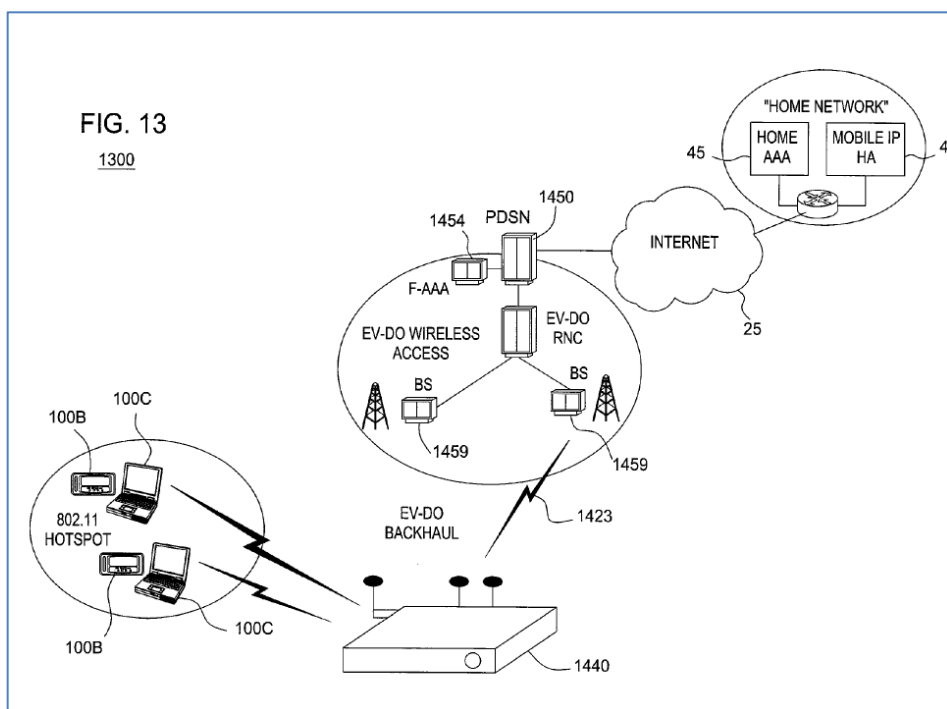
197. Depending on the type of gateway used the connection to the Internet is provided by an ISP. The connection is either wired or (for gateways of type 40) or 3G wireless (gateways of type 1440 – see below).

a) Another embodiment: the gateway also has a “wireless backhaul link”

198. There is another embodiment of the gateway described above. In this embodiment the gateway also has a “wireless backhaul link” (*id.* at 23:45) which

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

enables the gateway to connect to 3G. FIG. 13 shows an exemplary network implementation. The gateway 1440 is very similar to gateway 40 discussed above, with an additional wireless backhaul connection. In this topology mobile nodes connect to a local 802.11 network, which is connected by the gateway to a G3 access/core network through which the mobile nodes can access the internet.

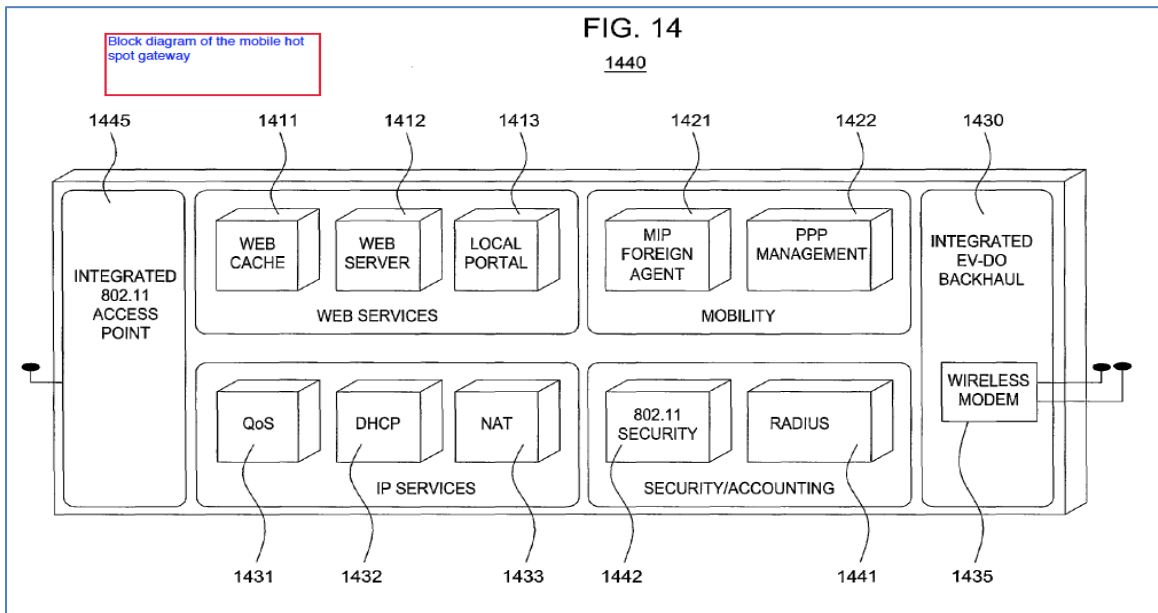


199. For example: “Subscribers can access the Internet in buses, trains, or hotspots using 802.11 in the same manner as they do at home and at work, to connect to the backhaul wireless data channel such as EV-DO, UMTS, 1xRTT, GPRS, or other such wireless packet data channel. The client may have both an 802.11 card and a 3G card. The client uses 802.11 to connect to the gateway 1440,

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

and the gateway 1440 connects to the rest of the Internet by a wide area wireless link.” *Id.* at 23:52-59.

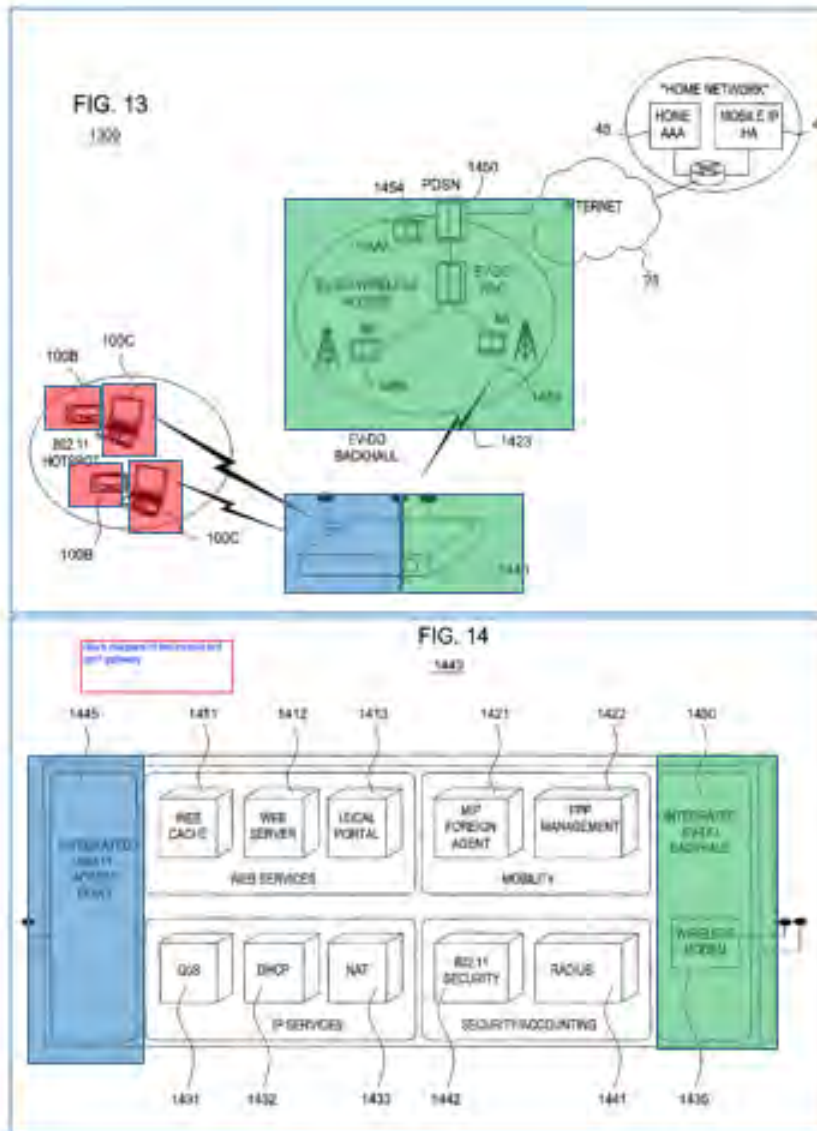
200. A block diagram of the mobile hot spot gateway is depicted in FIG. 14 (annotation added):



201. The NAT component 1433 enables the gateway to map between public and private IP addresses. The presence of a DHCP component 1432 implies that the gateway can allocate IP addresses to mobile nodes which use its LAN(s) as (a) home network(s).

202. Figure 14 shows the details of element 1440 which is depicted in the full network context in Figure 13.

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

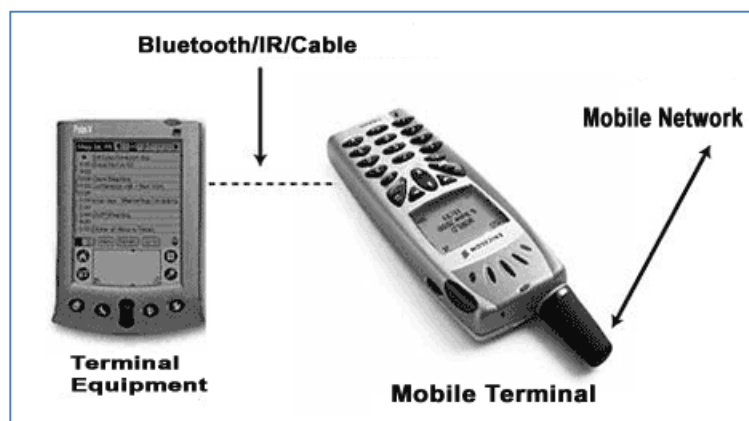


FIGs 13 and 14 (colors added). Red indicates stations, e.g. laptops or PDA's. Blue indicates a wireless access point. Green indicates components of a distribution systems which connect to the internet. FIG 14 (1440) depicts the details of FIG 13 element 1440.

203. Buddhikot is an innovative combination of common elements in a manner which solves the problem of connecting between local wireless networks and wide area wireless networks, in a manner which allows users to seamlessly access the Internet from devices which are not equipped with wide area wireless interfaces.

2. Summary of Lord US 6,763,012

204. Mobile IP discussed above is one example where devices can have multiple IP addresses which are assigned by a gateway. Lord describes another complementary scenario where IP addresses are assigned by a “mobile terminal” (MT) to “terminal equipment” (TE).

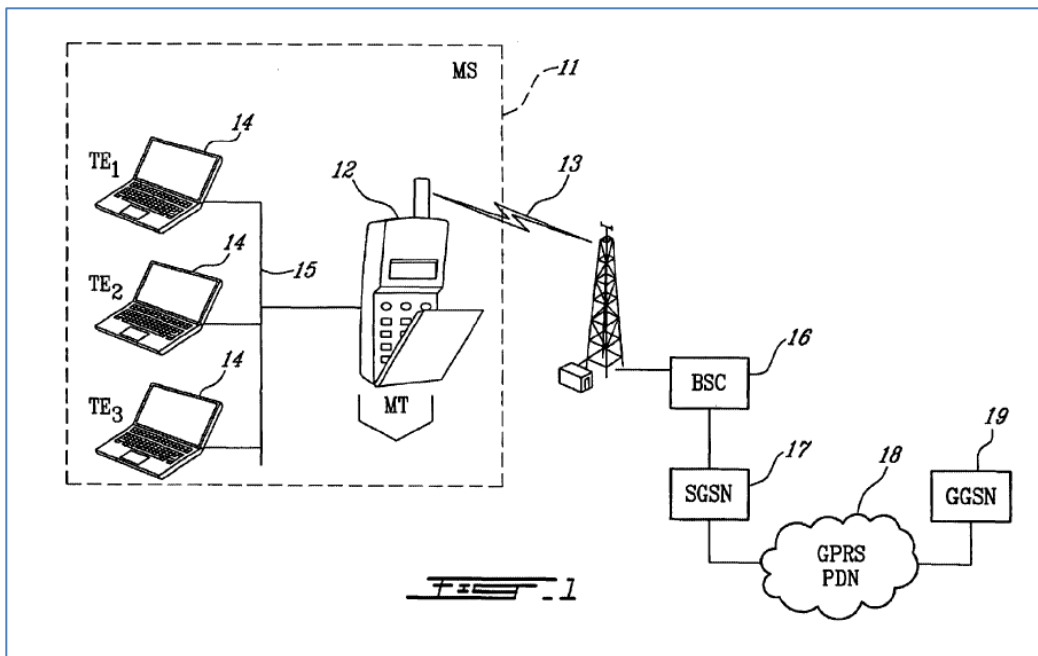


205. Lord describes the motivation for the invention as a need to provide “a user [who] may carry a plurality of devices (for example, a Personal Digital Assistant (PDA), a laptop computer, a mobile telephone, etc.) that are capable of communicating with each other over a wired or wireless network. These devices

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

may be connected to each other over a small Local Area Network (LAN) or a wireless LAN utilizing technologies such as Bluetooth, for example. It would be desirable to provide two-way connectivity to PDNs such as the Internet for all of the devices connected to the user's LAN.” Lord at 1:42-50.

206. The network architecture is depicted in FIG. 1:



207. Lord teaches an Internet connected mobile terminal 12 that is connected to a base station access point and also to laptops 14 via Wi-Fi. Specifically, “a Mobile Station (MS) 11 includes a Mobile Terminal (MT) 12 which handles the **GPRS air interface 13**, and a plurality of Terminal Equipment (TEs) 14 which are connected through a wired or **wireless LAN 15** to the MT. On the PDN side of the air interface is a Base Station Controller (ESC) 16, a Serving

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

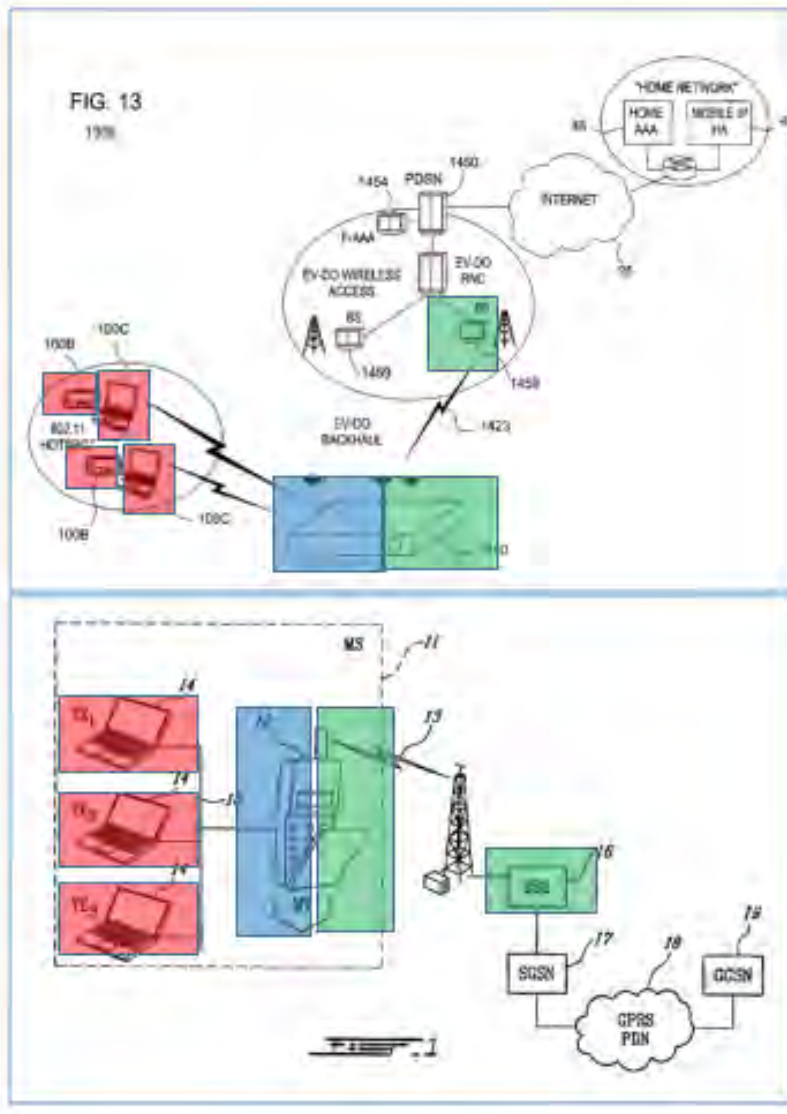
GPRS Support Node (SGSN) 17, the GPRS PDN 18, and a Gateway GPRS Support Node (GGSN) 19 which may be used to connect to other networks.” *Id.* at 3:34-42. **The mobile terminal 12 (e.g., a cell-phone) allows the laptops 14 to access the Internet. *Id.***

208. Lord describes a solution where “In one aspect, the present invention is a method of connecting a plurality of devices, each having a unique IP address, through a single MT and a **single wireless link to a PDN**. The plurality of devices are connected to the MT on a network such as a LAN.” *Id.* at 2:21-25 (emphasis added) . Lord “preferably” provides public IP addresses for its devices via DHCP. Lord at 4:41-51, Abstract, 5:5-7. Lord also virtually separates the mobile terminals on the LAN (2:32-38) to create separate logical connections. Lord at 3:56-60.

209. It is important to point out that the **address assignment and connection between the LAN and the PDN is done by the MT. The MT can be a mobile phone**, as shown in FIG. 1.

210. The figure below highlights the complementary relationship between Buddhikot and Lord. The figure of the top is Buddhikot Fig. 13. The figure on the bottom is Lord Fig. 1. Colors were added to highlight similar components.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306



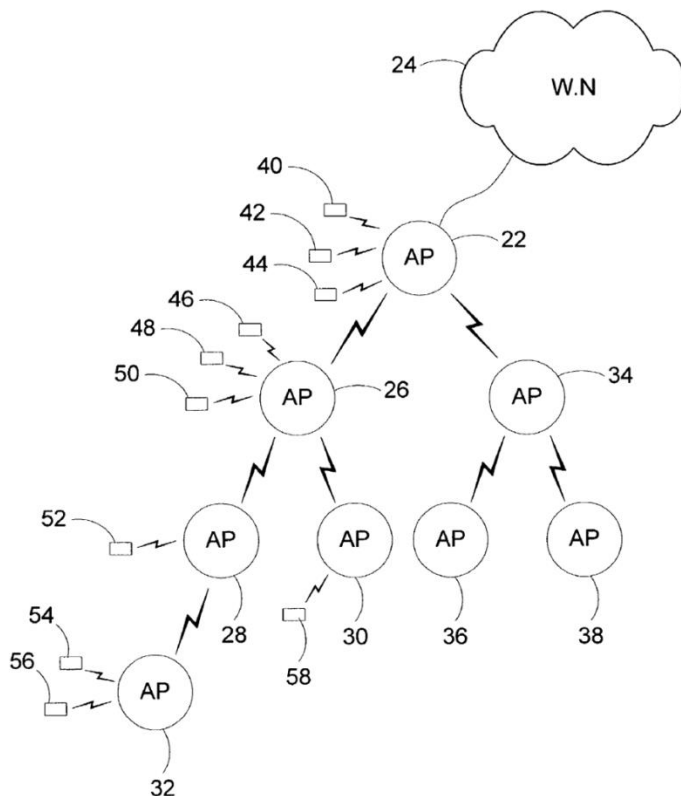
3. Summary of Kim U.S. Pub. No. 2004/0218611

211. Kim teaches functionality of a well-known extension of the NAT—the Network Address Port Translation (NAPT). Kim at ¶¶ 10-14, Fig. 1; EX1021, p. 11, §4.1.2.

4. Summary of Abhishek U.S. Pub. No. 2004/0103278

212. Abhishek teaches a network of 802.11 access points. Abhishek Fig. 1,
¶ 27. Any one of computers and access points shown below can route packets
through the multiple 802.11 access points to the Internet, accessible via the wired
network 24. See also Abhishek at ¶¶ 27-30.

FIG. 1



5. Summary of Moran PCT App. No. WO2005/069577

213. Moran discloses a system that uses Mobile IP with an IPSec/VPN
tunnel, where its home agent acts as the tunnel end-point and VPN gateway.
Moran at Abstract, 3:23-25. The home agent provides both VPN, IPSec, and

Mobile IP related services, including de-capsulation of Mobile IP and IPSec tunnels (e.g., 13:28-30) and VPN and Mobile IP related services (e.g., 14:29-15:10). The home agent acts as the gateway router for the home network that the mobile node can connect to. Moran at 6:17-31; *see also id.* at 7:1-34.

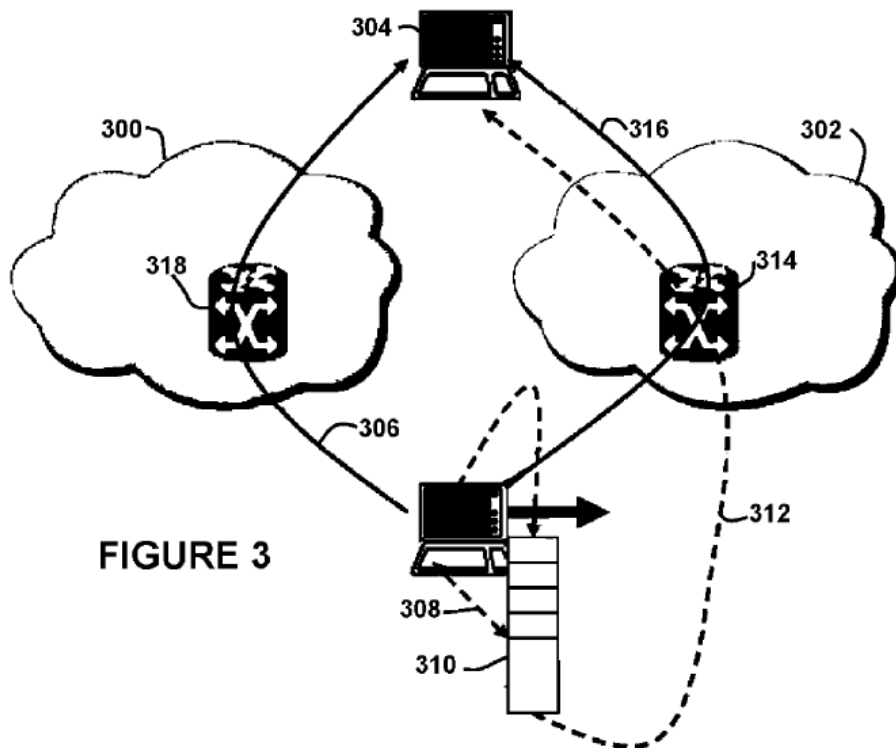
6. Summary of Fajardo U.S. Pub. No. 2007/0014259

214. Fajardo is an application from Toshiba that teaches a buffering node that facilitates hand-off when a mobile node roams to a new location. Fajardo at ¶31, ¶49, Abstract. The buffering node intercepts packets that are destined to an Internet location, temporarily buffers them, and forwards them to an Internet location after handoff. *Id.* at ¶82. Prior to moving to a new network, a mobile node signals the buffering node of its intent to roam, and then the buffering node starts to buffer traffic. *Id.* at ¶80.

215. Fajardo teaches buffering both incoming and outgoing traffic. *Id.* at ¶50, ¶100. Fajardo teaches that buffering out-bound traffic can be beneficial to the mobile node's software applications (*id.* at ¶39), can provide a recovery mechanism if handoff fails (*id.* at ¶102), and to save power (¶51).

216. Fajardo teaches an out-bound buffering situation in Figure 3, below, where prior to hand-off a mobile node is connected to a first access router 318. *Id.* at ¶99, Fig. 3. The mobile node has a buffer node 310 that "serves as a buffering

service to outgoing packets for the [mobile node] during the handoff period.” *Id.* at ¶101. After handoff, the buffer node 310 forwards packets to the new care-of address of the mobile node (*Id.* at ¶95), which a POSITA would understand to be the access routers’ 314 IP address, or an original addressed destination (*id.* at ¶102; *see also id.* at ¶¶93-96), depending on whether the buffer node is buffering ingress or egress traffic. As shown below, flow link 312 illustrates the buffered traffic between the correspondent node 304 over the Internet and the mobile node (*id.* at ¶99) and a POSITA would understand this traffic to be sent to the correspondent node 304 after hand-off.



217. Figure 3 shows the buffer node at the mobile node, but Fajardo makes clear that in the particular embodiment of Figure 3 that “[t]he BN [buffer node] may also be located within the previous access point to assist in buffering packets when the MN is actively scanning for finding a new access point or testing connectivity to a new access point.” Fajardo at ¶103; *see also id.* at ¶97 (stating that the buffer node can be at either access router or at the mobile node), ¶51 (stating that the buffer node may be located in the access point to assist the mobile node).

7. Summary of Vucina U.S. Pub. No. 2005/0261970

218. Vucina teaches an access point that provides multiple virtual access points (Vucina at Figs. 5-6, ¶¶167-190) that provide Internet access via different carriers (*id.* at Abstract). Vucina also teaches providing tunneling, among other services, for client devices connected to the virtual access points. *Id.* at ¶¶81, 105-106, 177.

8. Summary of Aarnio U.S. Patent No. 7,606,559

219. Aarnio is a Nokia patent that teaches a system for forwarding content, where a Wi-Fi capable (Aarnio at 7:56-62) and 3G capable (*id.* at 8:53) mobile terminal device can request to transfer media content from a source server to a destination server (*id.* at Abstract, 11:53-65) on the Internet (*id.* at 17:50-52).

After the source server completes the upload, it wirelessly sends an upload confirmation message to the mobile node. *Id.* at Fig. 8A, 17:60-63.

VI. PATENT OWNER DID NOT INVENT THE HOT-SPOT

220. As an initial matter, Patent Owner *did not* invent the Wi-Fi/3G hotspot.

221. It is clear that Patent Owner believes that they invented the Wi-Fi hotspot. They have made multiple representations of claim scope that clearly indicate their belief. *See* EX1024, EX1025, EX1036, EX1037. This is most clearly shown by a technology tutorial they submitted in the District Court litigation. The below is extracted from this tutorial, where Patent Owner provided an illustration of the tethering concepts it believed it invented. EX1037 at 3.

Tethering Illustration

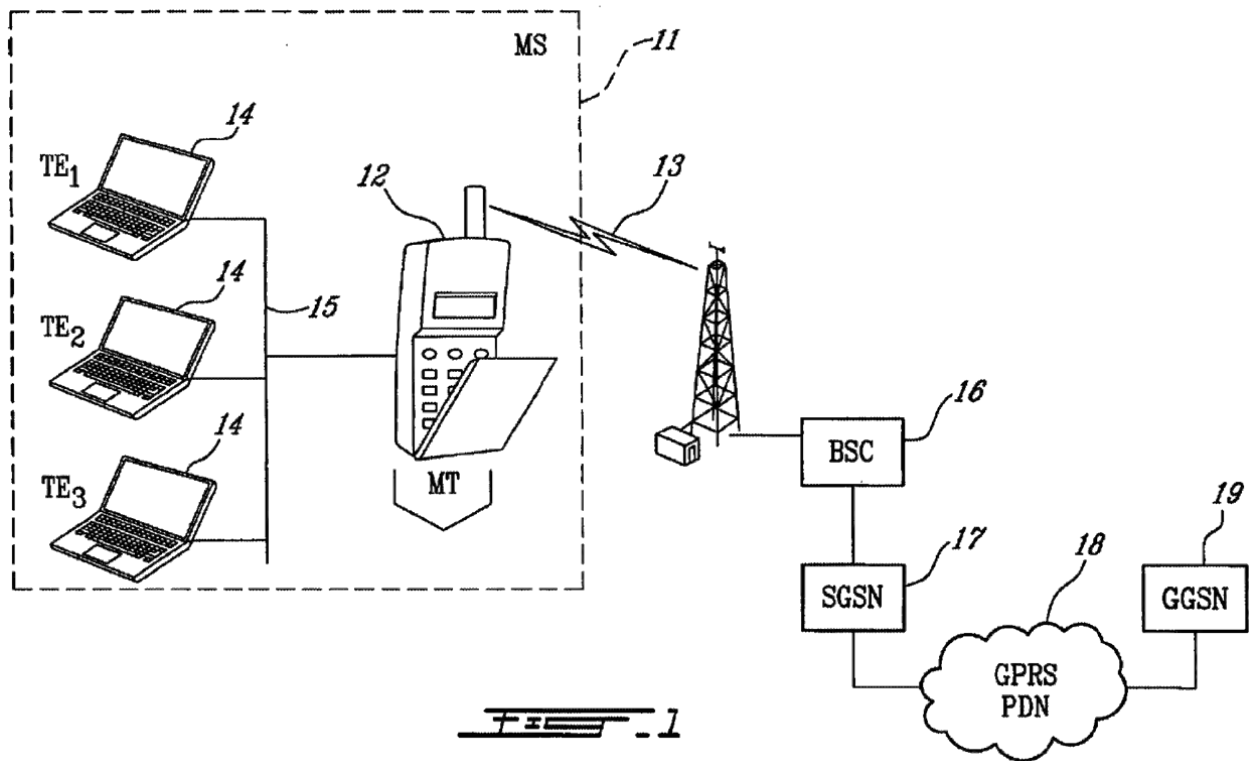
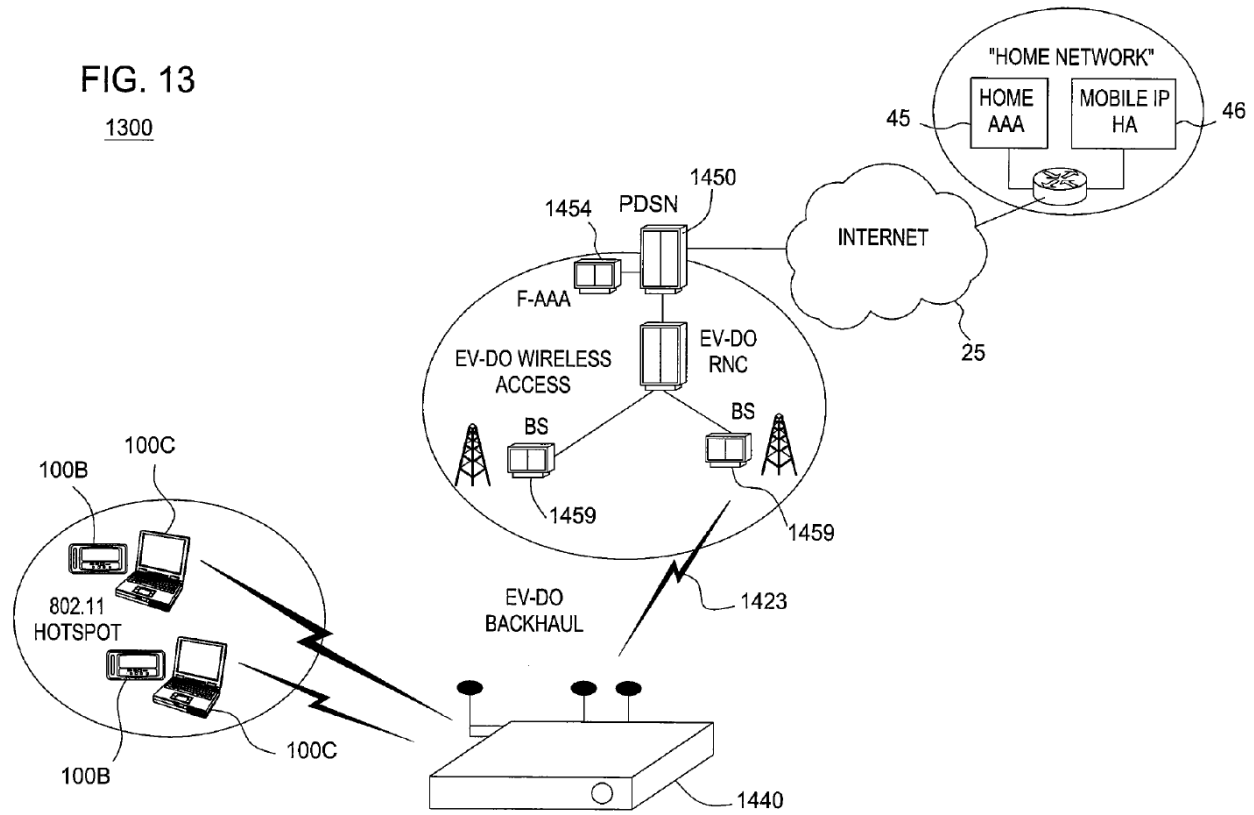
- Tethering enables, for example, a smartphone connected to the Internet via a 3G cellular data connection to act as a mobile WiFi hotspot for other wireless devices, such as a tablet computer, laptop computer, or personal data assistant (PDA).



222. This technology was clearly well-known. This illustration mirrors Buddhikot and Lord, which both illustration Wi-Fi hotspots allowing devices to connect to cellular networks. Buddhikot Fig. 13 and Lord Fig. 1 are shown below.

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

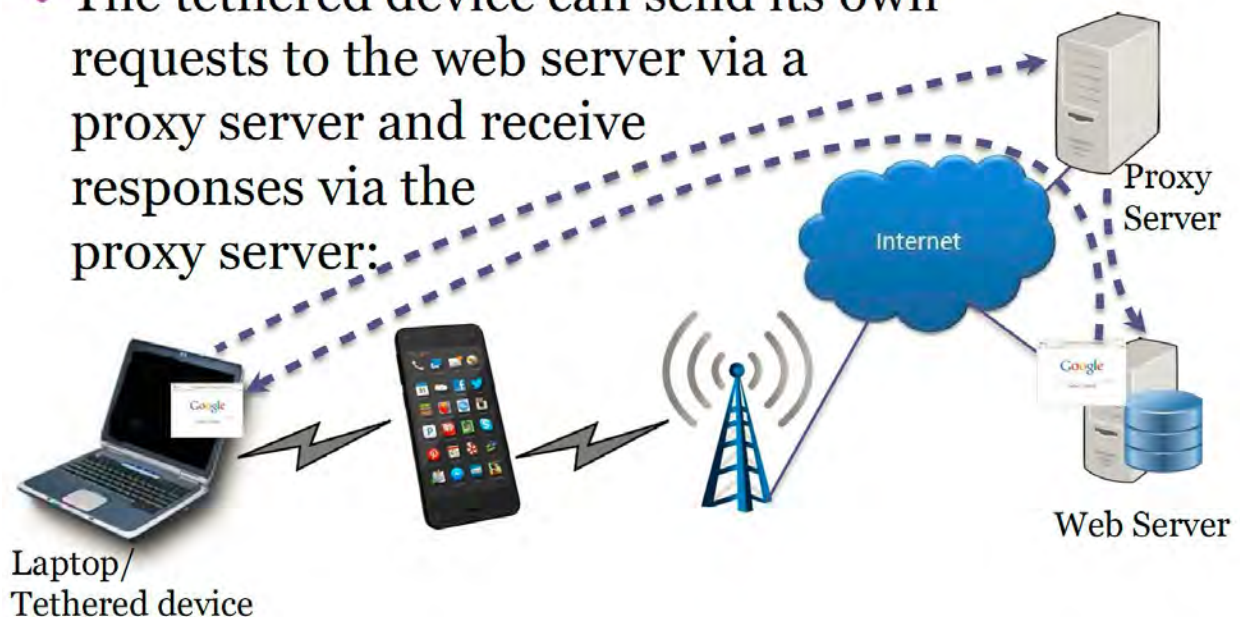
FIG. 13



223. The other part of the '306 and '369 patents claim tunneling data to an end-point gateway. They've described the claimed proxy server as "a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers." EX1024 at 1. As shown in Patent Owner's claim construction tutorial, below, this is simply browsing Google.com with some intermediary server. EX1037 at 16. This is simple routing of traffic.

Tethering - Retrieving Web Pages

- The tethered device can send its own requests to the web server via a proxy server and receive responses via the proxy server:



224. In my opinion, the claims of the '306 and '369 patents are invalid.

VII. UNPATENTABILITY OF the '306 patent

A. Buddhikot And Lord Render Claims 1-12, 14-22, 26-31, 34-37, 39-40, 43, 47-53, 56-63, 65-66 Obvious

225. **Rational and Motivation to Combine:** In my opinion, a POSITA would have been motivated to combine Buddhikot and Lord, with an expectation of success, because they provide compatible disclosures. Lord and Buddhikot teach similar methods for assigning IP addresses: client devices can be assigned public IP addresses using DHCP, or instead use private IP addresses where a NAT is used to route traffic to the private IP address. Lord at 4:67-5:13; Buddhikot at 16:34-52. In addition, a POSITA would have been motivated to combine by Buddhikot's teaching of DHCP to allocate public IPs for a hot-spot with Lord's detailed discussion of DHCP. Buddhikot teaches using DHCP (Buddhikot at 16:17-43, 24:38-55), in brief, and Lord provides further detail and implication specifics regarding how DHCP can be used to allocate public IP addresses (Lord at 3:43-4:13, 5-5-13).

226. In my opinion, a POSITA would have been motivated to combine Buddhikot and Lord because Lord focuses on the connectivity aspect of the hot-spot and IP address allocation and Buddhikot provides additional focus on security and authentication. As mentioned above, Lord teaches DHCP and address allocation particulars, and Buddhikot teaches security related features

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

(e.g., IPsec/VPN and SSL (9:43-56, Fig. 15), authentication (10:3-31), firewalls (10:55-67), various types of proxies (Figs. 2 and 14)) along with utilizing the Mobile IP standard. A POSITA looking to set up a mobile hotspot architecture would have been concerned with how to allocate public IP addresses, as well as how to provide end-to-end security for the connecting devices, motivating them to combine disclosures from Lord and Buddhikot.

227. In my opinion, it would have been obvious to a person of ordinary skill in the art to combine Buddhikot with his knowledge of the existing standards and technologies, with no change to their respective functions. A POSITA would know that the existing 802.11 technologies enabled a computing device with a wireless card, whether it is a computer (laptop, desktop, server) or a wireless access point to function as a station as well as an access point if it had the appropriate software which supported wireless standards and enabled packet forwarding accordingly. Buddhikot provides a specific implementation of such general functionality. The ability to connect to an “IP based network” via the “first wireless access point” is an inherent capability 802.11 or any wireless network or any layer 2 network whose packets can contain layer 3 packets. Buddhikot provides a specific implementation of this inherent capability. Wireless access points support many wireless protocols including 802.11 and 3G protocols.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

Gateway devices convert between packet structures of different protocols. A POSITA would have understood that Buddhikot makes specific use of such a conversion which is done between an 802.11 and 3G access points. A POSITA would know that the Internet is a very large IP based network that is used to connect computers using IP addresses. A POSITA would know that any form of connectivity is made of wired or wireless segments where wireless segments are connected by access points communicating with each other, or wireless access points which have a wired connection.

228. In my opinion, a POSITA would have been motivated to combine Buddhikot and Lord because they are focused on the same issue: how to integrate WLANs and cellular networks to achieve maximum connectivity. Buddhikot at 1:40-62; Lord at 1:62-2:18; They provide the same general network architecture as a solution, where client devices connect via Wi-Fi to a gateway device (a hot-spot), which itself connects to the Internet via a cellular network. *See* Buddhikot at Fig. 13 and Lord at Fig. 1. Because Buddhikot and Lord present a similar solution to a similar problem with complementary focuses on different issues, POSITA would have been motivated to combine them.

229. In my opinion, a POSITA would have been motivated to combine Lord and Buddhikot because they represent research by leaders in telecom. Lord

was assigned to Ericsson and Buddhikot was assigned to Lucent. A POSITA looking at hot-spots and networking, would have been motivated to look at research conducted by Ericsson and Lucent. It was normal practice for industry leaders to look at each-other's technology, to receive good implementation ideas, and a POSITA would have been motivated to do the same.

1. Element-by-Element Analysis of the Challenged Claims

230. **Claim 1, Preamble:** The Claim 1 preamble recites: “**A computing device comprising:**”

231. In my opinion, a computing device would have been obvious to a POSITA for wireless access point, gateway, firewall, proxy or packet filtering device. It would have been obvious in light of Buddhikot itself, that the gateway is a computing device, as it comprises of all these functions. Importantly, the gateways taught by Buddhikot run on Linux, which a POSITA would know is an operating system which runs on a computing device. In addition, such elements were implemented in standard software distributions of Linux packages. Buddhikot discloses a “gateway operation with wireless backhaul” mode of operation. “FIGS. 12-14 show another exemplary embodiment in which the gateway 1440 has a wireless backhaullink 1423 and is capable of functioning in a mobile environment. The MobileHotSpot Gateway 1440 combines an 802.11 AP 1445, a Wireless

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

modem 1435 for Backhaul, and a Public Access Gateway. The backhaul link 1423 is established via a 3G wireless data channel such as CDMA 1 x Evolution Data Only (EV-DO), UMTS, 1xRTT, GPRS, or CDMA 1x Evolution Data and Voice (EV-DV). Subscribers can access the Internet in buses, trains, or hotspots using 802.11 in the same manner as they do at home and at work, to connect to the backhaul wireless data channel such as EV-DO, UMTS, 1 xRTT, GPRS, 55 or other such wireless packet data channel. The client may have both an 802.11 card and a 3G card. The client uses 802.11 to connect to the gateway 1440, and the gateway 1440 connects to the rest of the Internet by a wide area wireless link (because the user does not have a wired link such as Ethernet or Sonet link available).” Id. 23:43-61.

232. Buddhikot teaches a MobileHotSpot Gateway 1440 (the claimed computing device) that “permits wireless communication between the gateway and a wireless network.” Buddhikot at 23:44-48; *see also id.* at 3:6-29, Figs. 1-3, 12, and 14 (illustrating other embodiments of the claimed computing device).

233. In my opinion, it would have been obvious to a person of ordinary skill in the art to combine Buddhikot with his knowledge of the existing standards and technologies, with no change to their respective functions. A POSITA would know that the existing 802.11 technologies enabled a computing device with a

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

wireless card, whether it is a computer (laptop, desktop, server) or a wireless access point to function as a station (STA) as well as an access point (AP) if it had the appropriate software which supported wireless standards and enabled packet forwarding accordingly. Buddhikot provides a specific implementation of such general functionality.

234. The ability to connect to an “IP based network” via the “first wireless access point” is an inherent capability 802.11 or any wireless network or any layer 2 network whose packets can contain layer 3 packets. Buddhikot provides a specific implementation of this inherent capability.

235. Wireless access points support many wireless protocols including 802.11 and 3G protocols. Gateway devices convert between packet structures of different protocols. A POSITA would have understood that Buddhikot makes specific use of such a conversion which is done between an 802.11 and 3G access points.

236. A POSITA would know that the Internet is a very large IP based network that is used to connect computers using IP addresses. A POSITA would know that any form of connectivity is made of wired or wireless segments where wireless segments are connected by access points communicating with each other, or wireless access points which have a wired connection.

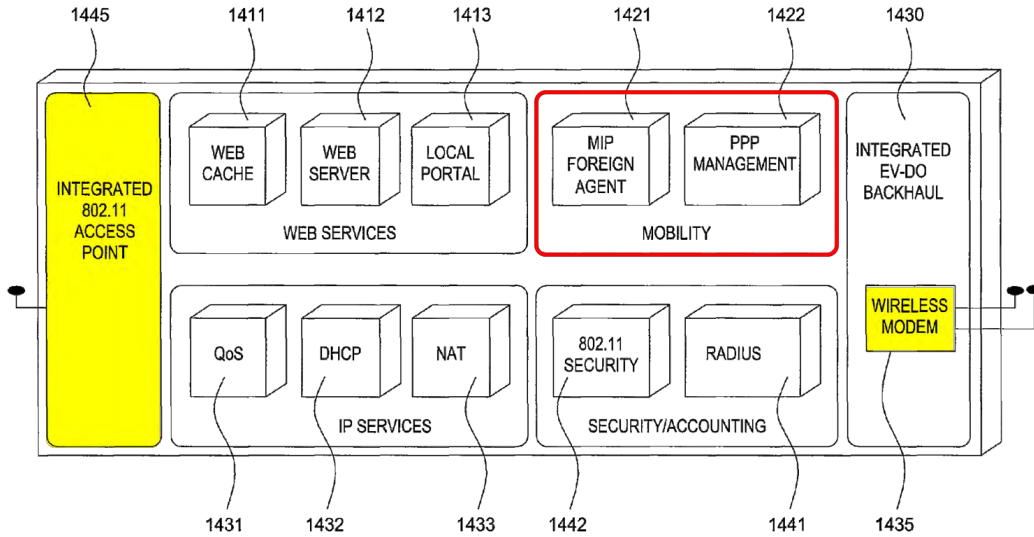
237. A person of ordinary skill in the art would have understood that there were many prior art ways to satisfy the preamble of Claim 1, one of which is disclosed by Buddhikot.

a) “at least one communication module adapted to:”

238. **Element [1.A]:** Claim 1, Element [1.A] recites: “**at least one communication module adapted to:**”

239. Buddhikot teaches a MobileHotSpot Gateway 1440, which contains *inter alia* mobility software as the claimed communication module. Buddhikot at 24:19-26; *see also id.* at Fig. 2 Element 202, 221, 222 (annotated in red below), Fig. 5. The mobility software manages clients that move across different wireless technologies (Buddhikot at 10:33-54, 18:31-43), interacts with both the wireless 802.11 access point 1445 and the wireless modem 1435 to send traffic to from the clients to the Internet. Buddhikot at 24:26-35, 23:46-49, Fig. 14 (yellow, below). A POSITA would understand that various other drivers, communication stacks, and protocols in the MobileHotSpot Gateway 1440 provide communication services to 802.11 and wireless modem interfaces.

FIG. 14
 1440



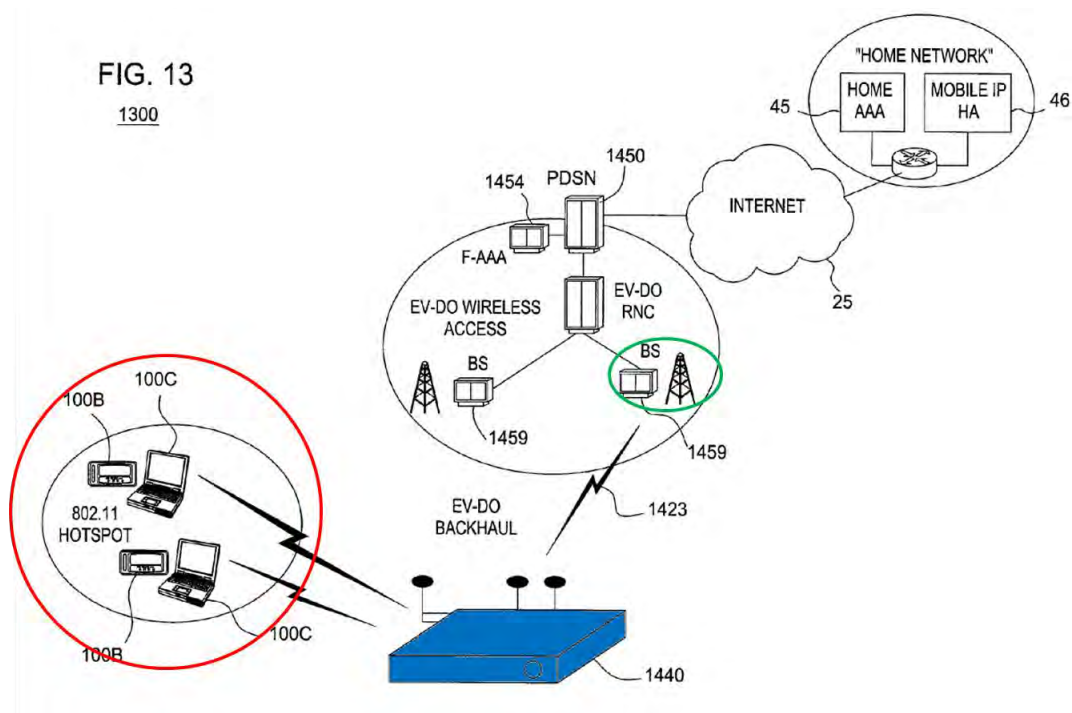
(1) “(1) wirelessly connect said computing device to an IP based network via a first wireless access point (AP) having a first AP Identification (APID); and”

240. **Element [1.A.1]:** Claim 1, Element [1.A.1] recites: “(1) wirelessly connect said computing device to an IP based network via a first wireless access point (AP) having a first AP Identification (APID); and”

241. A POSITA would have known that a wireless access point must have an access point identification (APID). A POSITA would know that APIDs are defined and required by the 802.11 standard. In fact, 802.11 cannot function without APIDs, therefore the claim element is simply restating parts of the standard. Buddhikot teaches that the MobileHotSpot Gateway 1440 is wirelessly connected through a wireless backhaul link to the Internet (an IP based network)

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

through a base station 1459 (first wireless access point (AP)). Buddhikot at 23:44-24:18, Figs. 12-13 (base station annotated in green below). A POSITA would understand that the base station 1459 (the claimed access point) has a cell ID (the claimed access point ID) because it would have been obvious to a POSITA that base stations have unique cell IDs. *See also* EX1030 at 5:25-55; EX1028 at 212.



242. In my opinion, Petitioner's application of [1.A, 1.A.1, and 1.A.2 (illustrated in more detail below)] is consistent with Patent Owner's belief of the BRI of [1.A, 1.A.1, and 1.A.2]. Patent Owner's Infringement Contentions against Petitioner state that the first access point and APID correspond to a cellular base

Declaration of Tal Lavian, Ph.D. in Support of
 Petitions for *Inter Partes* Review of
 U.S. Patent Nos. 8,559,369 and 9,042,306

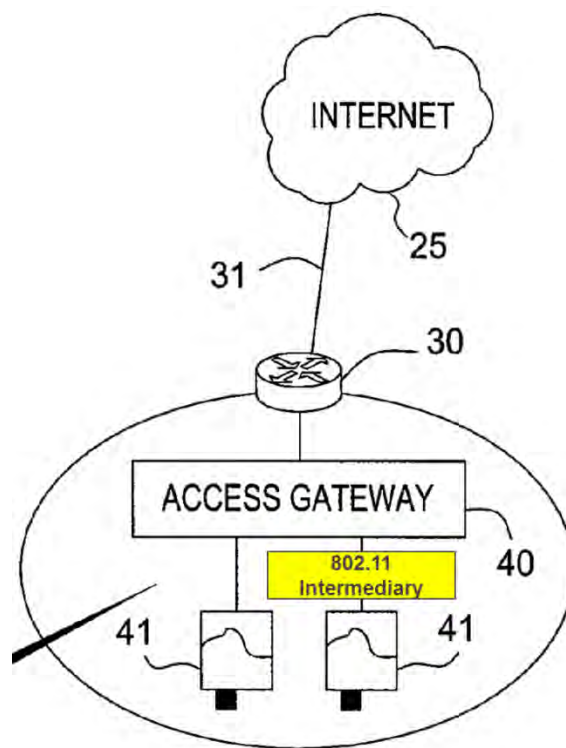
station and cell ID, respectively. EX1025 at 1 (annotated excerpt below); EX1037 at 13.

<p>at least one communication module adapted to: (1) wirelessly connect said computing device to an IP based network via a first wireless access point (AP) having a first AP Identification (APID); and (2) wirelessly communicate with other wireless enabled computing devices;</p>	<p><i>Accused component: Microsoft Lumia 435</i> <i>Basis of Infringement Contention: The Microsoft Lumia 435 provides communication module adapted to wirelessly connect the device to an IP-based network via a first wireless access point (AP) having a first AP Identification (APID).</i></p> <p><i>Cellular smartphones are able to communicate using radio frequency signals over a cellular network and over a wifi network. Conducting such communications requires communications circuitry that comprises one or more communication modules.</i></p> <p><i>A cellular base station acts as a first wireless AP and connects to an IP-based network through, e.g., a base transceiver station in GSM networks or node B in UMTS networks. Base stations, node B entities, and other similar devices have a unique identifier (e.g., cell ID) that enables mobile smartphones and the core cellular network to identify them and distinguish between different stations. Communications with a base station or node B are generally conducted using various cellular multiple access technologies.</i></p> <p><i>A cellular smartphone that is capable of supporting wireless tethering (acting as a mobile hotspot) wirelessly communicates with the tethered devices (e.g., laptop or tablet computer). Such communications are generally carried out using either a wifi or Bluetooth connection.</i></p> <p><i>See User Manual, pp. 90-91:</i></p>
--	---

243. In my opinion, it would have been obvious for a POSITA to have the MobileHotSpot Gateway connect to a 802.11 access point instead of a base station access point. This combination would have been obvious in light of Buddhikot alone, or further in view of Buddhikot and Lord. First, Buddhikot teaches providing LAN intermediaries between its gateways and its 802.11 access points. Buddhikot at 7:11-19. 802.11 is a wireless LAN, which the Buddhikot disclosure is replete with, and a POSITA would have found it obvious to provide a WLAN (802.11) intermediary between a 802.11 access point and a gateway. As I have previously described in ¶¶95-102, 802.11 intermediaries were well-known in the art to a POSITA. *See also* Abhishek at Abstract; EX1031 at Fig. 1., ¶¶7, 35. As a demonstrative, shown below is an annotated excerpt of Fig. 1, which shows

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

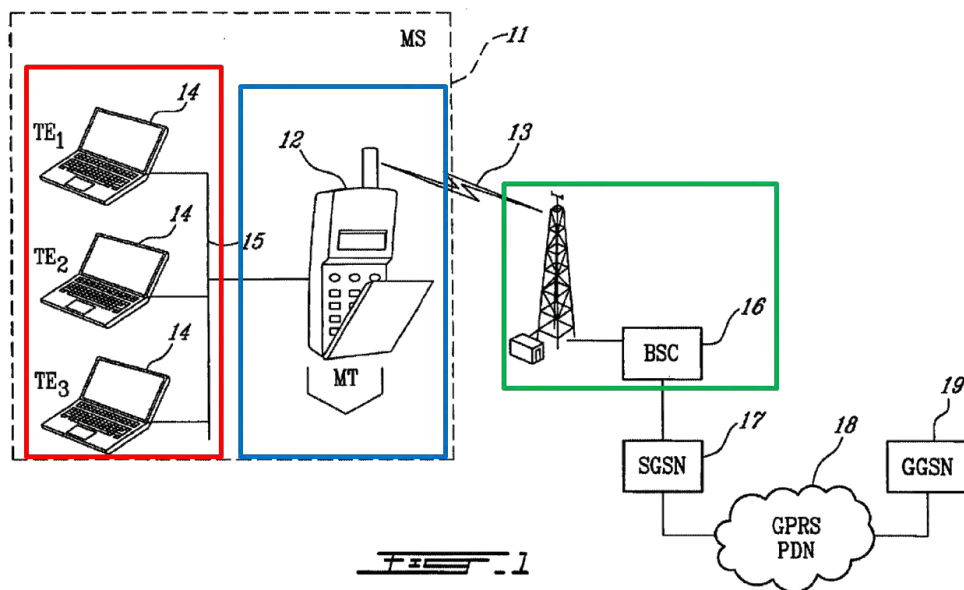
Buddhikot's teaching of a LAN intermediary (which would have been obvious for a POSITA to be 802.11) between access gateway 40 and access point 41. Accordingly, it would have been obvious for a POSITA to have the MobileHotSpot Gateway 1440 connect to an 802.11 intermediary access point, before connecting to the backhaul network to access the Internet.



244. Second, in my opinion, it would have been obvious to combine Lord's teaching of using any wireless network with Buddhikot's teachings. Lord teaches an Internet connected mobile terminal 12 (annotated in blue, below) that is connected to a base station access point (green) and also to laptops 14 (red) via a wireless LAN using the mobile terminal's 802.11 access point. Lord at 3:29-42,

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

Fig. 1. Lord states that the architecture of Fig. 1, although displaying a GPRS network (*i.e.*, using a base station access point), can be used in any type of wireless PDN (packet data network). Lord at 3:31-33, 3:19. A POSITA understands that a PDN refers to “[a] network that uses packet switching techniques” (EX1029 at 787, IEEE Dictionary), which would encompass 802.11. *See also* Lord at 3:47. Accordingly, in my opinion, a POSITA would have combined Lord’s teachings of having its mobile terminal act as a Wi-Fi hot-spot and connect to an 802.11 access point with Buddhikot’s teachings.



245. In my opinion, a POSITA would have been further motivated to combine Lord and Buddhikot because connecting to an 802.11 access point, instead of a base station access point, is a known technique that would yield predictable results. A POSITA would understand that this was commonplace in

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

the art—802.11s for wireless meshes was in development and proposed by the priority date of the '306 patent. EX1032. Furthermore, as described above in ¶¶78-102, providing 802.11 bridges, repeaters, and ad-hoc networks was well-known in the art. Abhishek at Abstract. A POSITA would understand that base station and 802.11 access points are known methods to wirelessly communicate data, and are can be easily substitutable.

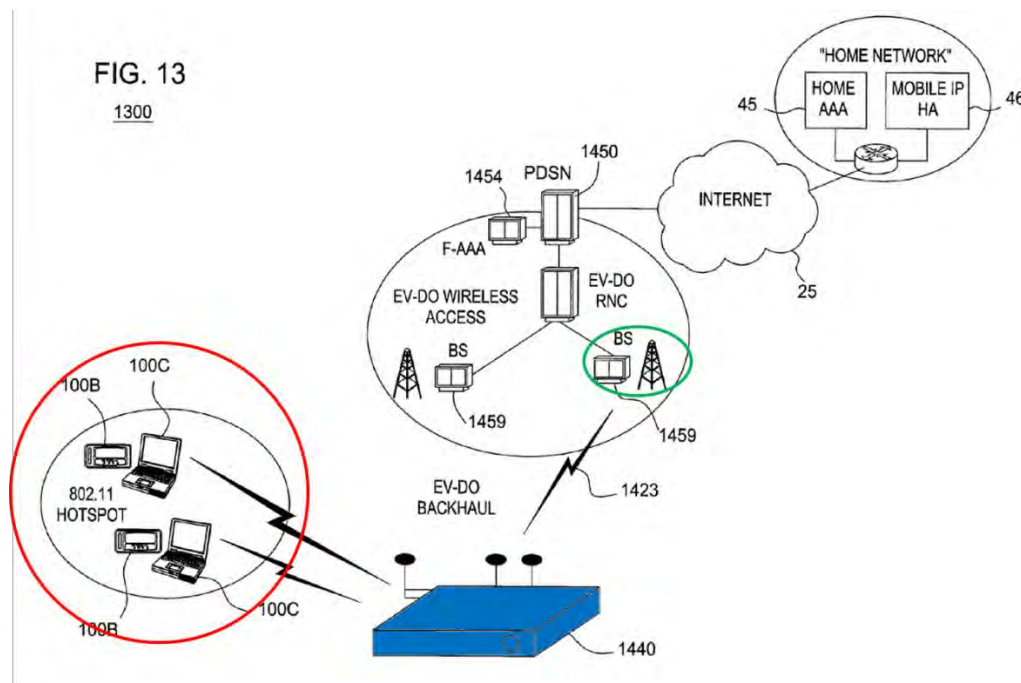
246. A POSITA would have found it obvious that an 802.11 access point would have an SSID. This is described in both EX1012 at ¶¶180-184 and EX1033 at 4:13-16.

(2) “(2) wirelessly communicate with other wireless enabled computing devices;”

247. **Element [1.A.2]:** Claim 1, Element [1.A.2] recites: “**(2) wirelessly communicate with other wireless enabled computing devices;**”

248. A POSITA would know that wireless access points can communicate with computing devices and with each other based on the 802.11 protocol standards. This is further emphasized in the standard clarification mentioned above. Buddhikot discloses that the MobileHotSpot Gateway 1440 (the claimed computing device) wirelessly communicates with other wireless enabled computing devices. Buddhikot at Fig. 14. The mobility software (the claimed

communication module) utilizes the 802.11 interface (*see, e.g.*, Fig. 14, Element 1446) of the MobileHotSpot Gateway 1440 to communicate with and authenticate the mobile nodes 100a-100c (circled in red, below). *Buddhikot* at 7:20-30, 13:48-59, 18:31-41, 9:44-52, 16:34-52.



b) “a user interface and display adapted to allow a user of said computing device to interact with destinations over the IP based network, through the first wireless AP, using a first public IP address associated with the computing device; and”

249. **Element [1.B]:** Claim 1, Element [1.B] recites: “a user interface and display adapted to allow a user of said computing device to interact with destinations over the IP based network, through the first wireless AP, using a first public IP address associated with the computing device; and”

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

250. Buddhikot teaches that the MobileHotSpot Gateway 1440 (the claimed computing device) interacts with destinations over the Internet (the claimed IP based network), through the 3G base station 1459 (the claimed first wireless AP), using a public IP address associated with the MobileHotSpot Gateway 1440. The MobileHotSpot Gateway 1440 is connected to the Internet via base station 1459, using its public IP address (the first public IP address). Buddhikot teaches that the MobileHotSpot Gateway 1440 can have a statically or dynamically configured public IP address (Buddhikot at 24:64-25:1-3), that can be obtained from a DHCP (*id.* at 25:29-34). In addition, Buddhikot teaches using the Mobile IP standard (*id.* at 6:34-36), where the MobileHotSpot Gateway 1440 can have a Mobile IP foreign agent (Fig. 14, Element 1421) having a “care of” address (25:2-3), which a POSITA would understand to be a public IP address. Finally, the MobileHotSpot Gateway is an Internet connected gateway, and a POSITA would understand that it would use its public IP address to communicate with Internet servers.¹¹ *See also* Buddhikot at 22:43-49 (describing FTP), 24:52-55 (describing a NAT which interfaces public/private IPs), 6:30-45 (stating that the

¹¹ In addition, Buddhikot’s gateways run on Linux (7:43-52, 10:58, 16:52), a robust Internet-capable operating system that provides many Internet applications.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

gateway interacts with the Home AAA services), 10:33-43 (stating that the foreign agent communicates with the home agent).

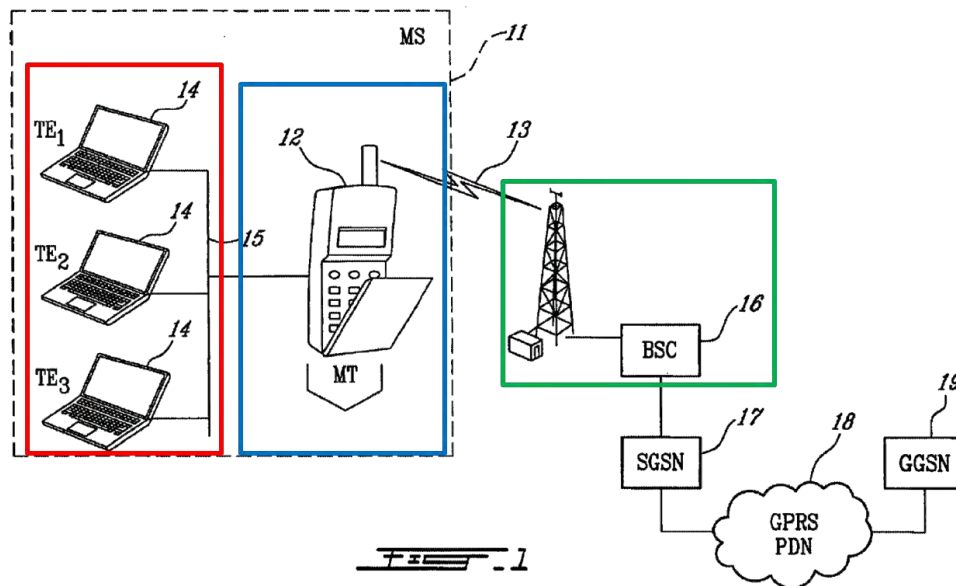
251. In my opinion, providing a user interface and display for the MobileHotSpot Gateway 1440 would have been obvious in light of Buddhikot itself. Importantly, the gateways taught by Buddhikot run on Linux. Buddhikot at 7:43-52, 10:58, 16:52. A POSITA would understand that Linux was robust and provided GUIs, Internet connectivity stacks and applications (*e.g.*, SSH, VPN, and Internet browsers).

252. In my opinion, a POSITA would understand that providing a display and a user interface with Linux would provide predictable results. This would be as simple as plugging a display into a display port to access the Linux GUI. Further, a POSITA would understand that the gateway's Linux software can be accessed via a remote display using remote access tools (*e.g.*, SSH) via the Internet. A POSITA would have been motivated to provide a display and user interface for the gateways—an employee installing, configuring, updating, and testing the gateways would necessarily need to access the gateways via an input like a display and user interface. A POSITA would understand that providing a display and user interface would be the normal and expected operation of the gateways running on Linux. Accordingly, in light of Buddhikot's Linux teachings

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

for its gateway, a POSITA would have found it obvious to provide a display and user interface for the gateways, like the MobileHotSpot Gateway 1440. For the same reasons, as discussed with Linux, a POSITA would have likewise been motivated to provide a display and user interface with Buddhikot's gateways because they can be "implemented in off-the-shelf rack-mountable PC servers." Buddhikot at 7:50-52. A POSITA would understand "PC" to refer to a Microsoft Windows based implementation, and would further understand that Windows is a GUI focused operating system synonymous with GUI access.

253. It would have further been obvious to combine Buddhikot with Lord. Lord teaches an Internet connected mobile terminal 12 (annotated in blue, below) that is connected to a base station (green) and also to laptops 14 (red) via Wi-Fi. Lord at 3:29-42, Fig. 1. As can be seen in Fig. 1, the mobile terminal 12 is an Internet connected cellular phone, which in my opinion, clearly teaches a display and a user interface. *See also* Lord at 1:42-50. A POSITA understands that the Internet connected cellular phone would provide a user interface and display. Indeed, by 2006, sophisticated operating systems for cellular phones (*e.g.*, Windows CE and Blackberry) were ubiquitously available and included Internet capable applications, displays, and user interfaces.



254. In addition to the motivations to combine described above, a POSITA would have been motivated to combine Lord and Buddhikot because they both describe handholds that have 3G and 802.11 interfaces. Buddhikot discloses a “preferred service scenario” where a “user has a laptop/handheld that has both a 3G and an 802.11 interface.” Buddhikot at 4:17-19. A laptop or handheld both have a user interface and display to allow the user to access an IP based network, like the Internet. As described above, Lord teaches a mobile terminal 12 that has a user interface and display. A POSITA would have been further motivated to combine Lord’s teachings of a user interface and display with Buddhikot’s teachings of a MobileHotSpot Gateway. As described above, both Lord and Buddhikot provide a Wi-Fi/Cellular gateway. Buddhikot describes its MobileHotSpot Gateway as mobile. Buddhikot at 23:44-49. Lord’s mobile

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

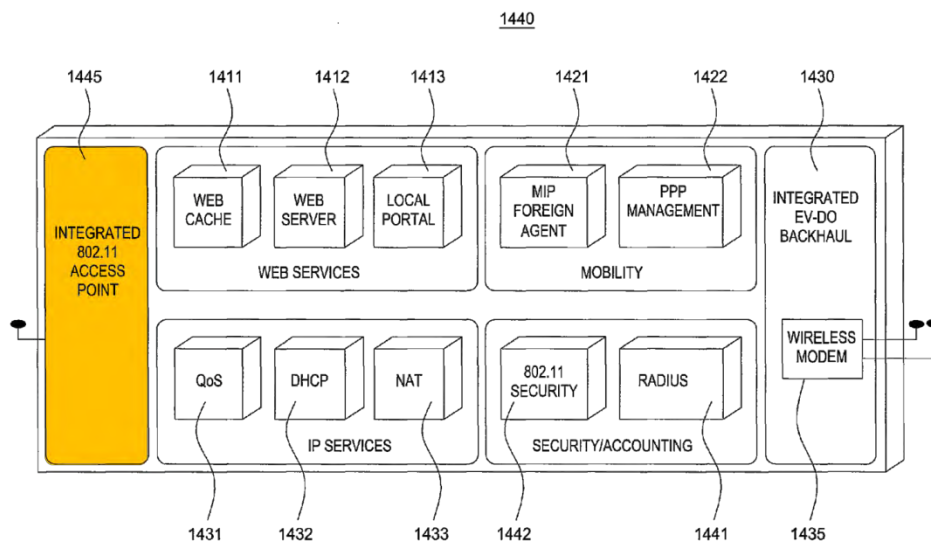
terminal is a cellular phone, which a POSITA would understand to be a mobile appliance. A POSITA would have been motivated to combine the teachings of Lord and Buddhikot to arrive at an Internet connected MobileHotSpot. Such a combination would yield predictable results, as Lord already provides the Wi-Fi/3G bridging technology.

c) “an AP module adapted to:”

255. **Element [1.C]:** Claim 1, Element [1.C] recites: “**an AP module adapted to:”**

256. Buddhikot teaches that the MobileHotSpot Gateway 1440 (the claimed computing device) includes an 802.11 Access Point 1445. Buddhikot at 23:46-39, 23:66-24:3, Fig. 14 (annotated in orange, below). The 802.11 Access Point 1445 provides Wi-Fi access. *Id.* at 10:33-44. A POSITA would understand that the 802.11 access point has corresponding software to provide wireless functionality.

FIG. 14



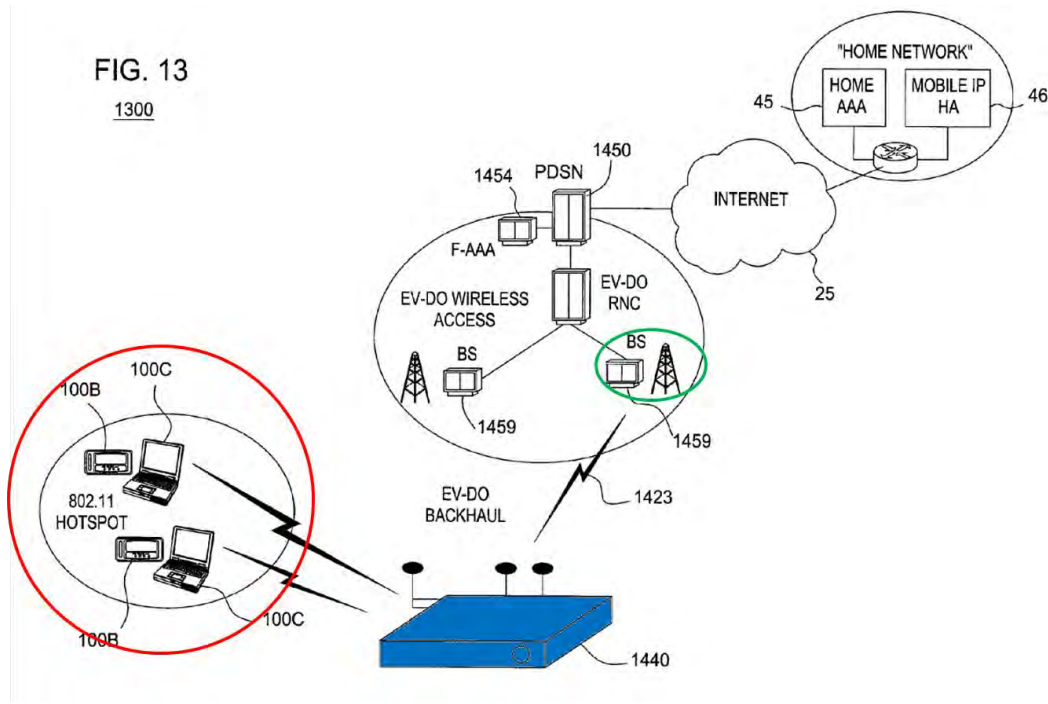
(1) (1) provide a given device of the other wireless enabled computing devices with access to the IP based network by causing said computing device to serve the given device as a second AP having a second APID, distinct from the first APID, and provide the given device access to the network via the first AP; and”

257. **Element [1.C.1]:** Claim 1, Element [1.C.1] recites: **(1) provide a given device of the other wireless enabled computing devices with access to the IP based network by causing said computing device to serve the given device as a second AP having a second APID, distinct from the first APID, and provide the given device access to the network via the first AP; and”**

258. Buddhikot teaches a computing device (the MobileHotSpot Gateway 1440 annotated in blue, below) that provides a given device of the other wireless enabled computing devices (the mobile nodes 100B/C, red) with access to the IP

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

based network (the Internet) via the first AP (the Base Station 1459, green).
Buddhikot at 23:46-61, 6:30-45.



259. As illustrated above and as described in more detail in [1.C], the MobileHotSpot Gateway 1440 (the claimed said computing device) serves as an 802.11 access point for the mobile nodes (the claimed serve the given device as a second AP). A POSITA would understand that the 802.11 access point has a SSID (the claimed second APID). *See also* Buddhikot at 17:16-62 (stating that the client would need 802.11 ESSIDs to connect to the 802.11 access point). This is described in both EX1012 at ¶180-184 and EX1033, U.S. Patent No. 6,950,628 at 4:13-16. A POSITA understands that the SSID is an alphanumeric identifier used to identify a Wi-Fi access point (e.g., “AirPort WiFi,” “Ted’s WiFi,” etc.),

allowing devices to connect to it. The SSID (the claimed second APID) for the MobileHotSpot Gateway's 802.11 access point is distinct from the base station's Cell ID (the claimed first APID).

(2) “(2) tunnel data traffic from the given device, through said computing device, through the first AP, through the IP network, to a proxy server, such that the proxy server acts as a proxy of the given device and the data traffic is secure from said computing device and first AP”

260. **Element [1.C.2]:** Claim 1, Element [1.C.2] recites: **“(2) tunnel data traffic from the given device, through said computing device, through the first AP, through the IP network, to a proxy server, such that the proxy server acts as a proxy of the given device and the data traffic is secure from said computing device and first AP”**

261. As explained in [1.C] and [1.C.1] Buddhikot teaches that data traffic from the mobile node (the claimed given device), is directed through the 802.11 access point, through the MobileHotSpot Gateway (the claimed computing device), to the Internet through the base station (the claimed first AP).

262. Buddhikot also teaches that the data traffic is tunneled (*e.g.*, with IPsec tunneling) from the mobile node through the Internet to a tunnel end-point (the claimed proxy server acting as a proxy of the given device), to provide “end-to-end privacy solutions.” Buddhikot at 17:44-50, 18:66-19:3. Because of

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

Buddhikot's teaching of IPsec, a POSITA would understand the tunnel end-point gateway would necessarily exist for the mobile node to connect to. A POSITA would understand that, with tunneling (*e.g.*, IPsec), a client encapsulates packets, then tunnels them to an end-point gateway that de-encapsulates them and sends them to an intended recipient on behalf of the client. EX1017 at 3-5, §3.2.1; EX1018; EX1019 at 10, §3.2; EX1016 at 3:54-60, 4:27-39; EX1015 at 9, §1.6; EX1014 at 96. Encapsulation is also discussed at ¶¶127-129. A POSITA would understand that packets are tunneled from a mobile node to a tunneled-to end-point gateway, which is a proxy server that acts as a proxy of the mobile node. A POSITA would understand that tunneling traffic to a proxy server is a commonly employed feature of Internet communication. Indeed, Buddhikot is replete with discussion of Mobile IP tunneling. *See, e.g.*, Buddhikot at 13:37-38, 15:16-18, 19:40-42, 22:4-5, 21:61-62; *see also* '306 patent at 4:25-29 (recognizing IP tunneling capability).

263. In addition to the above, Buddhikot also teaches providing a VPN (virtual private network) end-point gateway. Buddhikot at 17:44-50, 18:66-19:3. Because of Buddhikot's teaching of VPN, a POSITA would understand that a VPN gateway would necessarily exist for the mobile node to connect to. VPNs are well-known in the art to provide *inter alia* access to a VPN and the Internet,

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

through the VPN. *See also* U.S. Patent No. 8,411,650 at Abstract; Moran at Abstract. A POSITA would understand that a VPN gateway would act as an entry point to a VPN, and would act as a proxy for the particular client. *See also* EX1034, U.S. Patent No. 8,411,650 at 3:45-56, 3:8-12, 4:22-24. Accordingly, a POSITA would understand that the VPN gateway would be a proxy server that acts as a proxy of the mobile node.

264. The tunneled data traffic is secure from the MobileHotSpot Gateway 1440 (said computing device) and the base station (the first AP). Buddhikot teaches “end-to-end privacy” with IPSec/VPN or SSL encryption while stating that air-link only encryption is insufficient because it does not secure data from the 802.11 access point. Buddhikot at 9:53-58. Buddhikot at 17:44-50, 18:66-19:3. In other words, Buddhikot teaches IPSec/VPN or SSL to *prevent* the MobileHotSpot Gateway from snooping on client data packets.

265. Shown below in annotated Fig. 15, is a block diagram of a client (*e.g.*, Buddhikot at 4:9-10) showing an VPN/IPSec Control and Client Drivers, which illustrates that the VPN/IPSec encrypted tunnel starts at the mobile node, and would end at an end-point gateway. *See also* Buddhikot at 18:66-19:3. Accordingly, utilizing IPSec/VPN (along with or without SSL), traffic would be tunneled from mobile node (the claimed given device) through the MobileHotSpot

Gateway 1440 (said computing device), through the base station (the first AP), to a tunnel end-point gateway (the proxy server that acts as a proxy for the given device). This traffic would be secure from the MobileHotSpot Gateway and the base station access point. Furthermore and addition to the above, Buddhikot provides an authentication proxy for its gateways (*e.g.*, Fig. 2 Element 241), and it would have been obvious to include an authentication gateway as the end-point of the tunnel.

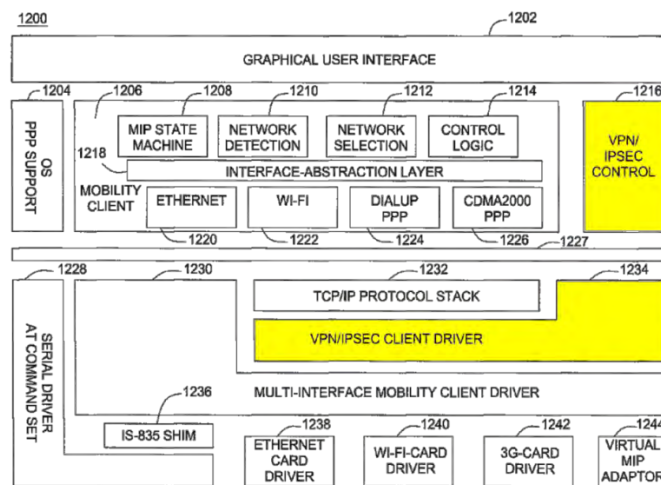


FIG. 15

(3) “and the given device operates on the network using a second public IP address distinct from the first public IP address, with the second public IP address associated with the given device.”

266. **Element [1.C.3]:** Claim 1, Element [1.C.3] recites: “**and the given device operates on the network using a second public IP address distinct from**

the first public IP address, with the second public IP address associated with the given device.”

267. Buddhikot teaches that the MobileHotSpot Gateway’s public IP address (the claimed first public IP address) is distinct from the public IP address associated with the mobile node (the given device’s public IP address). Buddhikot teaches this separately in both its Mobile IP and Simple IP embodiments, discussed below. Buddhikot at 6:30-45, 16:20-52.

268. A POSITA would understand that, under the Mobile IP framework, the mobile node (the claimed given device) would retain a fixed public IP address—its home address, (the claimed second public IP address) obtained from a DHCP server in its home network. *See also* Buddhikot at 20:24-47, 14:54-56; RFC3344 at p. 5, §1.5; Sharma at 3:1-16. The mobile node would use its home address in outgoing traffic as the source address even when it is in a foreign network. Buddhikot at 14:54-56. In the case that the mobile node is in a foreign network, *e.g.* within the network of the MobileHotSpot Gateway 1440, any packet addressed to the mobile node’s home address would be received by a home agent in the home network and forwarded to the MobileHotSpot Gateway 1440 via a care of public IP address. Buddhikot at 14:54-56, 18:38-42; *see also* Sharma at 3:9-27. As described in [1.A], the MobileHotSpot Gateway 1440 has a Mobile IP

foreign agent 1421 and a public IP address which is used in the Mobile IP framework as a “care of” address. *See, e.g.*, Buddhikot at Fig. 14; *id.* at 24:26-27, 25:1-23. A POSITA would understand that the MobileHotSpot Gateway’s “care of” and the mobile node’s “home address” to be two distinct public IP addresses. Buddhikot teaches this limitation with its Mobile IP embodiment.

269. Buddhikot also teaches this limitation with its Simple IP embodiment. In Simple IP, “[o]nce the client starts up, it gets its IP address through DHCP.” Buddhikot at 16:38-39. A POSITA would understand that this would provide the mobile node with a public IP address. As described in [1.B], the MobileHotSpot Gateway obtains a public IP address, obtained statically or dynamically. Buddhikot at 24:64-25:1-3, 25:29-34. A POSITA would understand that these two public IP addresses are distinct.

270. Furthermore, Patent Owner would be incorrect to argue that the NAT disclosure (Buddhikot at 24:36-43) necessarily means that the mobile node only has a private IP address. First and foremost, Buddhikot states that a NAT is not required to be on the MobileHotSpot Gateway. Buddhikot at 24:40-43. Absent a NAT on the MobileHotSpot Gateway, a POSITA would understand that the mobile node would necessarily receive a public IP address that is distinct from the MobileHotSpot Gateway’s public IP address. Second, there is no indication that

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

the NAT performs many-to-one mappings or many-to-many mappings. *See also* EX1035, U.S. Pub. No. 2005/0054343 at ¶¶5-6; EX1021, RFC2663 at p. 11, §4.1.1. In the case of many-to-many mappings, a POSITA would understand that the mobile node would have a private IP, but that private IP would be mapped to a public IP that is distinct from the public IP of the gateway.

271. Accordingly, Buddhikot's Simple IP embodiment teaches this limitation.

272. In my opinion, it would have been further obvious to combine Buddhikot's teachings of a Wi-Fi/cellular hotspot with Lord's teachings of providing unique public IPs for all devices in a Wi-Fi/cellular hotspot. As described in [1.B], Lord teaches an Internet connected cellular phone that acts as a Wi-Fi hotspot. Lord at 3:29-42. Lord teaches providing each individual device on the network, including the mobile terminal 12 (the cell phone) and the terminal equipment 14 (the laptops), "a unique IP address, through a single MT and a single wireless link to a PDN" where "[p]referably, the invention provides public IP addresses rather than private IP addresses so that each device can be addressed from PDNs such as the Internet." Lord at 2:21-25, 4:41-51; *see also id.* at 3:18-22.

273. In addition to the general motivations described above and in [1.B], a POSITA would have combined Lord and Buddhikot because providing unique public IP addresses was well-known, and would lead to predictable results. Using the Internet with a public IP address is commonly known and practiced. Providing public IP addresses for all networked devices, as taught in Lord, is a simple substitution. No matter any anticipated novelty argument Patent Owner can present to state that Buddhikot does not disclose public IP addresses, making them public IP addresses as taught in Lord instead is a simple substitution.

2. Dependent claims 2-12, 14, and 15

274. **Claim 2:** Claim 2 recites: “**The computing device of claim 1 wherein the second APID is associated with the proxy server.**”

275. As discussed above for limitation [1.C.1], the second APID is the SSID for the 802.11 access point in the MobileHotSpot Gateway.

276. In my opinion, a POSITA would understand that the BRI for “associated” to be a weak association (*i.e.*, the proxy server does not have to be the same server that provides the second AP). As discussed above for limitation [1.C.2], the end-point of the tunnel is the “proxy server [that] *acts as a proxy* of the given device.” Claim 2 states that the “second APID is *associated* with the proxy server,” but claim [1.C.2] states that the “data traffic is *secure from* said

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

computing device [(i.e., the hotspot)].” (emphasis added). Because data must be secure from the hotspot, a POSITA would understand that this claim does not place the proxy on the hotspot. A POSITA would understand that the association would be weak and would include a connection point to access the proxy.

277. In my opinion, a POSITA would understand that the MobileHotSpot’s SSID would be associated with the proxy server because it is a point along the client to proxy tunnel, and because the client must necessarily connect to the MobileHotSpot Gateway’s SSID (the second APID) before connecting to the tunnel gateway. Accordingly, Buddhikot teaches this limitation.

278. In my opinion, Buddhikot also teaches that its gateways (like the MobileHotSpot gateway) can have a proxy (e.g., Fig. 2 Element 241, annotated in yellow below) and other proxy type services (e.g., foreign 202 and home 222 agents and web services 201). Buddhikot at 24:36-55. A POSITA would have found it obvious to include these services in the MobileHotSpot Gateway, or make the MobileHotSpot gateway itself (or any gateway embodiment disclosed in Buddhikot) be the end-point of the tunnel.

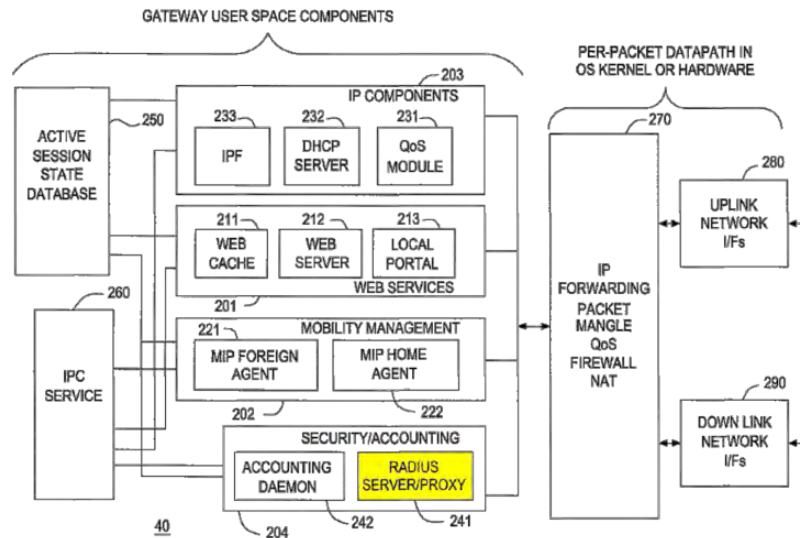


FIG. 2

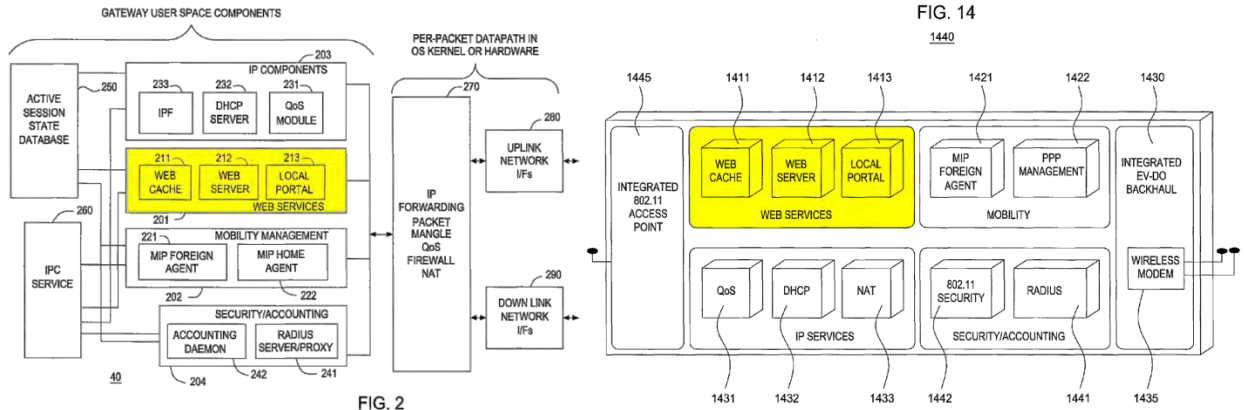
279. **Claim 3:** Claim 3 recites: “The computing device of claim 1 wherein the AP module is adapted to tunnel data traffic from the given device through said computing device, through the first AP, through the IP network, to a proxy server in response to at least activating data service for the given device through a captive portal web interface.”

280. Claim 3 states that tunneling data traffic can occur in response to activating the data service via a captive portal web interface. As explained in [1.C.2], the AP module tunnels traffic via the Internet to a tunnel end-point gateway (the proxy server).

281. Buddhikot teaches, in its Simple IP embodiment, that a log-in is required before using the Internet. Buddhikot at 16:25-29. “Once the client starts up. . . the client’s web browser [is redirected to] to the local authentication page

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

served over a Secure Socket Layer (SSL) connection.” *Id.* at 16:34-52. These portals are included on any of the gateways. *Id.* at Fig. 2, Element 213, Fig. 14, Element 1413 (annotated below). Accordingly, Buddhikot teaches this limitation.



282. **Claim 4:** Claim 4 recites: “**The computing device of claim 1 wherein the first wireless AP is included in a cellular telephone network.**”

283. Buddhikot teaches a 3G cellular network base station access point (first access point). Buddhikot at 23:44-24:18; Figs. 12-13

284. **Claim 5:** Claim 5 recites: “**The computing device of claim 4 wherein the first wireless AP is a cellular cell.**”

285. Buddhikot is a 3G cellular network base station access point, which is synonymous with a “cellular cell.” Buddhikot at 23:44-24:18; Figs. 12-13.

286. **Claim 6:** Claim 6 recites: “**The computing device of claim 1 wherein the AP module is controlled by a network entity in a cellular system.**”

287. Buddhikot teaches this limitation because, as described in claim 3, access to the Internet is controlled via an authentication entity. Access to the cellular network needs to be granted/not-granted—“the Authentication, Authorization and Accounting (AAA) server of the 3G carrier”—which is a network entity in a cellular system—is responsible for “authentica[ting] the user” for “the 802.11 provider,” *i.e.* the AP module as was discussed above for limitation [1.C].

288. In my opinion, a POSITA would understand that the MobileHotSpot Gateway would be controlled by a network provider (*i.e.*, a network entity in a cellular system).

289. **Claim 7:** Claim 7 recites: “**The computing device of claim 1 wherein the first wireless AP and the second wireless AP use different wireless communication protocols.**”

290. Buddhikot teaches a 3G cellular wireless connection for the first wireless AP, and an 802.11 access point for the second wireless AP, which are different wireless communications protocols. Buddhikot at 23:45-50.

291. **Claim 8:** Claim 8 recites: “**The computing device of claim 1 wherein the first wireless AP is included in a terrestrial wireless network.**”

292. Buddhikot teaches a base station access point (the claimed first

wireless AP) which is included in a 3G network (the terrestrial wireless network).
Buddhikot at 23:44-24:18.

293. **Claim 9:** Claim 9 recites: “**The computing device of claim 1 wherein the proxy server acts as a proxy of the given device for the given device to interact with destinations over the IP based network.**”

294. As discussed above [1.C.2], Buddhikot discloses a proxy server that acts as a proxy of the given device. As also discussed in [1.C.2], at the end of an tunnel, the gateway de-capsulates a packet, and forwards that packet to the ultimate destination over the Internet. As also discussed in [1.C.2], the gateway can provide VPN services, which a POSITA would understand to provide a VPN connection to the Internet. A POSITA would understand that the tunnel endpoint and VPN gateway each interact with Internet destinations.

295. **Claim 10:** Claim 10 recites: “**The computing device of claim 1 wherein the AP module is adapted to prevent data packets destined for the user interface and display from being accessed by the given device.**”

296. Lord teaches that its mobile terminal 12 and laptops 14 are provided unique public IP addresses (Lord at 4:41-51), where each device is separated into a virtual one-to-one links. *Id.* at 5:54-67, Abstract, 2:33-42. A POSITA would understand that any traffic directed to the mobile terminal’s IP address (the

claimed computing device that has the AP module) would not be forwarded to any of the other Wi-Fi connected devices.

297. **Claim 11:** Claim 11 recites: “**The computing device of claim 1 wherein the AP module is adapted to disconnect the given device after a predetermined period.**”

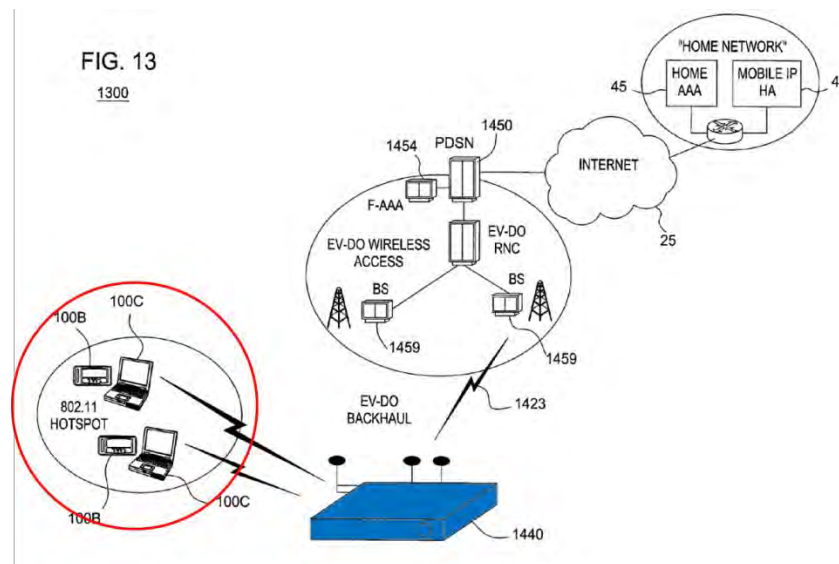
298. Buddhikot discloses that “time and duration data” for a network “user session” can be obtained for “time-based” pricing for that session. Buddhikot at 13:65-14:10. Further, as discussed in claim 3, Buddhikot discloses a log-in process before accessing the Internet. Buddhikot at 16:34-42. It was obvious to POSITA that after the time period, the user would be disconnected from the Internet by the MobileHotSpot Gateway.

299. **Claim 12:** Claim 12 recites: “**The computing device of claim 1 wherein the second public IP address is shared by the given device with at least one other device of the other wireless enabled computing devices.**”

300. As discussed above for limitation [1.C.3], Buddhikot teaches a Mobile IP implementation where a mobile node (given device) has a home address (second public IP address) which it retains in the foreign network. Buddhikot discloses Wi-Fi enabled user Laptops and PDAs. Buddhikot at 2:28-29, Figs. 12-13, Elements 100B and 100C. Buddhikot states that 100B and 100C represents

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

devices of individual users (*i.e.*, user C has two devices).¹² Buddhikot at 24:9. A POSITA would understand that, under a Mobile IP implementation, a user's two devices could share the same care-of-address.



301. Furthermore, the gateways also “support[] private addressing schemes, using the NAT implementation.” Buddhikot at 16:34-52 (emphasis added). It was obvious to a POSITA that a NATed network can provide many-to-one mappings (*i.e.*, where the devices behind the NAT share a single public IP address to the external world). *See also* EX1035, U.S. Pub. No. 2005/0054343 at ¶¶5-6; RFC2663 at p. 11, §4.1.1. Accordingly, even if Mobile IP were not

¹² A POSITA reading this and Fig. 13 would also understand that an individual user can have a PDA/Laptop.

implemented, it would have been obvious for a POSITA to provide a single public IP address via the NAT functionality.

302. **Claim 14:** Claim 14 recites: “**The computing device of claim 1 wherein the AP module is adapted to prevent the given device from accessing internal IP addresses associated with the computing device.**”

303. Buddhikot teaches a dynamic firewall (*e.g.*, Linux IP Filter architecture) that protects the gateway (the claimed given device) against traffic coming from malicious clients. Buddhikot at 10:55-59, 11:4-8. A POSITA would understand that the Linux IP Filter architecture allows the gateway to specify filtering rules to restrict traffic to or from a group of IP addresses. A POSITA would further understand that a user of the gateway can therefore disallow traffic to the internal IP addresses of the gateway by specifying so in the filtering rules. A POSITA would have had found it obvious to specify that a client on the Wi-Fi network is not allowed to access other clients on the Wi-Fi network. Indeed, having security measures in place to prevent other Wi-Fi hotspot connected users from accessing one’s laptop would have been the normal and expected operation of a Wi-Fi hot spot. Furthermore, in my opinion, a POSITA would have had found it obvious to specify that a client on the Wi-Fi network is not allowed to

access other clients on the Wi-Fi network through MAC address filtering, for example.

304. In addition, Lord discloses virtually separating the physical network connection for each device on its WLAN connected to the Internet (Lord at 2:33-39), which creates separate logical connections with each device (*id.* at 3:57-62). A POSITA would understand that, because these connections are logically separated and have a public IP address, that the mobile terminal would treat them as Internet traffic. A POSITA understands that a NAT and a firewall would prevent Internet traffic from accessing internal IP addresses of the router.

305. Having security measures in place to prevent other Wi-Fi hotspot connected users from accessing one's laptop would have been the normal and expected operation of a Wi-Fi hot spot. In my opinion, a POSITA would have expected security preventing Wi-Fi connected devices from accessing each-other when connecting to a public Wi-Fi.

306. **Claim 15:** Claim 15 recites: **“The computing device of claim 1 wherein the AP module is adapted to restrict the given device from accessing a predetermined set of IP addresses.”**

307. As described in claim 14, in my opinion a POSITA would have found it obvious to restrict one Wi-Fi hotspot connected device (given device) from

accessing the other devices (the predetermined set of IP addresses the all other devices that the given device is prevented access to). Furthermore, a POSITA would have also understood to set the predetermined set of IP addresses specified in a filtering rule used in the Linux IP Filter Architecture.

3. Independent claim 16

308. **Claim 16, Preamble:** The Claim 16 preamble recites: “**A system comprising:**”

309. The teachings of the system, with respect to the claimed devices, was explained with respect to elements of claim 1.

310. **Element [16.A]:** Claim 16, Element [16.A] recites: “**a first wireless access point (AP) connected to an IP based network, the first wireless AP having a first AP Identification (APID);**”

311. [16.A] is substantively similar to limitation [1.A.1], and is taught for the same reasons.

312. **Element [16.B]:** Claim 16, Element [16.B] recites: “**a proxy server connected to the IP based network and adapted to act as a proxy of at least a subset of computing devices that connect via the first wireless AP; and**”

313. [16.B] is substantively similar to a portion of limitation [1.C.3] , and is taught for the same reasons.

314. **Element [16.C]:** Claim 16, Element [16.C] recites: “**a first computing device having a user interface, wherein the first computing device is adapted to: wirelessly connect to the IP based network via the first wireless AP; wirelessly communicate with other wireless enabled computing devices;**”

315. [16.C] is substantively identical to limitation [1.A.2] combined with a portion of limitation [1.B], and is taught for the same reasons. For clarity, the first computing device maps to the claimed computing device of claim 1 (*i.e.*, the hot-spot).

316. **Element [16.D]:** Claim 16, Element [16.D] recites: “**enable a user of the first computing device to interact, through the user interface, with destinations over the IP based network, through the first wireless AP, using a first public IP address associated with the first computing device;**”

317. [16.D] is substantively identical to a portion of limitation [1.B], and is taught for the same reasons.

318. **Element [16.E]:** Claim 16, Element [16.E] recites: “**provide a second computing device of the other wireless enabled computing devices with access to the IP based network by causing the first computing device to serve the second computing device as a second AP having a second APID, distinct from the first APID, and provide the second computing device access to the IP**

based network via the first AP; and”

319. [16.E] is substantively identical to limitation [1.C.1], and is taught for the same reasons. For clarity, the second computing device maps to the given device of claim 1 (*i.e.*, the client device connected to the hot-spot).

320. **Element [16.F]:** Claim 16, Element [16.F] recites: **“tunnel data traffic from the second computing device, through the first computing device, through the first AP, through the IP network, to the proxy server, wherein the proxy server acts as a proxy of the second computing device and the data traffic is secure from the first computing device and”**

321. [16.F] is substantively identical to limitation [1.C.2], and is taught for the same reasons.

322. **Element [16.G]:** Claim 16, Element [16.G] recites: **“the first AP and the second computing device operates on the IP based network using a second public IP address distinct from the first public IP address, with the second public IP address associated with the second computing device.”**

323. [16.G] is substantively identical to limitation [1.C.3], and is disclosed for the same reasons.

4. Dependent claims of Claim 16

324. **Element 17[a]:** Claim 17, Element 17[a] recites: **“The system of**

claim 16 wherein the first computing device is further adapted to: provide a third computing device of the other wireless enabled computing devices with access to the IP based network by causing the first computing device to serve the third computing device as the second AP having the second APID, distinct from the first APID, and provide the third computing device access to the IP based network via the first AP; and”

325. [17.A] adds another Wi-Fi device to the hotspot. As explained in [1.C.1], Buddhikot and Lord teach providing hotspot access multiple clients. Buddhikot at Fig. 13; Lord at Fig. 1.

326. **Element 17[b]:** Claim 17, Element 17[b] recites: **“tunnel data traffic from the third computing device, through the first computing device, through the first AP, through the IP network, to the proxy server, wherein the proxy server acts as a proxy of the third computing device and the data traffic is secure from the first computing device and the first AP and the third computing device operates on the IP based network using a third public IP address distinct from the first public IP address, with the third public IP address associated with the third computing device.”**

327. [17.B] applies the tunneling and IP address limitations [16.F] and [16.G] (which are substantively identical to limitations [1.C.2] and [1.C.3]) to the

extra connected device added by [17.A]. As a result, this limitation is disclosed for the same reasons as discussed for limitations [1.C.2], [1.C.3]. and [17.A].

328. **Claim 18:** Claim 18 recites: “**The system of claim 16 wherein the second APID is associated with the proxy server.**”

329. This claim is substantively identical to claim 2, and is disclosed for the same reasons.

330. **Claim 19:** Claim 19 recites: “**The system of claim 16 wherein the proxy server acts as a proxy of the second computing device for the second computing device to interact with destinations over the IP based network.**”

331. This claim is substantively identical to claim 9, and is disclosed for the same reasons.

332. **Claim 20:** Claim 20 recites: “**The system of claim 16 wherein the second computing device connects to the IP based network through a third AP having a third APID, distinct from the first APID and the second APID, concurrently with connecting to the IP based network through the second AP.**”

333. Claim 20 describes the second computing device (the hotspot) connecting to a third access point to access the Internet, separate from the claimed

first access point (the base station). It also describes the ability to seamlessly choose between the two interfaces.

334. Buddhikot teaches an interface selection algorithm, which probes for the best wireless network (*i.e.*, there are multiple cellular network cells—at least a third AP having a third APID) to use for Internet traffic. Buddhikot at 20:48-52; The algorithm may choose between wireless networks based on signal strength or priority (*id.* at 20:54-57, 21:18-32) and that there can be any number of wireless interfaces (*id.* at 20:64-65). Buddhikot also teaches an interface driver that intercepts packets from the TCP/IP stack, and seamlessly determines which interface to use to send the packets. *Id.* at 19:36-42, 19:47-53.

335. **Claim 21:** Claim 21 recites: “**The system of claim 16 wherein the first computing device is adapted to prevent data packets destined for the user interface from being accessed by the second computing device.**”

336. This claim is substantively identical to claim 10, and is disclosed for the same reasons.

337. **Claim 22:** Claim 22 recites: “**The system of claim 16 wherein the first computing device is adapted to disconnect the second computing device from the second AP after a predetermined period.**”

338. This claim is substantively identical to claim 11, and is disclosed for

the same reasons.

339. **Claim 26:** Claim 26 recites: “**The system of claim 16 wherein the first computing device is adapted to tunnel data traffic from the second computing device through the first computing device, through the first AP, through the IP network, to a proxy server in response to at least activating data service for the second computing device through a captive portal web interface.**”

340. This claim is substantively identical to claim 3, and is disclosed for the same reasons.

341. **Claim 27:** Claim 27 recites: “**The system of claim 16 wherein the first wireless AP is included in a cellular telephone network.**”

342. This claim is substantively identical to claim 4, and is disclosed for the same reasons.

343. **Claim 28:** Claim 28 recites: “**The system of claim 27 wherein the first wireless AP is a cellular cell.**”

344. This claim is substantively identical to claim 5, and is disclosed for the same reasons.

345. **Claim 29:** Claim 29 recites: “**The system of claim 16 wherein the second AP is controlled, at least in part, by a network entity in a cellular**

system.”

346. This claim is substantively identical to claim 6, and is disclosed for the same reasons.

347. **Claim 30:** Claim 30 recites: “**The system of claim 16 wherein the first wireless AP and the second wireless AP use different wireless communication protocols.**”

348. This claim is substantively identical to claim 7, and is disclosed for the same reasons.

349. **Claim 31:** Claim 31 recites: “**The system of claim 16 wherein the first wireless AP is included in a terrestrial wireless network.**”

350. This claim is substantively identical to claim 8, and is disclosed for the same reasons.

351. **Claim 34:** Claim 34 recites: “**The system of claim 16 wherein the tunneled data traffic includes data packets for use in conducting a IP based phone call.**”

352. Buddhikot teaches that a user’s “ongoing network sessions (e.g., network radio, voice chat etc)...should be transparently maintained” when a user switches from the 3G cellular to the 802.11 network. Buddhikot at 4:35-64. A POSITA would understand “voice chat” over the Internet to include an IP based

phone call. In my opinion, it was obvious to POSITA that data traffic tunneled through the Internet would include “data packets for use in conducting an IP based phone call.”

353. **Claim 35:** Claim 35 recites: “**The system of claim 16 wherein the tunneled data traffic includes data packets representing data to be uploaded to a remote server from the second computing device.**”

354. In my opinion, a POSITA would understand that data traffic tunneled through to an IP based network like the Internet would include “data to be uploaded to a remote server.” Moreover, Buddhikot teaches FTP, so it is obvious to a POSITA that a file could be uploaded to an external server because FTP is well-known to be used for upload and download. Buddhikot at 22:43-49. Uploading data is an expected use of the Internet to a POSITA.

355. **Claim 36:** Claim 36 recites: “**The system of claim 16 wherein data packets received by the second computing device from a destination through the proxy server, through the IP network, through the first AP, through the first computing device, through the second AP include at least one of a picture, video, or audio.**”

356. In my opinion it was obvious to a POSITA that data traffic received through the Internet would include “a picture, video, or audio.” Buddhikot

teaches that when a user switches from the 3G cellular network to an 802.11 network, the user's "ongoing network sessions (e.g., network radio, voice chat etc)...should be transparently maintained" and "network radio" over an IP based network is a type of audio data (as is "voice chat"). *Buddhikot* at 4:35-64. Transferring picture, video, or audio is an obvious and common use of the Internet, well known to POSITA.

357. **Claim 37:** Claim 37 recites: "**The system of claim 16 wherein the second public IP address is shared by the second computing device with at least one other device of the other wireless enabled computing devices.**"

358. This claim is substantively identical to claim 12, and is disclosed for the same reasons.

359. **Claim 39:** Claim 39 recites: "**The system of claim 16 wherein the first computing device is adapted to prevent the second computing device from accessing internal IP addresses associated with the first computing device.**"

360. This claim is substantively identical to claim 14, and is disclosed for the same reasons.

361. **Claim 40:** Claim 40 recites: "**The system of claim 16 wherein the first computing device is adapted to restrict the second computing device from**

accessing a predetermined set of IP addresses.”

362. This claim is substantively identical to claim 15, and is disclosed for the same reasons.

5. Claim 43

363. **Claim 43, Preamble:** The Claim 43 preamble recites: “**A method comprising:**”

364. As discussed for Claim 1, Buddhikot discloses a “gateway for mobile access” for client devices, which can perform the method in Claim 43 and its dependent claims. Buddhikot at 23:45-50.

365. **Element [43.A]:** Claim 43, Element [43.A] recites: “**wirelessly connecting a first computing device to an IP based network via a first wireless access point (AP) having a first AP Identification (APID), wherein the first computing device wirelessly communicates with other wireless enabled computing devices;**”

366. [43.A] is substantively identical to limitation [1.A], and is taught for the same reasons.

367. **Element [43.B]:** Claim 43, Element [43.B] recites: “**enabling a user of the first computing device to interact, through a user interface of the first computing device, with destinations over the IP based network, through the**

first wireless AP, using a first public IP address associated with the first computing device;”

368. [43.B] is substantively identical to limitation [1.B], and is taught for the same reasons.

369. **Element [43.C]:** Claim 43, Element [43.C] recites: **“providing a second computing device of the other wireless enabled computing devices with access to the IP based network via the first wireless AP by causing the first computing device to serve the second computing device as a second AP having a second APID, distinct from the first APID, and provide the second computing device access to the IP based network via the first AP; and”**

370. [43.C] is substantively identical to limitations [1.C] and [1.C.1], and is taught for the same reasons.

371. **Element [43.D]:** Claim 43, Element [43.D] recites: **“tunneling data traffic from the second computing device, through the first computing device, through the first AP, through the IP network, to a proxy server, wherein the proxy server acts as a proxy of the second computing device and the data traffic is secure from the first computing device and the first AP and”**

372. [43.D] is substantively identical to limitation [1.C.2], and is taught for the same reasons.

373. **Element [43.E]:** Claim 43, Element [43.E] recites: “**the second computing device operates on the IP based network using a second public IP address distinct from the first public IP address, with the second public IP address associated with the second computing device.**”

374. [43.E] is substantively identical to limitation [1.C.3], and is taught for the same reasons.

6. Dependent Claims of Claim 43

375. **Claim 47:** Claim 47 recites: “**The method of claim 43 wherein the second APID is associated with the proxy server.**”

376. This claim is substantively identical to claim 2, and is disclosed for the same reasons.

377. **Claim 48:** Claim 48 recites: “**The method of claim 43 further comprising tunneling data traffic from the second computing device through the first computing device, through the first AP, through the IP network, to a proxy server in response to at least activating data service for the second computing device through a captive portal web interface.**”

378. This claim is substantively identical to claim 3, and is disclosed for the same reasons.

379. **Claim 49:** Claim 49 recites: “**The method of claim 43 wherein the**

first wireless AP is included in a cellular telephone network.”

380. This claim is substantively identical to claim 4, and is disclosed for the same reasons.

381. **Claim 50:** Claim 50 recites: “**The method of claim 49 wherein the first wireless AP is a cellular cell.**”

382. This claim is substantively identical to claim 5, and is disclosed for the same reasons.

383. **Claim 51:** Claim 51 recites: “**The method of claim 43 wherein tunneling data traffic is controlled, at least in part, by a network entity in a cellular system.**”

384. This claim is substantively identical to claim 6, and is disclosed for the same reasons.

385. **Claim 52:** Claim 52 recites: “**The method of claim 43 wherein the first wireless AP and the second wireless AP use different wireless communication protocols.**”

386. This claim is substantively identical to claim 7, and is disclosed for the same reasons.

387. **Claim 53:** Claim 53 recites: “**The method of claim 43 wherein the first wireless AP is included in a terrestrial wireless network.**”

388. This claim is substantively identical to claim 8, and is disclosed for the same reasons.

389. **Claim 56:** Claim 56 recites: “**The method of claim 43 wherein the proxy server acts as a proxy of the second computing device for the second computing device to interact with destinations over the IP based network.**”

390. This claim is substantively identical to claim 9, and is disclosed for the same reasons.

391. **Claim 57:** Claim 57 recites: “**The method of claim 43 wherein the second computing device connects to the IP based network through a third AP having a third APID, distinct from the first APID and the second APID, concurrently with connecting to the IP based network through the second AP.**”

392. This claim is substantively identical to claim 20, and is disclosed for the same reasons.

393. **Claim 58:** Claim 58 recites: “**The method of claim 43 further comprising preventing data packets destined for the user interface from being accessed by the second computing device.**”

394. This claim is substantively identical to claim 10, and is disclosed for the same reasons.

395. **Claim 59:** Claim 59 recites: “**The method of claim 43 further comprising disconnecting the second computing device after a predetermined period.**”

396. This claim is substantively identical to claim 11, and is disclosed for the same reasons.

397. **Claim 60:** Claim 60 recites: “**The method of claim 43 wherein the tunneled data traffic includes data packets for use in conducting a IP based phone call.**”

398. This claim is substantively identical to claim 34, and is disclosed for the same reasons.

399. **Claim 61:** Claim 61 recites: “**The method of claim 43 wherein the tunneled data traffic includes data packets representing data to be uploaded to a remote server from the second computing device.**”

400. This claim is substantively identical to claim 35, and is disclosed for the same reasons.

401. **Claim 62:** Claim 62 recites: “**The method of claim 43 wherein data packets received by the second computing device from a destination through the proxy server, through the IP network, through the first AP, through the first computing device, through the second AP include at least one of a**

picture, video, or audio.”

402. This claim is substantively identical to claim 36, and is disclosed for the same reasons.

403. **Claim 63:** Claim 63 recites: **“The method of claim 43 wherein the second public IP address is shared by the second computing device with at least one other device of the other wireless enabled computing devices.”**

404. This claim is substantively identical to claim 12, and is disclosed for the same reasons.

405. **Claim 65:** Claim 65 recites: **“The method of claim 43 further comprising preventing the second computing device from accessing internal IP addresses associated with the first computing device.”**

406. This claim is substantively identical to claim 14, and is disclosed for the same reasons.

407. **Claim 66:** Claim 66 recites: **“The method of claim 43 further comprising restricting the second computing device from accessing a predetermined set of IP addresses.”**

408. This claim is substantively identical to claim 15, and is disclosed for the same reasons.

**B. Buddhikot, Lord, and Moran Render Claims 23-25, 41-42, 44-46,
and 67-68**

1. Motivation to Combine Moran With Buddhikot and Lord

409. Section VI.A provide detailed analysis regarding rational and motivation to combine Buddhikot and Lord. A POSITA would have been motivated to combine Moran with Buddhikot because they both teach using IPsec tunnels, VPN, and Mobile IP. Moran at Abstract; Buddhikot at 9:53-58, 17:44-49, 18:66-19:3. A POSITA would have further been motivated to combine Buddhikot and Moran because Buddhikot describes using IPsec and VPN and Moran provides a verbose description on how IPsec/VPN can be achieved with Mobile IP. A POSITA would have understood Buddhikot to suggest to a POSITA to seek out references like Moran that teach how to implement VPN and IPsec. A POSITA would have been motivated to combine Moran and Buddhikot because they're in analogous fields with similar focuses—they are all directed towards mobile devices, and all seek to improve upon the ability for mobile devices to roam across cellular networks. Moran at 8:8-16; Buddhikot at 10:3-32. A POSITA would have been motivated to combine Moran and Buddhikot because Moran describes an improved home agent. Buddhikot describes using IPsec/VPN, but not describe the tunnel endpoint location; Moran describes the home agent as the endpoint. Moran at Abstract; Buddhikot at 9:53-58. Further,

both Buddhikot and Moran are directed at security. Moran at 8:9-15; Buddhikot at Abstract. A POSITA would look at Buddhikot's security disclosures, and seek extra security with an IPSec/VPN implementation such as one taught in Moran.

2. Element-by-Element Analysis of the Challenged Claims

410. **Element [23.A]**: Claim 23, Element [23.A] recites: **“The system of claim 16 wherein the proxy server allocates the second public IP address for the second computing device and”**

411. As described in [1.C.2], Buddhikot teaches that a tunnel is established from the mobile node (the second computing device) to a tunnel end-point (the claimed proxy server), and also teaches establishing a VPN session from the mobile node to a VPN gateway (the claimed proxy server). As also explained in [1.C.3], the home agent allocates the home IP address.

412. It would have been obvious to have the tunnel end-point be the home agent. Moran teaches that the Mobile IP home agent (which it describes as T-HA, *e.g.*, 4:20-5:9) acts as the VPN gateway (Moran at 3:1-6, 3:23-25, 4:20-25, Figs. 1-3) and tunnel end-point gateway (*e.g.* of an IPSec tunnel, *id.* at 13:28-30, Figs. 1-3), The T-HA forwards out-going and receives in-coming traffic for the mobile node. *Id.* at 3:1-6. Moran also teaches that the home agent allocates the mobile

node's home address through the I-HA (internal home agent). *Id.* at 7:9-10, 7:27-30, 14:30-15:3.

413. **Element [23.B]:** Claim 23, Element [23.B] recites: “**forwards data packets destined for the second public IP address to a current IP address associated with the second computing device, wherein the current IP address is distinct from the first public IP address and the second public IP address.**”

414. [23.B] describes basic Mobile IP functionality and adds that a mobile node roamed and obtained a new care-of address for the foreign agent router it is at (the claimed current IP address). A POSITA would understand that, in Mobile IP, a mobile node can roam and obtain a new foreign agent care of address, which is registered at the home agent. *See also* Moran at 15:5-11. As described in more detail in [1.C.3] and [23.A], a POSITA would understand that the home agent (the claimed proxy) would forward packets received at the home address (the claimed second public IP address) to the roamed to care-of address (the claimed current address). *See also* Moran at 15:10-11. The mobile device's roamed to care-of address (the claimed current address) is distinct from the roamed-from MobileHotSpot Gateway's public IP address (the claimed first IP address) and the home address (the claimed second public IP address).

415. **Element [24.A]:** Claim 24, Element [23.A] recites: “**The system of**

claim 23 wherein the proxy server replaces the current IP address with the second public IP address in data packets destined for other servers on the IP based network,”

416. As described in claim 23, a new foreign agent care-of address is obtained when a device roams.

417. Moran teaches that its home agent (the claimed proxy server) replaces the new care-of address (the current IP address) with the home address (second public IP address) in data packets destined for servers on the IP based network. This is because the home agent de-capsulates packets from the foreign care-of address and sends them on the network using the home address.

418. As taught in Moran, two separate tunnels terminate at the home agent (the claimed proxy): (1) an IPSec tunnel; and (2) a Mobile IP tunnel, where the IPSec tunnel is encapsulated in the Mobile IP tunnel. Moran at 6:24-26, 8:32-9:1. Moran also teaches that the foreign agent can be the entry point for the Mobile IP tunnel to the home agent (1:17-8), and that the mobile node can be the entry point for the IPSec tunnel (6:24-26, 7:10-13, 7:31-33). Buddhikot likewise teaches that the mobile node can be the entry point for the IPSec tunnel. Buddhikot at 9:53-58. Moran teaches that the home agent is the end-point of the tunnels. Moran at 7:33-8:4. A demonstrative of the tunneling taught by Moran is provided below:



419. In my opinion a POSITA would understand that, at the home agent, the twice-encapsulated tunnel would have a source address of the foreign agent (the claimed current address). A POSITA would understand that, after decapsulation of both the MobileIP and the IPSec tunnels, the data packet would now have a source address which is the mobile node's home address (which the mobile node uses for all out-going traffic). The home agent would forward the packet with the home address as the source to servers on the Internet. *See also* Moran at 10:19-23. The home agent (the proxy server) replaces the foreign agent care-of address (current IP address) with the home address (second public IP address) in data packets destined for other servers on the IP based network.

420. **Element [24.B]:** Claim 24, Element [23.B] recites: “**wherein the data packets are tunneled from the second computing device, through the first computing device, through the first AP, through the IP network, to the proxy server.**”

421. This limitation is substantially similar to [1.C.2], and is taught for the same reasons. As also demonstrated in [24.A], an IPSec tunnel is established between the mobile node and home agent.

422. **Claim 25:** Claim 25 recites: “**The system of claim 23 wherein the proxy server updates the current IP address associated with the second computing device in response to the second computing device connecting through a different AP.**”

423. As discussed in [23.A], in my opinion a POSITA would have found it obvious to use the home agent gateway as the proxy. As described in [1.C.3], Buddhikot teaches that the home agent updates a device’s newest care of address (the claimed updating the current IP address) whenever a mobile node (the claimed second computing device) enters a different foreign network (the claimed connects through a different AP). *See, e.g.*, Buddhikot at Fig. 14; *id.* at 24:26-27, 25:1-23. Moran teaches the same. Moran at 15:5-7.

424. **Claim 41:** Claim 41 recites: “**The system of claim 16 wherein the proxy server provides a dynamic host configuration protocol (DHCP) service that assigns an IP address for the second computing device.**”

425. As described in [23.A], the home agent (the proxy server) allocates the mobile node’s home address (*i.e.*, it assigns an IP address for the second

computing device). A POSITA would understand that DHCP is the mechanism to allocate IP addresses, and would understand that the home agent would provide a DHCP service. *See also* Buddhikot at 15:29-31; Lord at 2:29-32.

426. **Claim 42:** Claim 42 recites: “**The system of claim 16 wherein the proxy server provides a network address translation (NAT) service that translates IP addresses for the second computing device.**”

427. As described in [23.A], the home agent is the proxy server. In my opinion, a POSITA would understand that the mobile node (the second computing device) would sometimes be in the home network, where the home agent (the proxy server) would act as its router. It would have been obvious for a POSITA to provide a NAT service at the home agent.

428. Buddhikot teaches providing NATs at gateway routers. Buddhikot at Fig. 14, 16:50-53, 24:38-44. Moran suggests the usage of a NAT with the home agent (Moran at 8:26-28, 9:14) and illustrates the home agent as a gateway router (Figs. 1-3). Further, in the typical operation of Mobile IP, a POSITA would understand it to be obvious to provide a NAT in order for the home agent to receive packets destined to the home address.

429. .

430. **Element 44[a]:** Claim 44, Element 44[a] recites: “**The method of**

claim 43 further comprising: allocating, by the proxy server, the second public IP address for the second computing device; and”

431. This limitation is substantively identical to limitation [23.A], and is disclosed for the same reasons.

432. **Element 44[b]**: Claim 44, Element 44[b] recites: “**forwarding data packets destined for the second public IP address to a current IP address associated with the second computing device, wherein the current IP address is distinct from the first public IP address and the second public IP address.**”

433. This limitation is substantively identical to limitation [23.B], and is disclosed for the same reasons.

434. **Claim 45**: Claim 45 recites: “**The method of claim 44 further comprising: replacing, by the proxy server, the current IP address with the second public IP address in data packets destined for other servers on the IP based network, wherein the data packets are tunneled from the second computing device, through the first computing device, through the first AP, through the IP network, to the proxy server.**”

435. This claim is substantively identical to claim 24, and is disclosed for the same reasons.

436. **Claim 46**: Claim 46 recites: “**The method of claim 44 further**

comprising updating, by the proxy server, the current IP address associated with the second computing device in response to the second computing device connecting through a different AP.”

437. This claim is substantively identical to claim 25, and is disclosed for the same reasons.

438. **Claim 67:** Claim 67 recites: “**The method of claim 43 wherein the proxy server provides a dynamic host configuration protocol (DHCP) service that assigns an IP address for the second computing device.**”

439. This claim is substantively identical to claim 41, and is disclosed for the same reasons.

440. **Claim 68:** Claim 68 recites: “**The method of claim 43 wherein the proxy server provides a network address translation (NAT) service that translates IP addresses for the second computing device.**”

441. This claim is substantively identical to claim 42, and is disclosed for the same reasons.

C. Buddhikot, Lord, and Abhishek Render Claims 32-33 and 54-55 Obvious

1. Motivation to Combine Abhishek With Buddhikot and Lord

442. Section VI.A provide detailed analysis regarding rational and motivation to combine Buddhikot and Lord. In my opinion, a POSITA would have been motivated to combine Buddhikot, Lord, and Abhishek because they are in the same field. Buddhikot, Lord, and Abhishek are replete with discussion of 802.11 access points to allow computing devices to obtain Internet access.

443. Further, Abhishek describes Wi-Fi bridging functionality, which can be used to extend the range of wireless networks. A POSITA would have been motivated to take teachings of Abhishek to extend the range (or overcome obstructions) of the 802.11 devices taught in Buddhikot and Lord. Further, Abhishek teaches its usage in large-scale Wi-Fi deployments (*e.g.*, ¶5), and a POSITA would have been motivated to provide Buddhikot's MobileHotSpot Gateway and Mobile Nodes in large scale deployments.

444. Furthermore, as described in more detail in [1.A.1], Buddhikot provides a suggestion to include LAN intermediaries (which a POSITA would understand to include 802.11 intermediaries) between its access points and Internet gateways. Buddhikot at 7:11-19. Abhishek teaches 802.11

intermediaries like those suggested by Buddhikot, and a POSITA would have accordingly been motivated to combine them.

2. Element-by-Element Analysis of the Challenged Claims

445. **Claim 32:** Claim 32 recites: “**The system of claim 16 wherein the second computing device is further adapted to provide a third wireless enabled computing device with access to destinations on the IP based network by causing the second computing device to serve the third device as a third AP having a third APID, distinct from the first APID and from the second APID, and provide the third device access to the IP based network via the second AP using the connection between the first computing device and the IP based network through the first AP.**”

446. In claim 32 the third device is connected to the second device (the mobile node) in a linear chain, rather than to the first device (the MobileHotSpot gateway). In other words, this claim describes the second device (the mobile node) acting as a Wi-Fi bridge to connect other devices (the third device).

447. As described above, Abhishek teaches providing access point bridges to provide an Internet connection. Abhishek also states that these access points can be general purpose computers. *See, e.g., ¶¶ 29-30.* Accordingly, it would

have been obvious to provide a mobile node an access point that a third device can connect to.

448. **Claim 33:** Claim 33 recites: “**The system of claim 32 wherein the second computing device restricts destinations on the IP based network accessible by the third device.**”

449. Buddhikot teaches a firewall to protect devices. Buddhikot at 10:55-59, 11:4-8. In my opinion, a POSITA would understand that firewall specifies filtering rules to restrict traffic to or from a group of IP addresses. In my opinion, a POSITA would understand that a firewall would prevent Internet traffic from accessing internal IP addresses of the router. Lord teaches virtually separating connections for each device (Lord at 2:33-39, 3:57-62) and A POSITA would understand that, because these connections are logically separated and have a public IP address, that the mobile terminal hot-spot (the computing device) would treat them as Internet traffic, which is firewalled and restricted.

450. **Claim 54:** Claim 54 recites: “**The method of claim 43 further comprising providing, by the second computing device, a third wireless enabled computing device with access to destinations on the IP based network by causing the second computing device to serve the third device as a third AP having a third APID, distinct from the first APID and from the second APID,**

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

and provide the third device access to the IP based network via the second AP using the connection between the first computing device and the IP based network through the first AP.”

451. This claim is substantively identical to claim 32, and is disclosed for the same reasons.

452. **Claim 55:** Claim 55 recites: **“The method of claim 54 further comprising restricting destinations on the IP based network accessible by the third device.”**

453. This claim is substantively identical to claim 33, and is disclosed for the same reasons.

D. Buddhikot, Lord, and Kim Render Claims 13, 38 and 64 Obvious

1. Motivation to Combine Kim With Buddhikot and Lord

454. Buddhikot, Lord, and Kim are all focused on networking. Buddhikot and Lord describe a hotspot, and Kim describes communication techniques for communicating between network devices across different networks (Kim at Abstract). Accordingly, in my opinion, a POSITA would have been motivated to combine Buddhikot, Lord, and Kim because they are in the same field.

455. Further, Buddhikot teaches using a NAT. Buddhikot at 16:50-53, 24:40, 24:54. Kim teaches functionality of a well-known extension of the NAT—the Network Address Port Translation (NAPT). Kim at ¶¶ 10-14, Fig. 1; *see also* RFC2663, p. 11, §4.1.2. Because the NAPT is an extension of the NAT, a POSITA would have found it obvious to implement an extension of a protocol that Buddhikot teaches using. NAPT is a common protocol well known to a POSITA. It is simply a well-known way to route using a NAT. NAPT is a common protocol well known to a POSITA. It is simply a well-known way to route using a NAT. In my opinion, A POSITA would view Buddhikot's disclosure of a NAT to include a NAPT, or at least a suggestion to use a particular type of NAT function like NAPT. A POSITA would understand that a NAPT is a subset of NAT functionality.

2. Element-by-Element Analysis of the Challenged Claims

456. **Claim 13:** Claim 13 recites: “**The computing device of claim 12 wherein data packets destined for each of the given device and the at least one other device are differentiated using different port numbers.**”

457. Buddhikot teaches using a NAT in the MobileHotSpot Gateway. Claim 13 is simply a recitation of the NAPT. Kim teaches the NAPT functionality—it uses ports to determine which device to route an incoming packet to. Kim at ¶¶ 10-14, Fig. 1. It would have been obvious for a POSITA to combine the NAPT functionality as taught in Kim with Buddhikot’s teachings.

458. **Claim 38:** Claim 38 recites: “**The system of claim 37 wherein data packets destined for each of the second computing device and the at least one other device are differentiated using different port numbers.**”

459. This claim is substantively identical to claim 13, and is disclosed for the same reasons.

460. **Claim 64:** Claim 64 recites: “**The method of claim 63 wherein data packets destined for each of the second computing device and the at least one other device are differentiated using different port numbers.**”

461. This claim is substantively identical to claim 13, and is disclosed for the same reasons.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

VIII. GROUNDS FOR UNPATENTABILITY OF THE '369 PATENT

A. **Buddhikot and Lord Render Claims 1-7 Obvious**

1. **Motivation to Combine Buddhikot With Lord**

462. The motivation to combine Buddhikot with Lord is the same as was discussed above under the '306 Patent.

2. **Element-by-Element Analysis of the Challenged Claims**

463. **Claim 1:** Claim 1 recites: “**A computing device comprising: a communication module adapted to: (1) wirelessly connect said computing device to an IP based network via a first wireless access point (AP) having a first AP Identification (APID); and (2) wirelessly communicate with other wireless enabled computing devices ; a user interface and display adapted to allow a user of said computing device to interact with destinations over the IP based network, through the first wireless AP, using a first public IP address; and an AP module adapted to: (1) provide a given device of the other wireless enabled computing devices with access to the IP based network by causing said computing device to serve the given device as a second AP having a second APID, distinct from the first APID, and provide the given device access to the network via the first AP; and (2) tunnel data traffic from the given device, through said computing device, through the first AP, through**

the IP network, to a proxy server, such that the proxy server acts as a proxy of the given device and the data traffic is secure from said computing device and first AP and the given device operates on the network with a public IP address distinct from the first public IP address.”

464. Claim 1 of the '369 Patent is substantively identical to Claim 1 of the '306 Patent, with only very minor cosmetic wording differences, and is therefore taught by Buddhikot and Lord for the reasons discussed above for Claim 1 of the '306 Patent.

465. **Claim 2:** Claim 2 recites: “**A computing device according to claim 1, wherein the second APID is associated with the proxy server.**”

466. Claim 2 of the '369 Patent is substantively identical to Claim 2 of the '306 Patent, with only very minor cosmetic wording differences, and is therefore taught by Buddhikot and Lord for the reasons discussed above for Claim 2 of the '306 Patent.

467. **Claim 3:** Claim 3 recites: “**A computing device according to claim 2, wherein said AP module tunnels data traffic to the proxy server in response to the given device using the second APID.**”

468. Claim 3 states that tunneling data traffic can occur in response to a given device using the second APID. As explained in '306 Patent limitation

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

[1.C.2], Buddhikot's 802.11 Access Point 1445 (the claimed AP module) tunnels traffic for the connecting mobile nodes (the claimed given device) via the Internet to a tunnel end-point gateway (the proxy server). The mobile nodes themselves connect via the MobileHotSpot Gateway's 802.11 access point which has its own SSID (the claimed second APID), as explained above in '306 Patent limitation [1.C.1].

469. **Claim 4:** Claim 4 recites: “**A computing device according to claim 1, wherein said computing device is a mobile device.**”

470. As explained in '306 Patent limitation [1.B], Buddhikot describes its MobileHotSpot Gateway (the claimed computing device) as mobile. Buddhikot at 23:44-49.

471. **Claim 5:** Claim 5 recites: “**A computing device according to claim 4, wherein said computing device is a cellular phone.**”

472. Lord teaches that the claimed computing device can be an Internet connected cellular phone, among other types of computing devices. *See, e.g.*, Lord at Fig. 1.

473. **Claim 6:** Claim 6 recites: “**A computing device according to claim 4, wherein said computing device is a laptop computer.**”

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

474. Buddhikot discloses a “preferred service scenario” where a “user has a *laptop*/handheld that has both a 3G and an 802.11 interface.” Buddhikot at 4:17-19 (emphasis added). Moreover, as discussed above, Buddhikot teaches its MobileHotSpot Gateway (the claimed computing device) can use “the Linux Operating System” and it was obvious to a POSITA that a computing device running Linux could be a laptop. Buddhikot at 7:42-52. Lord, meanwhile, teaches that “a user may carry a plurality of devices (for example, a Personal Digital Assistant (PDA), *a laptop computer*, a mobile telephone, etc.) that are capable of communicating with each other over a wired or wireless network...It would be desirable to provide two-way connectivity to PDNs such as the Internet for all of the devices connected to the user’s LAN.” Lord at 1:42-51 (emphasis added).

475. **Claim 7:** Claim 7 recites: “**A computing device according to claim 1, wherein said computer device prevents the other wireless enabled computing devices from accessing its inner network.**”

476. Buddhikot teaches a dynamic firewall (*e.g.*, Linux IP Filter architecture) that protects the gateway (the claimed given device) against traffic coming from malicious clients. Buddhikot at 10:55-59, 11:4-8. In my opinion, a POSITA would understand that the Linux IP Filter architecture allows the

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

gateway to specify filtering rules to restrict traffic to or from a group of IP addresses. A POSITA would further understand that a user of the gateway can therefore disallow traffic to the inner network of the gateway by specifying so in the filtering rules. A POSITA would have had found it obvious to specify that a client on the Wi-Fi network is not allowed to access other clients on the Wi-Fi network. Indeed, having security measures in place to prevent other Wi-Fi hotspot connected users from accessing one's laptop would have been the normal and expected operation of a Wi-Fi hot spot. Furthermore, a POSITA would have had found it obvious to specify that a client on the Wi-Fi network is not allowed to access other clients on the Wi-Fi network through MAC address filtering, for example.

477. In addition, Lord discloses virtually separating the physical network connection for each device on its WLAN connected to the Internet (Lord at 2:33-39), which creates separate logical connections with each device (*id.* at 3:57-62). A POSITA would understand that, because these connections are logically separated and have a public IP address, that the mobile terminal would treat them as Internet traffic. A POSITA understands that a NAT and a firewall would prevent Internet traffic from accessing the inner network of the router.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

478. In my opinion, having security measures in place to prevent other Wi-Fi hotspot connected users from accessing one's laptop would have been the normal and expected operation of a Wi-Fi hot spot. A POSITA would have expected security preventing Wi-Fi connected devices from accessing each-other when connecting to a public Wi-Fi.

B. Buddhikot, Lord, and Fajardo Render Claims 8-11 Obvious

1. Motivation to Combine Fajardo with Buddhikot and Lord

479. Buddhikot and Fajardo (as well as the '369 patent) are concerned with handovers for a mobile user roaming between wireless networks. For example, Buddhikot is concerned with the “deployment of architectures that allow users to seamlessly switch between” wireless networks, and as discussed above, Buddhikot discloses security features and procedural aspects like IP address allocation for seamless wireless handovers, as well as linking heterogeneous wireless networks. Buddhikot at 1:24-40. Fajardo, meanwhile, is concerned with data loss during the same wireless handovers. *E.g.*, Fajardo at 5:26-38. The '369 patent itself is concerned with allowing “for fast handovers in wireless networks, such as 802.11 networks.” '369 Patent at 2:3-11. Like Buddhikot and Lord, Fajardo simply wants to permit “a mobile device with an IP-based wireless network interface” to move “from one network into another network.” Fajardo at 4:16-23. As a result, Buddhikot, Lord, and Fajardo have disclosures that are compatible with each other regarding procedures for dealing with heterogeneous wireless networks, like Wi-Fi and cellular. These references' disclosures are therefore compatible with each other, and in the same field of endeavor, and a POSITA would have combined them.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

480. Further, Buddhikot describes data loss as a concern, because it discloses “non-volatile storage device that stores the session state data, and means for transmitting the stored session state data to the client if the client loses a connection with the gateway and resumes the connection with the gateway.” Buddhikot at claim 20. Fajardo expands on this simple disclosure by teaching an expanded data buffering scheme that allows for incoming and outgoing data to be stored and sent even if a device is disconnected. Fajardo at Abstract. In this way, Fajardo’s disclosures complement those of Buddhikot on the same data-loss issue, and one of skill would also have been motivated to combine them. At the time of the ’369 patent’s invention, one of skill looking at Buddhikot’s disclosures on data loss prevention between heterogeneous wireless networks would have been motivated to investigate those data loss prevention aspects discussed in Buddhikot in greater detail, and that motivation and Buddhikot’s teaching would have inevitably led to Fajardo.

481. In my opinion, a POSITA would have been motivated to combine Buddhikot with Fajardo because they both describe using Mobile IP (Fajardo at ¶118; Buddhikot at 6:34-36), 802.11 (Fajardo at ¶12; Buddhikot at 23:46-49), and IPsec (Fajardo at ¶183; Buddhikot at 9:53-58).

2. Element-by-Element Analysis of the Challenged Claims

482. **Claim 8, Preamble:** The Claim 8 preamble recites: “**A computing device comprising:**”

483. [8.Preamble] is identical to '306 Patent's [1.Preamble] and is taught for the same reasons.

484. **Element [8.A]:** Claim 8, Element [8.A] recites: “**a first communication module adapted to communicate over an IP network, using a first public IP address, via a first wireless access point (AP), the first wireless AP having a first AP Identification (APID);**”

485. [8.A] is substantially identical to '306 Patent limitations [1.A, 1.A.1, and 1.A.2] and is taught for the same reasons.

486. **Element [8.B]:** Claim 8, Element [8.B] recites: “**a second communication module adapted to wirelessly communicate, as a second access point (AP) having a second APID, with other wireless enabled computing devices and provide the other wireless enabled computing devices access to the IP network via the first wireless AP, wherein data traffic from the other wireless enabled computing devices is tunneled by the second AP through the first AP to a proxy server such that the proxy server acts as a proxy of the other wireless enabled computing devices and the data traffic is**

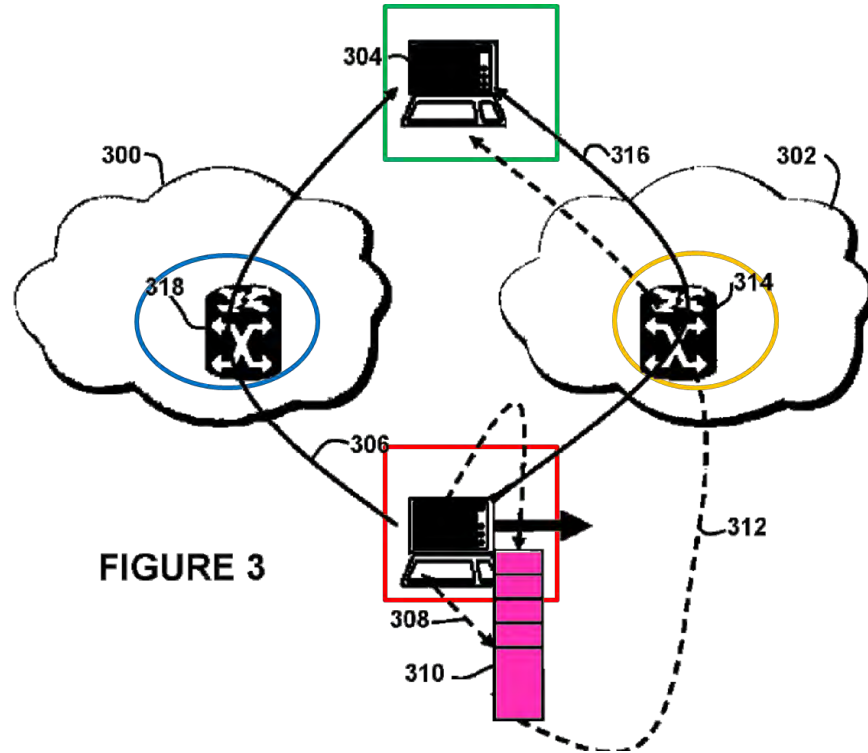
secure from the first and second APs and the other wireless enabled computing devices operate on the IP network with a public IP address distinct from the first public IP address;”

487. [8.B] is substantially identical to ’306 Patent limitations [1.C.1, 1.C.2, and 1.C.3] and is taught for the same reasons.

488. **Element [8.C]:** Claim 8, Element [8.C] recites: **“data storage adapted to store data, addressed to a destination on the IP network, received wirelessly via said second communication module, from a given device of the other wireless enabled computing devices;”**

489. Both Buddhikot and Fajardo teach a data storage adapted to store data. Fajardo at Abstract, Figs. 1-3, ¶¶31, 49, 82, 85-87 (teaching a buffer node); Buddhikot at claim 20 (teaching non-volatile storage).

490. Fajardo teaches that the buffer node buffers traffic (stores data) from a mobile node (given device) intended for a correspondent node on the Internet (a destination on the IP network). As discussed above, and as illustrated below, Fajardo teaches a buffer node 310 (pink), that intercepts traffic between from the mobile node (red) to the correspondent node 304 (green). Fajardo at ¶99; *see also id.* at Figs. 2-3, ¶¶97-103. Buffering occurs when handoff is occurring. *See, e.g.,* Fajardo at ¶39.



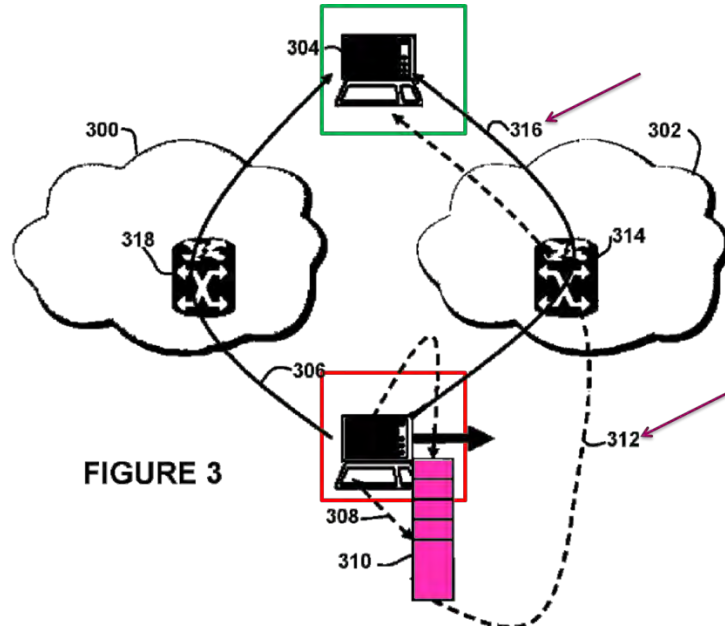
491. Fajardo teaches that the buffer node can be placed at the first access router 318 (blue, above). Fajardo at ¶103; *see also id.* at ¶¶51, 80, 97. The connection between the mobile node and access routers is a wireless connection (*id.* at 10-18), including 802.11 (*id.* at ¶20).

492. As described above, a POSITA would have been motivated to provide buffering services as taught in Fajardo with Buddhikot's teachings to improve hand-off and decrease data loss. Accordingly, a POSITA would have been motivated to provide Fajardo's teachings of a buffer node in Buddhikot's teachings of a MobileHotSpot Gateway.

493. In my opinion, it would have been obvious to include a buffering node with a MobileHotSpot gateway that has a buffering node that intercepts data from the mobile node destined to an Internet destination.

494. **Element [8.D]**: Claim 8, Element [8.D] recites: “**transmission logic adapted to transmit the stored data to the destination, over the IP network, after communications between said computing device and the given device are disconnected, such that data may be uploaded from a client device to said computing device and subsequently uploaded by said computing device to a destination on the internet.**”

495. Fajardo teaches transmitting buffered data to the correspondent node over the Internet. As previously described, prior to handoff the buffer node buffers traffic between the mobile node and the correspondent node 304. Fajardo at ¶99 (describing flow line 312), Fig. 3 Element 312 (annotated with arrow, below). After handoff, the traffic continues to the correspondent node. Fajardo at ¶99 (describing flow line 316), Fig. 3 Element 316 (annotated with arrow, below). After handoff the packets that are buffered are forwarded to the destination address of each packet. Fajardo at ¶102; *see also id.* at ¶¶93-96.



496. Fajardo teaches transmitting the buffered data after the given device (the mobile node) disconnects from the access point (the claimed said computing device, which as described above is the MobileHotSpot gateway taught in Buddhikot). As taught in Fajardo, buffered packets are forwarded “when the handoff process is completed” (Fajardo at ¶28; *see also id.* at ¶31), which a POSITA understands to be when the mobile node disconnects from one access point to its new access point. The forwarding process begins when the mobile node completes its handoff and signals to the buffer node to begin forwarding. Fajardo at ¶32.

497. The data is uploaded from the mobile node (the client device) to the first access point (the claimed said computing device, which as described above is the MobileHotSpot gateway taught in Buddhikot), and subsequently uploaded by

the first access point to the correspondent node (the destination point). As described above, the buffer node can be at the first access point. Fajardo at ¶103; *see also id.* at ¶¶51, 80, 97. As also described above, after handoff, buffered packets are forwarded to their intended destination. Fajardo at ¶102; *see also id.* at ¶¶93-96.

498. **Claim 9:** Claim 9 recites: “**A computing device according to claim 8, wherein said computing device is a mobile device.**”

499. This claim is substantively identical to claim 4, and is taught for the same reasons.

500. **Claim 10:** Claim 10 recites: “**A computing device according to claim 9, wherein said computing device is a cellular phone.**”

501. This claim is substantively identical to claim 5, and is taught for the same reasons.

502. **Claim 11:** Claim 11 recites: “**A computing device according to claim 9, wherein said computing device is a laptop computer.**”

503. This claim is substantively identical to claim 6, and is taught for the same reasons.

C. Buddhikot, Lord, Fajardo, and Aarnio Render Claim 12 Obvious

1. Motivation to Combine Aarnio with Buddhikot, Lord, and Fajardo

504. Additional motivations to combine Buddhikot, Lord, and Fajardo are provided above.

505. In my opinion, a POSITA would have been further motivated to combine Buddhikot, Lord, Fajardo, and Aarnio because they are all in similar fields. Aarnio describes Wi-Fi capable (Aarnio at 7:56-62) and 3G capable (*id.* at 8:53) mobile terminal devices. *See also id.* at Fig. 5, 6:9-11. Buddhikot and Lord likewise teach the same. *See* Buddhikot at Fig. 13 and Lord at Fig. 1

506. In my opinion, a POSITA would have been further motivated to combine Aarnio with Buddhikot, Lord, and Fajardo because they are complimentary. Buddhikot describes an infrastructure, along with security elements. Buddhikot at 9:43-56, Fig. 15, 10:3-31, 10:55-67, Figs. 2 and 14. Lord provides particular infrastructure related teachings regarding how to use DHCP to allocate public IPs. Lord at 3:43-4:13, 5-5-13. Buddhikot and Fajardo discuss handoffs, and seek to improve efficiencies in mobile handoff. Buddhikot at 1:24-40, Fajardo at 5:26-38. Fajardo teaches providing a buffer to store data during a handoff period. Fajardo at Abstract, Figs. 1-3, ¶¶31, 49, 82, 85-87. Buddhikot, Lord, and Fajardo teach an infrastructure, and Aarino describes an application to

use on a wireless infrastructure. A POSITA would have been motivated to look at applications (like those taught in Aarnio) that can be used on the network infrastructure taught by Buddhikot, Lord, and Fajardo, and vice versa. Indeed, a POSITA would be motivated to provide both a user-useful application and an infrastructure.

507. In my opinion, a POSITA would have been further motivated to combine Fajardo and Aarnio because they both teach an intermediary to store data. Fajardo teaches providing an intermediary with instructions to store data and then upload that stored data to an external server. Aarnio likewise teaches an intermediary that stores data and receives an upload signal to upload data.

508. In my opinion, a POSITA would have been further motivated to combine Aarnio with Fajardo because Aarnio describes the desire to transmit on higher latency networks and Fajardo is directed to instances when a mobile terminal is going through hand-off—a state that potentially has a considerable amount of packet loss.

2. Element-by-Element Analysis of the Challenged Claims

509. **Claim 12:** Claim 12 recites: “**A computing device according to claim 8, wherein the computing device is further adapted to send to the given**

device, over the IP network, a confirmation once the data is completely transmitted to the destination.”

510. As explained above, it is my opinion that the Buddhikot, Lord, and Fajardo teach claim 8. The MobileHotSpot Gateway of Buddhikot teaches the claimed computing device, and it would have been obvious to include Fajardo’s teaching of a buffering in the MobileHotSpot Gateway teaching of Buddhikot. As also discussed in [8.D], Fajardo further teaches that, upon a signal from the mobile node, that the buffering node transmits data to the destination over the Internet.

511. Aarnio teaches a wireless command from a mobile terminal 10 that instructs an intermediary server (a source 102) to transmit content to a destination server (destination 104), upon. *See, e.g.*, Aarnio at 16:52-63; *see also id.* Fig. 8A. Upon completion of that transmission, the source agent 108 at the source 102 “send[s] a transaction complete notification to the terminal to thereby notify the terminal that the source agent [at the source server] has completed the content upload to the destination.” Aarnio at 17:57-65; *see also* Fig. 6, 8A. Aarnio teaches that “the content can be uploaded to the destination across the Internet.” Aarnio at 17:50-52.

512. In my opinion, a POSITA would have been motivated to include Aarnio’s intermediary source upload confirmation because Aarnio teaches an

improvement to Fajardo. Aarnio's intermediary provides an upload confirmation notice after it uploads its data to an external server. In many instances, a POSITA would be concerned with data-loss during handover, and would need to know when certain data had not been properly sent. A POSITA would understand that this would be necessary in particular applications where the transmission of data is important, from activities such as providing reliable financial transactions to the personal (*e.g.*, ensuring uploaded media are not missing data that would degrade their quality or otherwise corrupt them). Absent a confirmation, a POSITA would understand that the data would need to be re-sent, or that other software related remedial measures would need to occur.

D. Claim 13 is Obvious Over Buddhikot

1. Element-by-Element Analysis of the Challenged Claims

513. **Claim 13, Preamble:** The Claim 13 preamble recites:

“Communication circuitry adapted to:”

514. As discussed above for Claim 1 of the '306 Patent, Buddhikot is replete with teachings providing communication circuitry within a gateway. Buddhikot at 23:44-24:18; Figs. 12-14.

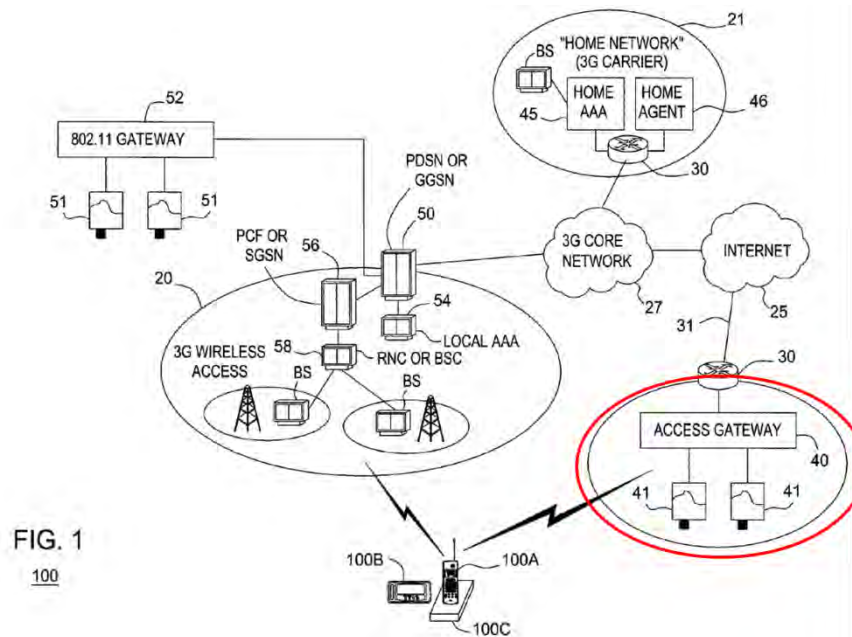
515. **Element [13.A]**: Claim 13, Element [13.A] recites: “**(1) generate a second access point identification (APID) associated with an access point (AP) having a first APID;**”

516. This limitation is substantially similar to '306 Patent limitation [1.C] in its disclosure of an 802.11 access point with an access point identification, which are taught by Buddhikot as discussed above under '306 Patent limitation [1.C]. Buddhikot at 23:45-50; 17:55-61. The difference from '306 Patent limitation [1.C] is that this limitation is concerned with creating a *second* APID from the same device, and this functionality is also disclosed in Buddhikot, as described below.

517. Buddhikot teaches that the communication circuitry generates a first and second APID (two SSIDs) associated with an access point (the gateway). Buddhikot teaches that “[e]ach gateway system 40 preferably serves multiple 802.11 access points 41 in a hot-spot.” Buddhikot at 7:11-19. The 802.11 access points can be directly connected to the gateway. Buddhikot at 7:11, Fig. 1. As show in Fig. 1, the gateway the (access point) has at least two 802.11 access points (annotated in red, below). As explained in '306 Patent [1.A.1], it would be obvious to a POSITA that an 802.11 access point has an SSID. EX1033, U.S.

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

Patent No. 6,950,628 at 4:13-16; *see also* Buddhikot at 17:16-62 (stating that the client would need 802.11 ESSIDs to connect to the 802.11 access point).



518. **Element [13.B]:** Claim 13, Element [13.B] recites: “(2) provide a tunnel for wireless devices connecting to said AP using the second APID.”

519. This limitation is substantively the same as the secure tunneling portion of '306 Patent limitation [1.C.2], and is taught by Buddhikot for the same reason, because it teaches end-to-end IPsec/VPN tunneling from the Wi-Fi enabled client device connected to the gateway. *See also* Buddhikot at 9:53-58, 17:44-50, 18:66-19:3, Fig. 15 (showing the client with an IPsec driver and control, and also Wi-Fi to connect to the hotspot). It would have been obvious to

a POSITA that a Wi-Fi enabled device can connect to a Wi-Fi network using an SSID (*i.e.*, to the AP using the second APID).

E. Claim 13 is Anticipated By Vucina

520. Buddhikot teaches providing a gateway as the claimed access point, and renders claim 13 obvious. If the Board construes the access point to be restricted to an 802.11 access point, this feature is also taught in the art. Indeed, providing multiple SSIDs for a single 802.11 access point is a very well-known technique in the art referred to as providing a “virtual access point.” *See also APP. AA*, U.S. Pub. No. 2007/0206527 at Abstract; **APP. AB**, U.S. Pub. 2004/0214572 at ¶41; **APP. AC**, U.S. Patent No. 7,860,978 at 9:54-61.

521. Claim 13 is very simple and broad, and anticipated by Vucina as discussed below.

1. Element-by-Element Analysis of the Challenged Claims

522. **Claim 13, Preamble:** The Claim 13 preamble recites:
“Communication circuitry adapted to:”

523. Vucina teaches communication circuitry. As shown below, Vucina discloses an 802.11 (Vucina at ¶21) wireless access point 120 (*id.* at ¶62, ¶91).

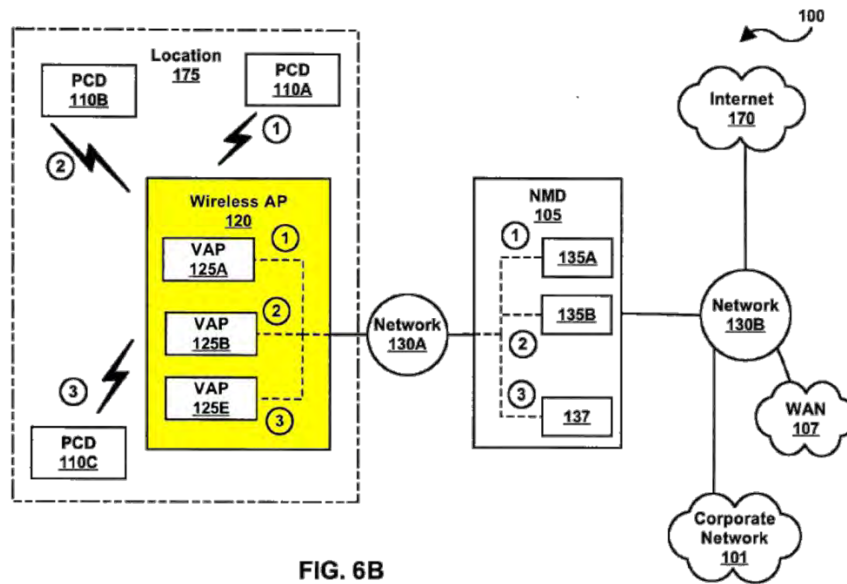
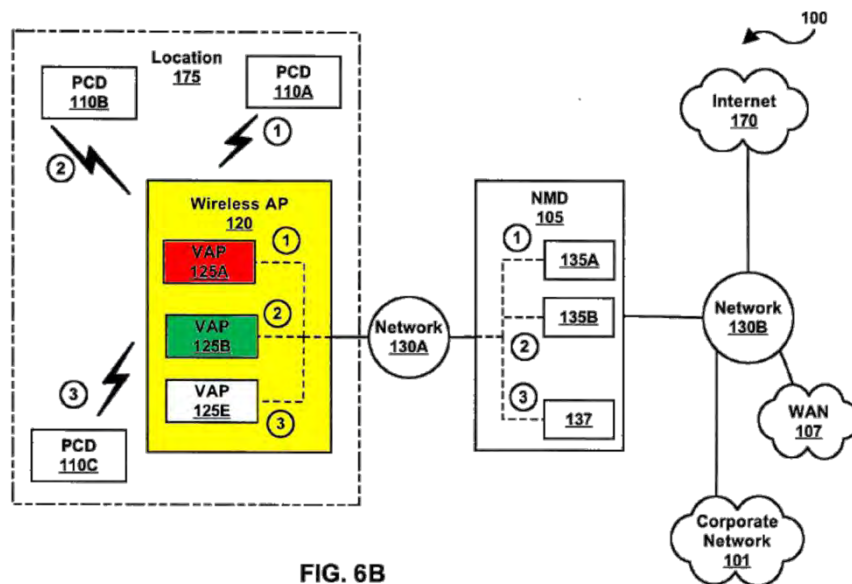


FIG. 6B

524. **Element [13.A]:** Claim 13, Element [13.A] recites: “(1) generate a second access point identification (APID) associated with an access point (AP) having a first APID;”

525. Vucina teaches that the wireless access point generates two APIDs (e.g., VAP125A and 125B and their SSIDs). For example, the wireless access point can generate a virtual access point 125A (red, below), having an SSID titled “Foobazco.” Vucina at ¶180 and Fig. 6B; *see also id.* at ¶170. A virtual access point 125B (green) having a different SSID called “Wellcent” can also be generated by the wireless AP. *Id.* at ¶182 at Fig. 6B. Accordingly, Vucina teaches this limitation.



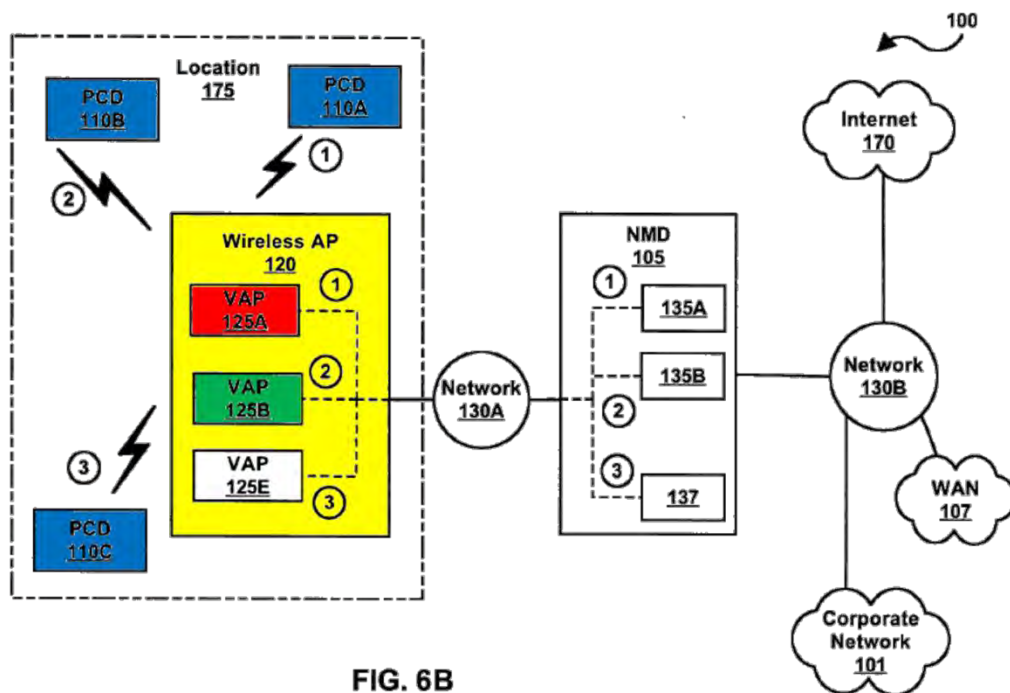
526. A virtual access point is a term well-known in the art. EX1003 at ¶526. Vucina teaches what a virtual access point is—“a wireless AP120 may be operable to appear as one of a plurality of different location wireless Aps, meaning that a single wireless AP may ‘pretend to be’ or behave as an access point dedicated to a particular location and/or carriers for each of a plurality of different locations and/or carriers. In other words . . . wireless AP 120 may execute one or more software programs that allow it to act as a *wireless AP or virtual AP (VAP) 125 for each of the plurality of locations and/or carriers.*”

527. **Element [13.B]:** Claim 13, Element [13.B] recites: “(2) provide a tunnel for wireless devices connecting to said AP using the second APID.”

528. Vucina teaches that the wireless access point provides a tunnel for wireless devices connecting to the second APID. Vucina provides providing a

Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

device (e.g., PCD110C) services including *inter alia* tunneling and VPN tunneling. Vucina at ¶¶81, 105-106. Other tunneling schemes taught include: “IPSec, GRE, PPTP, IP over IP, or other tunnel[s]” or combinations of tunnels. *Id.* at ¶106. For example, Vucina teaches that its virtual access points 125 provide tunnels from the wireless access point to the network management device 105 over the network 130. *Id.* at ¶177; see also Figs. 5A-B. A POSITA would understand that these aforementioned tunnels would be provided to any Wi-Fi network devices (e.g., PCD 110A-C (blue, below) via any VAP 125A-B, including the claimed second APID. .



Declaration of Tal Lavian, Ph.D. in Support of
Petitions for *Inter Partes* Review of
U.S. Patent Nos. 8,559,369 and 9,042,306

IX. CONCLUSION

529. It is my conclusion that Claims 1-68 of the '306 Patent and Claims 1-13 of the '369 Patent are unpatentable as obvious and/or anticipated over the prior art, as discussed in detail above.

530. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in this proceeding. If required, I will appear for cross-examination at the appropriate time. I reserve the right to offer opinions relevant to the invalidity of the Challenged Patent claims at issue and/or offer testimony in support of this Declaration.

531. I hereby declare under penalty of perjury that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Executed:

3/19/2017

Date

Tal Lavian
3/19/2017

Tal Lavian, Ph.D.
Sunnyvale, California

Tal Lavian, Ph.D.



<http://telecommnet.com>
<http://cs.berkeley.edu/~tlavian>
tlavian@telecommnet.com



1640 Mariani Dr.
 Sunnyvale, CA 94087
 (408)-209-9112

Research and Consulting: Telecommunications, Network Communications, and Mobile Wireless Technologies

Scientist, educator, and technologist with over 25 years of experience; co-author on over 25 scientific publications, journal articles, and peer-reviewed papers; named inventor on over 100 issued and filed patents; industry fellow and lecturer at UC Berkeley Engineering–Center for Entrepreneurship and Technology (CET)

EDUCATION

- **Ph.D.**, Computer Science specializing in networking and communications, UC Berkeley
- **M.Sc.**, Electrical Engineering, Tel Aviv University
- **B.Sc.**, Mathematics and Computer Science, Tel Aviv University

EXPERTISE

Network communications, telecommunications, Internet protocols, and mobile wireless:

- **Communication networks:** Internet protocols; TCP/IP suite; TCP; UDP; IP; VoIP; Ethernet; network protocols; network software applications; data link, network, and transport layers (L2, L3, L4)
- **Internet software:** Internet software applications; distributed computing; cloud computing; Web applications; FTP; HTTP; Java; client server; file transfer; multicast; streaming media
- **Routing/switching:** LAN; WAN; VPN; routing protocols; RIP; BGP; MPLS; OSPF; IS-IS; DNS; QoS; switching; packet switching; network infrastructure; network communication architectures
- **Mobile wireless:** wireless LAN; 802.11; cellular systems; mobile devices; smartphone technologies

ACCOMPLISHMENTS

- Selected as principal investigator for three US Department of Defense (DARPA) projects
- Directed research project on networking computation for the US Air Force Research Lab (AFRL)
- Led and developed the first network resourcescheduling service for grid computing
- Administered wireless research project for an undisclosed US federal agency
- Managed and engineered the first demonstrated transatlantic dynamic allocation of 10Gbps Lambdas as a grid service
- Spearheaded the development of the first demonstrated wire-speed active network on commercial hardware
- Invented over 100 patents; over 50 prosecuted *pro se* in front of the USPTO
- Created and chaired Nortel Networks' EDN Patent Committee

PROFESSIONAL EXPERIENCE

University of California, Berkeley, Berkeley, California 2000-Present
Berkeley Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, Nortel's Scientist Liaison

Some positions and projects were concurrent, others sequential

- Serves as an industry fellow and lecturer at the Center for Entrepreneurship and Technology (CET).
- Studied network services, telecommunication systems and software, communications infrastructure, and data centers
- Developed long-term technology for the enterprise market, integrating communication and computing technologies
- Conducted research projects in data centers (RAD Labs), telecommunication infrastructure (SAHARA), and wireless systems (ICEBERG)
- Acted as scientific liaison between Nortel Research Lab and UC Berkeley, providing tangible value in advanced technologies
- Earned a Ph.D. in Computer Science with a specialization in communications and networking

TelecommNet Consulting, Inc. (Innovations-IP) Sunnyvale, California 2006-Present
Principal Scientist

- Consults in the areas of network communications, telecommunications, Internet protocols, and smartphone mobile wireless devices
- Provides architecture and system consultation for projects relating to computer networks, mobile wireless devices, and Internet web technologies
- Acts as an expert witness in network communications patent infringement lawsuits

VisuMenu, Inc., Sunnyvale, California 2010-Present
Co- Founder and Chief Technology Officer (CTO)

- Designs and develops architecture and system of visual IVR technologies for smartphones and wireless mobile devices in the area of network communications
- Designs crawler/spider system for IVR / PBX using Asterisk, SIP, and VoIP
- Deploys the system as cloud networking and cloud computing utilizing Amazon Web Services

Ixia, Santa Clara, California 2008 - 2008
Communications Consultant

- Researched and developed advanced network communications testing technologies:
 - IxNetwork/IxN2X — tested IP routing and switching devices and broadband access equipment. Provided traffic generation and emulation for the full range of protocols: routing, MPLS, layer 2/3 VPNs, carrier Ethernet, broadband access, and data center bridging
 - IxLoad — quickly and accurately modeled high-volume video, data, and voice subscribers and servers to test real-world performance of multiservice delivery and security platforms
 - IxCatapult — emulated a broad range of wireless access and core protocols to test wireless components and systems that, when combined with IxLoad, provides an end-to-end solution for testing wireless service quality
 - IxVeriWave — employed a client-centric model to test Wi-Fi and wireless LAN networks by generating repeatable large-scale, real-world test scenarios that are virtually impossible to create by any other means

- Test automation — provided simple, comprehensive lab automation to help test engineering teams create, organize, catalog, and schedule execution of tests

Nortel Networks, Santa Clara, California

1996 - 2007

Originally employed by Bay Networks, which was acquired by Nortel Networks

Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer

- Held scientific and research roles at Nortel Labs, Bay Architecture Labs, and in the office of the CTO

Principal Investigator for US Department of Defense (DARPA) Projects

- Conceived, proposed, and completed three research projects: active networks, DWDM-RAM, and a networking computation project for Air Force Research Lab (AFRL)
- Led a wireless research project for an undisclosed US federal agency

Academic and Industrial Researcher

- Analyzed new technologies to reduce risks associated with R&D investment
- Spearheaded research collaboration with leading universities and professors at UC Berkeley, Northwestern University, University of Amsterdam, and University of Technology, Sydney
- Evaluated competitive products relative to Nortel's products and technology
- Proactively identified prospective business ideas, which led to new networking products
- Predicted technological trends through researching the technological horizon and academic sphere
- Designed software for switches, routers, and network communications devices
- Developed systems and architectures for switches, routers, and network management
- Researched and developed the following projects:

▪ Data-Center Communications: network and server orchestration	2006-2007
▪ DRAC: SOA-facilitated L1/L2/L3 network dynamic controller	2003-2007
▪ Omega: classified wireless project for undisclosed US Federal Agency	2006-2006
▪ Open platform: project for the US Air Force Research Laboratory (AFRL)	2005-2005
▪ Network resource orchestration for Web services workflows	2004-2005
▪ Proxy study between Web/grids services and network services	2004-2004
▪ Streaming content replication: real-time A/V media multicast at edge	2003-2004
▪ DWDM-RAM: US DARPA-funded program on agile optical transport	2003-2004
▪ Packet capturing and forwarding service on IP and Ethernet traffic	2002-2003
▪ CO2: content-aware agile networking	2001-2003
▪ Active networks: US DARPA-funded research program	1999-2002
▪ ORE: programmable network service platform	1998-2002
▪ JVM platform: Java on network devices	1998-2001
▪ Web-based device management: network device management	1996-1997

Technology Innovator and Patent Leader

- Created and chaired Nortel Networks' EDN Patent Committee
- Facilitated continuous stream of innovative ideas and their conversion into intellectual property rights
- Developed intellectual property assets through invention and analysis of existing technology portfolios

Aptel Communications, Netanya, Israel

1994-1995

Software Engineer, Team Leader

Start-up company focused on mobile wireless CDMA spread spectrum PCN/PCS

- Developed a mobile wireless device using an unlicensed band [Direct Sequence Spread Spectrum (DSSS)]
- Designed and managed a personal communication network (PCN) and personal communication system (PCS), which are the precursors of short text messages (SMS)
- Designed and developed network communications software products (mainly in C/C++)
- Brought a two-way paging product from concept to development

Scitex Ltd., Herzeliya, Israel

1990-1993

Software Engineer, Team Leader

Software and hardware company acquired by Hewlett Packard (HP)

- Developed system and network communications (mainly in C/C++)
- Invented Parallel SIMD Architecture
- Participated in the Technology Innovation group

Shalev, Ramat-HaSharon, Israel

1987-1990

Start-up company

Software Engineer

- Developed real-time software and algorithms (mainly in C/C++ and Pascal)

PROFESSIONAL ASSOCIATIONS

- IEEE senior member
- IEEE CNSV co-chair, Intellectual Property SIG (2013)
- President Next Step Toastmasters (an advanced TM club in the Silicon Valley) (2013-2014)
- Technical co-chair, IEEE Hot Interconnects 2005 at Stanford University
- Member, IEEE Communications Society (COMMSOC)
- Member, IEEE Computer Society
- Member, IEEE Systems, Man, and Cybernetics Society
- Member, IEEE-USA Intellectual Property Committee
- Member, ACM, ACM Special Interest Group on Data Communication (SIGCOM)
- Member, ACM Special Interest Group on Hypertext, Hypermedia, and Web (SIGWEB)
- Member, IEEE Consultants' Network (CNSV)
- Global Member, Internet Society (ISOC)
- President Java Users Group – Silicon Valley Mountain View, CA, 1999-2000
- Toastmasters International

ADVISORY BOARDS

- Quixey – search engine for wireless mobile apps
- Mytopia – mobile social games
- iLeverage – Israeli Innovations

PROFESSIONAL AWARDS

- Top Talent Award – Nortel
- Top Inventors Award – Nortel EDN
- Certified IEEE-WCET - Wireless Communications Engineering Technologies
- Toastmasters International - Competent Communicator (twice)
- Toastmasters International - Advanced Communicator Bronze

Patents and Publications

*(Not an exhaustive list)***Patents Issued**

US 9,184,989	Grid proxy architecture for network resources	Link
US 9,083,728	Systems and methods to support sharing and exchanging in a network	Link
US 9,021,130	Photonic line sharing for high-speed routers	Link
US 9,001,819	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,949,846	Time-value curves to provide dynamic QoS for time sensitive file transfers	Link
US 8,929,517	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,903,073	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,898,274	Grid proxy architecture for network resources	Link
US 8,880,120	Device and method for providing enhanced telephony	Link
US 8,879,703	System method and device for providing tailored services when call is on-hold	Link
US 8,879,698	Device and method for providing enhanced telephony	Link
US 8,867,708	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,787,536	Systems and methods for communicating with an interactive voice response system	Link
US 8,782,230	Method and apparatus for using a command design pattern to access and configure network elements	Link
US 8,762,963	Translation of programming code	Link
US 8,762,962	Methods and apparatus for automatic translation of a computer program language code	Link
US 8,745,573	Platform-independent application development framework	Link
US 8,731,148	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,688,796	Rating system for determining whether to accept or reject objection raised by user in social network	Link
US 8,619,793	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link
US 8,572,303	Portable universal communication device	Link
US 8,553,859	Device and method for providing enhanced telephony	Link

APPENDIX A

US 8,548,131	Systems and methods for communicating with an interactive voice response system	Link
US 8,537,989	Device and method for providing enhanced telephony	Link
US 8,341,257	Grid proxy architecture for network resources	Link
US 8,161,139	Method and apparatus for intelligent management of a network element	Link
US 8,146,090	Time-value curves to provide dynamic QoS for time sensitive file transfer	Link
US 8,078,708	Grid proxy architecture for network resources	Link
US 7,944,827	Content-aware dynamic network resource allocation	Link
US 7,860,999	Distributed computation in network devices	Link
US 7,734,748	Method and apparatus for intelligent management of a network element	Link
US 7,710,871	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link
US 7,580,349	Content-aware dynamic network resource allocation	Link
US 7,433,941	Method and apparatus for accessing network information on a network device	Link
US 7,359,993	Method and apparatus for interfacing external resources with a network element	Link
US 7,313,608	Method and apparatus for using documents written in a markup language to access and configure network elements	Link
US 7,260,621	Object-oriented network management interface	Link
US 7,237,012	Method and apparatus for classifying Java remote method invocation transport traffic	Link
US 7,127,526	Method and apparatus for dynamically loading and managing software services on a network device	Link
US 7,047,536	Method and apparatus for classifying remote procedure call transport traffic	Link
US 7,039,724	Programmable command-line interface API for managing operation of a network device	Link
US 6,976,054	Method and system for accessing low-level resources in a network device	Link
US 6,970,943	Routing architecture including a compute plane configured for high-speed processing of packets to provide application layer support	Link
US 6,950,932	Security association mediator for Java-enabled devices	Link
US 6,850,989	Method and apparatus for automatically configuring a network switch	Link

APPENDIX A

US 6,845,397	Interface method and system for accessing inner layers of a network protocol	Link
US 6,842,781	Download and processing of a network management application on a network device	Link
US 6,772,205	Executing applications on a target network device using a proxy network device	Link
US 6,564,325	Method of and apparatus for providing multi-level security access to system	Link
US 6,175,868	Method and apparatus for automatically configuring a network switch	Link
US 6,170,015	Network apparatus with Java co-processor	Link
US 8,687,777	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,681,951	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,625,756	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,594,280	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,548,135	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,406,388	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,345,835	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,223,931	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,160,215	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,155,280	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,054,952	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,000,454	Systems and methods for visual presentation and selection of IVR menu	Link
EP 1,905,211	Technique for authenticating network users	Link
EP 1,142,213	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link
EP 1,671,460	Method and apparatus for scheduling resources on a switched underlay network	Link
CA 2,358,525	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link

Patent Applications Published and Pending*(Not an exhaustive list)*

US 20150058490	Grid Proxy Architecture for Network Resources	Link
US 20150010136	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20140379784	Method and Apparatus for Using a Command Design Pattern to Access and Configure Network Elements	Link
US 20140105025	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20140105012	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20140012991	Grid Proxy Architecture for Network Resources	Link
US 20130080898	Systems and Methods for Electronic Communications	Link
US 20130022191	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20130022183	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20130022181	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20120180059	Time-Value Curves to Provide Dynamic QOS for Time Sensitive File Transfers	Link
US 20120063574	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20110225330	Portable Universal Communication Device	Link
US 20100220616	Optimizing Network Connections	Link
US 20100217854	Method and Apparatus for Intelligent Management of a Network Element	Link
US 20100146492	Translation of Programming Code	Link
US 20100146112	Efficient Communication Techniques	Link
US 20100146111	Efficient Communication in a Network	Link
US 20090313613	Methods and Apparatus for Automatic Translation of a Computer Program Language Code	Link
US 20090313004	Platform-Independent Application Development Framework	Link
US 20090279562	Content-aware dynamic network resource allocation	Link
US 20080040630	Time-Value Curves to Provide Dynamic QoS for Time Sensitive File	Link

Transfers

US 20070169171	Technique for authenticating network users	Link
US 20060123481	Method and apparatus for network immunization	Link
US 20060075042	Extensible Resource Messaging Between User Applications and Network Elements in a Communication Network	Link
US 20050083960	Method and Apparatus for Transporting Parcels of Data Using Network Elements with Network Element Storage	Link
US 20050076339	Method and Apparatus for Automated Negotiation for Resources on a Switched Underlay Network	Link
US 20050076336	Method and Apparatus for Scheduling Resources on a Switched Underlay Network	Link
US 20050076173	Method And Apparatus for Preconditioning Data to Be Transferred on a Switched Underlay Network	Link
US 20050076099	Method and Apparatus for Live Streaming Media Replication in a Communication Network	Link
US 20050074529	Method and apparatus for transporting visualization information on a switched underlay network	Link
US 20040076161	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20020021701	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
WO 2006/063052	Method and apparatus for network immunization	Link
WO 2007/008976	Technique for authenticating network users	Link
WO2000/0054460	Method and apparatus for accessing network information on a network device	Link
US 20140156556	Time-variant rating system and method thereof	Link
US 20140156758	Reliable rating system and method thereof	Link

Publications

(Not an exhaustive list)

- “R&D Models for Advanced Development & Corporate Research” Understanding Six Models of Advanced R&D - Ikhtlaq Sidhu, Tal Lavian, Victoria Howell - University of California, Berkeley. Accepted paper for 2015 ASEE Annual Conference and Exposition- June 2015
- “Communications Architecture in Support of Grid Computing”, Tal Lavian, Scholar's Press 2013 ISBN 978-3-639-51098-0.
- [“Applications Drive Secure Lightpath Creation across Heterogeneous Domains](#), Feature Topic Optical Control Planes for Grid Networks: Opportunities, Challenges and the Vision.” Gommans L.; Van Oudenaarde B.; Dijkstra F.; De Laat C.; Lavian T.; Monga I.; Taal A.; Travostino F.; Wan A.; IEEE Communications Magazine, vol. 44, no. 3, March 2006, pp. 100-106.
- [Lambda Data Grid: Communications Architecture in Support of Grid Computing](#). Tal I. Lavian, Randy H. Katz; Doctoral Thesis, University of California at Berkeley. January 2006.
- “Information Switching Networks.” Hoang D.B.; T. Lavian; The 4th Workshop on the Internet, Telecommunications and Signal Processing, WITSP2005, December 19-21, 2005, Sunshine Coast, Australia.
- [“Impact of Grid Computing on Network Operators and HW Vendors.](#)” Allcock B.; Arnaud B.; Lavian T.; Papadopoulos P.B.; Hasan M.Z.; Kaplow W.; IEEE Hot Interconnects at Stanford University 2005, pp.89-90.
- [DWDM-RAM: A Data Intensive Grid Service Architecture Enabled by Dynamic Optical Networks](#). Lavian T.; Mambretti J.; Cutrell D.; Cohen H.J.; Merrill S.; Durairaj R.; Daspit P.; Monga I.; Naiksatam S.; Figueira S.; Gutierrez D.; Hoang D.B., Travostino F.; CCGRID 2004, pp. 762-764.
- [DWDM-RAM: An Architecture for Data Intensive Service Enabled by Next Generation Dynamic Optical Networks](#). Hoang D.B.; Cohen H.; Cutrell D.; Figueira S.; Lavian T.; Mambretti J.; Monga I.; Naiksatam S.; Travostino F.; Proceedings IEEE Globecom 2004, Workshop on High-Performance Global Grid Networks, Houston, 29 Nov. to 3 Dec. 2004, pp.400-409.
- [Implementation of a Quality of Service Feedback Control Loop on Programmable Routers](#). Nguyen C.; Hoang D.B.; Zhao, I.L.; Lavian, T.; Proceedings, 12th IEEE International Conference on Networks 2004. (ICON 2004) Singapore, Volume 2, 16-19 Nov. 2004, pp.578-582.
- [A Platform for Large-Scale Grid Data Service on Dynamic High-Performance Networks](#). Lavian T.; Hoang D.B.; Mambretti J.; Figueira S.; Naiksatam S.; Kaushil N.; Monga I.; Durairaj R.; Cutrell D.; Merrill S.; Cohen H.; Daspit P.; Travostino F.; GridNets 2004, San Jose, CA., October 2004.
- [DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks](#). Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Daspit, P.; Gutierrez D.; Hoang D. B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F.; Proceedings, 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004, pp. 707-714.
- [DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks](#). Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Gutierrez D.; Hoang D.B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F.; 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004.
- [An Extensible, Programmable, Commercial-Grade Platform for Internet Service Architecture](#). Lavian T.; Hoang D.B.; Travostino F.; Wang P.Y.; Subramanian S.; Monga I.; IEEE Transactions on Systems, Man, and Cybernetics on Technologies Promoting Computational

Intelligence, Openness and Programmability in Networks and Internet Services Volume 34, Issue 1, Feb. 2004, pp.58-68.

- [*DWDM-RAM: An Architecture for Data Intensive Service Enabled by Next Generation Dynamic Optical Networks*](#). Lavian T.; Cutrell D.; Mambretti J.; Weinberger J.; Gutierrez D.; Naiksatam S.; Figueira S.; Hoang D. B.; Supercomputing Conference, SC2003 Igniting Innovation, Phoenix, November 2003.
- [*Edge Device Multi-Unicasting for Video Streaming*](#). Lavian T.; Wang P.; Durairaj R.; Hoang D.; Travostino F.; Telecommunications, 2003. ICT 2003. 10th International Conference on Telecommunications, Tahiti, Volume 2, 23 Feb.-1 March, 2003 pp. 1441-1447.
- [*The SAHARA Model for Service Composition Across Multiple Providers*](#). Raman B.; Agarwal S.; Chen Y.; Caesar M.; Cui W.; Lai K.; Lavian T.; Machiraju S.; Mao Z. M.; Porter G.; Roscoe T.; Subramanian L.; Suzuki T.; Zhuang S.; Joseph A. D.; Katz Y.H.; Stoica I.; Proceedings of the First International Conference on Pervasive Computing. ACM Pervasive 2002, pp. 1-14.
- [*Enabling Active Flow Manipulation in Silicon-Based Network Forwarding Engines*](#). Lavian T.; Wang P.; Travostino F.; Subramanian S.; Duraraj R.; Hoang D.B.; Sethaput V.; Culler D.; Proceeding of the Active Networks Conference and Exposition, 2002.(DANCE) 29-30 May 2002, pp. 65-76.
- [*Practical Active Network Services within Content-Aware Gateways*](#). Subramanian S.; Wang P.; Durairaj R.; Rasimas J.; Travostino F.; Lavian T.; Hoang D.B.; Proceeding of the DARPA Active Networks Conference and Exposition, 2002.(DANCE) 29-30 May 2002, pp. 344-354.
- [*Active Networking on a Programmable Network Platform*](#). Wang P.Y.; Lavian T.; Duncan R.; Jaeger R.; Fourth IEEE Conference on Open Architectures and Network Programming (OPENARCH), Anchorage, April 2002.
- [*Intelligent Network Services through Active Flow Manipulation*](#). Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; IEEE Intelligent Networks 2001 Workshop (IN2001), Boston, May 2001.
- [*Intelligent Network Services through Active Flow Manipulation*](#). Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; Intelligent Network Workshop, 2001 IEEE 6-9 May 2001, pp.73 -82.
- [*Enabling Active Flow Manipulation in Silicon-based Network Forwarding Engine*](#). Lavian, T.; Wang, P.; Travostino, F.; Subramanian S.; Hoang D.B.; Sethaput V.; Culler D.; Journal of Communications and Networks, March 2001, pp.78-87.
- [*Active Networking on a Programmable Networking Platform*](#). Lavian T.; Wang P.Y.; IEEE Open Architectures and Network Programming, 2001, pp. 95-103.
- [*Enabling Active Networks Services on a Gigabit Routing Switch*](#). Wang P.; Jaeger R.; Duncan R.; Lavian T.; Travostino F.; 2nd Workshop on Active Middleware Services, 2000.
- [*Dynamic Classification in Silicon-Based Forwarding Engine Environments*](#). Jaeger R.; Duncan R.; Travostino F.; Lavian T.; Hollingsworth J.; Selected Papers. 10th IEEE Workshop on Metropolitan Area and Local Networks, 1999. 21-24 Nov. 1999, pp.103-109.
- [*Open Programmable Architecture for Java-Enabled Network Devices*](#). Lavian, T.; Jaeger, R. F.; Hollingsworth, J. K.; IEEE Hot Interconnects Stanford University, August 1999, pp. 265-277.
- *Open Java SNMP MIB API*. Rob Duncan, Tal Lavian, Roy Lee, Jason Zhou, Bay Architecture Lab Technical Report TR98-038, December 1998.
- *Java-Based Open Service Interface Architecture*. Lavian T.; Lau S.; BAL TR98-010 Bay Architecture Lab Technical Report, March 1998.

- *Parallel SIMD Architecture for Color Image Processing*. Lavian T. Tel – Aviv University, Tel – Aviv, Israel, November 1995.
- [Grid Network Services, Draft-ggf-ghpn-netservices-1.0](#). George Clapp, Tiziana Ferrari, Doan B. Hoang, Gigi Karmous-Edwards, Tal Lavian, Mark J. Leese, Paul Mealor, Inder Monga, Volker Sander, Franco Travostino, Global Grid Forum(GGF).
- [Project DRAC: Creating an applications-aware network](#). Travostino F.; Keates R.; Lavian T.; Monga I.; Schofield B.; Nortel Technical Journal, February 2005, pp. 23-26.
- [Optical Network Infrastructure for Grid, Draft-ggf-ghpn-opticalnets-1](#). Dimitra Simeonidou, Reza Nejabati, Bill St. Arnaud, Micah Beck, Peter Clarke, Doan B. Hoang, David Hutchison, Gigi Karmous-Edwards, Tal Lavian, Jason Leigh, Joe Mambretti, Volker Sander, John Strand, Franco Travostino, Global Grid Forum(GGF) GHPN Standard GFD-I.036 August 2004.
- [Popeye - Using Fine-grained Network Access Control to Support Mobile Users and Protect Intranet Hosts](#). Mike Chen, Barbara Hohlt, Tal Lavian, December 2000.

Presentations and Talks

(Not an exhaustive list)

- [Lambda Data Grid: An Agile Optical Platform for Grid Computing and Data-intensive Applications](#).
- [Web Services and OGSA](#)
- [WINER Workflow Integrated Network Resource Orchestration](#).
- [Technology & Society](#)
- [Abundant Bandwidth and how it affects us?](#)
- [Active Content Networking\(ACN\)](#)
- [DWDM-RAM:Enabling Grid Services with Dynamic Optical Networks](#)
- [Application-engaged Dynamic Orchestration of Optical Network Resources](#)
- [A Platform for Data Intensive Services Enabled by Next Generation Dynamic Optical Networks](#)
- [Optical Networks](#)
- [Grid Optical Network Service Architecture for Data Intensive Applications](#)
- [Optical Networking & DWDM](#)
- [OptiCal Inc.](#)
- [OptiCal & LUMOS Networks](#)
- [Optical Networking Services](#)
- [Business Models for Dynamically Provisioned Optical Networks](#)
- [Business Model Concepts for Dynamically Provisioned Optical Networks](#)
- [Optical Networks Infrastructure](#)
- [Research Challenges in agile optical networks](#)
- [Services and Applications' infrastructure for agile optical networks](#)
- [Impact on Society](#)
- [TeraGrid Communication and Computation](#)
- [Unified Device Management via Java-enabled Network Devices](#)
- [Active Network Node in Silicon-Based L3 Gigabit Routing Switch](#)
- [Active Nets Technology Transfer through High-Performance Network Devices](#)
- [Programmable Network Node: Applications](#)
- [Open Innovation via Java-enabled Network Devices](#)
- [Practical Considerations for Deploying a Java Active Networking Platform](#)
- [Open Java-Based Intelligent Agent Architecture for Adaptive Networking Devices](#)

- [Java SNMP Oplet](#)
- [Open Distributed Networking Intelligence: A New Java Paradigm](#)
- [Open Programmability](#)
- [Active Networking On A Programmable Networking Platform](#)
- [Open Networking through Programmability](#)
- [Open Programmable Architecture for Java-enabled Network Devices](#)
- [Integrating Active Networking and Commercial-Grade Routing Platforms](#)
- [Programmable Network Devices](#)
- [To be smart or not to be?](#)

ANSI/IEEE Std 802.11, 1999 Edition

[Adopted by ISO/IEC and redesignated as
ISO/IEC 8802-11:1999(E)]

**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

**Adopted by the ISO/IEC and redesignated as
ISO/IEC 8802-11:1999(E)**

Sponsor
LAN/MAN Standards Committee
of the
IEEE Computer Society

Abstract: The medium access control (MAC) and physical characteristics for wireless local area networks (LANs) are specified in this standard, part of a series of standards for local and metropolitan area networks. The medium access control unit in this standard is designed to support physical layer units as they may be adopted dependent on the availability of spectrum. This standard contains three physical layer units: two radio units, both operating in the 2400–2500 MHz band, and one baseband infrared unit. One radio unit employs the frequency-hopping spread spectrum technique, and the other employs the direct sequence spread spectrum technique.

Keywords: ad hoc network, infrared, LAN, local area network, mobility, radio frequency, wireless

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

Copyright © 1999 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1999. Printed in the United States of America.

ISBN 0-7381-1658-0

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

20 August 1999

SH94740

ANSI/IEEE Std 802.11, 1999 Edition

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. 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.

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.

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 all 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 technical 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 on standards and requests for interpretations should be addressed to:

Secretary, IEEE 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 all 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.

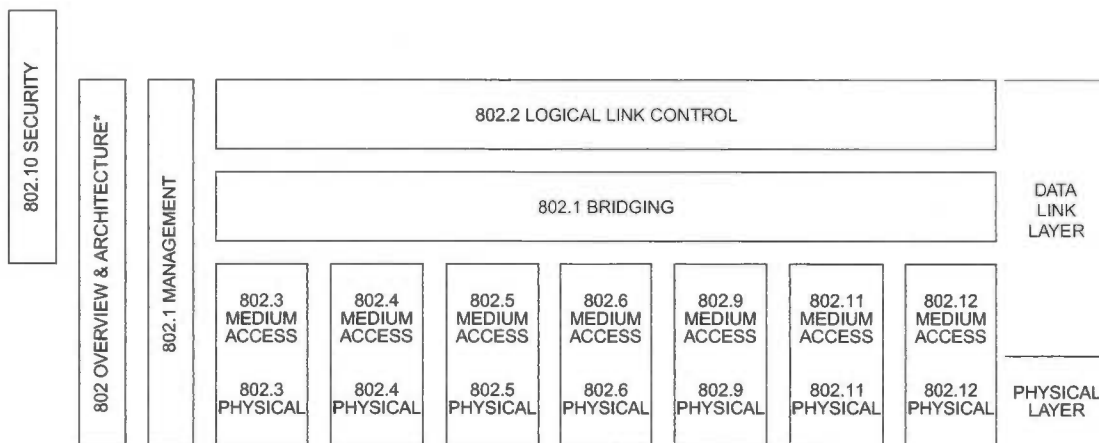
The patent holder has, however, filed a statement of assurance that it will grant a license under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to all applicants desiring to obtain such a license. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreement offered by the patent holder. Contact information may be obtained from the IEEE Standards Department.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction to ANSI/IEEE Std 802.11, 1999 Edition

(This introduction is not a part of ANSI/IEEE Std 802.11, 1999 Edition or of ISO/IEC 8802-11: 1999, but is included for information purpose only.)

This standard is part of a family of standards for local and metropolitan area networks. The relationship between the standard and other members of the family is shown below. (The numbers in the figure refer to IEEE standard numbers.)



* Formerly IEEE Std 802.1A.

This family of standards deals with the Physical and Data Link layers as defined by the International Organization for Standardization (ISO) Open Systems Interconnection (OSI) Basic Reference Model (ISO/IEC 7498-1: 1994). The access standards define seven types of medium access technologies and associated physical media, each appropriate for particular applications or system objectives. Other types are under investigation.

The standards defining the access technologies are as follows:

- IEEE Std 802 *Overview and Architecture.* This standard provides an overview to the family of IEEE 802 Standards.
- ANSI/IEEE Std 802.1B and 802.1k [ISO/IEC 15802-2] *LAN/MAN Management.* Defines an OSI management-compatible architecture, and services and protocol elements for use in a LAN/MAN environment for performing remote management.
- ANSI/IEEE Std 802.1D [ISO/IEC 15802-3] *Media Access Control (MAC) Bridges.* Specifies an architecture and protocol for the interconnection of IEEE 802 LANs below the MAC service boundary.
- ANSI/IEEE Std 802.1E [ISO/IEC 15802-4] *System Load Protocol.* Specifies a set of services and protocol for those aspects of management concerned with the loading of systems on IEEE 802 LANs.
- IEEE Std 802.1F *Common Definitions and Procedures for IEEE 802 Management Information*
- ANSI/IEEE Std 802.1G [ISO/IEC 15802-5] *Remote Media Access Control (MAC) Bridging.* Specifies extensions for the interconnection, using non-LAN communication technologies, of geographically separated IEEE 802 LANs below the level of the logical link control protocol.
- ANSI/IEEE Std 802.2 [ISO/IEC 8802-2] *Logical Link Control*

- ANSI/IEEE Std 802.3 [ISO/IEC 8802-3] *CSMA/CD Access Method and Physical Layer Specifications*
- ANSI/IEEE Std 802.4 [ISO/IEC 8802-4] *Token Passing Bus Access Method and Physical Layer Specifications*
- ANSI/IEEE Std 802.5 [ISO/IEC 8802-5] *Token Ring Access Method and Physical Layer Specifications*
- ANSI/IEEE Std 802.6 [ISO/IEC 8802-6] *Distributed Queue Dual Bus Access Method and Physical Layer Specifications*
- ANSI/IEEE Std 802.9 [ISO/IEC 8802-9] *Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers*
- ANSI/IEEE Std 802.10 *Interoperable LAN/MAN Security*
- IEEE Std 802.11 [ISO/IEC DIS 8802-11] *Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications*
- ANSI/IEEE Std 802.12 [ISO/IEC DIS 8802-12] *Demand Priority Access Method, Physical Layer and Repeater Specifications*

In addition to the family of standards, the following is a recommended practice for a common Physical Layer technology:

- IEEE Std 802.7 *IEEE Recommended Practice for Broadband Local Area Networks*

The following additional working group has authorized standards projects under development:

- IEEE 802.14 *Standard Protocol for Cable-TV Based Broadband Communication Network*

Conformance test methodology

An additional standards series, identified by the number 1802, has been established to identify the conformance test methodology documents for the 802 family of standards. Thus the conformance test documents for 802.3 are numbered 1802.3.

ANSI/IEEE Std 802.11, 1999 Edition [ISO/IEC 8802-11: 1999]

This standard is a revision of IEEE Std 802.11-1997. The Management Information Base according to OSI rules has been removed, many redundant management items have been removed, and Annex D has been completed with the Management Information Base according to SNMP. Minor changes have been made throughout the document.

This standard defines the protocol and compatible interconnection of data communication equipment via the “air”, radio or infrared, in a local area network (LAN) using the carrier sense multiple access protocol with collision avoidance (CSMA/CA) medium sharing mechanism. The medium access control (MAC) supports operation under control of an access point as well as between independent stations. The protocol includes authentication, association, and reassociation services, an optional encryption/decryption procedure, power management to reduce power consumption in mobile stations, and a point coordination function for time-bounded transfer of data. The standard includes the definition of the management information base (MIB) using Abstract Syntax Notation 1 (ASN.1) and specifies the MAC protocol in a formal way, using the Speci-

fication and Description Language (SDL). Both ASN.1 and SDL source code have been added on a floppy diskette.

The infrared implementation of the PHY supports 1 Mbit/s data rate with an optional 2 Mbit/s extension. The radio implementations of the PHY specify either a frequency-hopping spread spectrum (FHSS) supporting 1 Mbit/s and an optional 2 Mbit/s data rate or a direct sequence spread spectrum (DSSS) supporting both 1 and 2 Mbit/s data rates.

This standard contains state-of-the-art material. The area covered by this standard is undergoing evolution. Revisions are anticipated to this standard within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material. Information on the current revision state of this and other IEEE 802 standards may be obtained from

Secretary, IEEE Standards Board
 445 Hoes Lane
 P.O. Box 1331
 Piscataway, NJ 08855-1331 USA

Participants

At the time the draft of the 1999 version of this standard was sent to sponsor ballot, the IEEE 802.11 working group had the following voting members:

Victor Hayes, Chair **Stuart J. Kerry and Al Petrick, Vice Chairs**
Bob O'Hara, 802.11rev Task Group Chair and Technical Editor
George Fishel, Secretary

David Bagby, Mac Group Chair **Dean Kawaguchi, PHY Group and FH Chair**
Jan Boer, Direct Sequence Chair
Michael Fischer and Allen Heberling, State Diagram Editors

Naftali Chayat, Task Group a Chair **John Fakatselis, Task Group b Chair**
Victoria M. Poncini, Task Group c Chair

Jeff Abramowitz
 Keith B. Amundsen
 Carl F. Andren
 Kazuhiro Aoyagi
 Phil Belanger
 John Biddick
 Simon Black
 Ronald Brockmann
 Wesley Brodsky
 John H. Cafarella
 Ken Clements
 Wim Diepstraten
 Darrol Draper
 Peter Ecclesine
 Darwin Engwer
 Jeff Fischer
 Matthew Fischer
 Michael Fischer
 John Fisher
 Motohiro Gochi
 Tim Godfrey
 Jan Haagh

Karl Hannestad
 Robert Heile
 Maarten Hoeben
 Duane Hurne
 Masayuki Ikeda
 Richard Jai
 Donald C. Johnson
 Nobuo Karaki
 Isao Masaki
 Jim McDonald
 Gene Miller
 Akira Miura
 Masaharu Mori
 Masahiro Morikura
 Ravi P. Nalamati
 Colin Nayler
 Richard van Nee
 Tomoki Ohsawa
 Kazuhiro Okanoue
 Richard H. Paine
 Bob Pham
 Stanley A. Reible

William Roberts
 Kent G. Rollins
 Oren Rosenfeld
 Michael Rothenberg
 Clemens C. W. Ruppel
 Chandos Rypinski
 Anil K. Sanwalka
 Roy Sebring
 Mike Shiba
 Thomas Siep
 Donald I. Sloan
 Hitoshi Takanashi
 Satoru Toguchi
 Cherry Tom
 Mike Trompower
 Tom Tsoulogiannis
 Sarosh N. Vesuna
 Nien C. Wei
 Harry Worstell
 Timothy M. Zimmerman
 Jonathan M. Zweig
 Jim Zyren

Major contributions to the 1999 edition were received from the following individuals:

Henri Moelard

Richard Ozer

Amoud Zwemmer

The following members of the balloting committee voted on the 1999 version of this standard:

Kit Athul
 Thomas W. Bailey
 Peter K. Campbell
 James T. Carlo
 David E. Carlson
 Brian J. Casey
 Naftali Chayat
 Robert S. Crowder
 Wim Diepstraten
 Thomas J. Dineen
 Christos Douligeris
 Paul S. Eastman
 Philip H. Enslow
 Changxin Fan
 John W. Fendrich
 Michael A. Fischer
 George R. Fishel
 Harvey A. Freeman
 Robert J. Gagliano
 Patrick S. Gonía
 Julio Gonzalez-Sanz
 Chris G. Guy
 Victor Hayes
 Donald N. Heirman
 Henry Hoyt
 Raj Jain

A. Kamerman
 Dean M. Kawaguchi
 Edward R. Kelly
 Gary C. Kessler
 Yongbum Kim
 Stephen Barton Kruger
 Joseph Kubler
 Lanse M. Leach
 Jai Yong Lee
 Randolph S. Little
 Ronald Mahany
 Peter Martini
 Richard McBride
 Bennett Meyer
 Gene E. Milligan
 David S. Millman
 Hiroshi Miyano
 Warren Monroe
 John E. Montague
 Wayne D. Moyers
 Shimon Muller
 Ken Naganuma
 Paul Nikolich
 Robert O'Hara
 Donal O'Mahony
 Roger Pandanda

Ronald C. Petersen
 John R. Pickens
 Alberto Profumo
 Vikram Punj
 James A. Renfro
 Gary S. Robinson
 Edouard Y. Rocher
 James W. Romlein
 Floyd E. Ross
 Michael Rothenberg
 Christoph Ruland
 Anil K. Sanwalka
 James E. Schuessler
 Rich Seifert
 Leo Sintonen
 Patricia Thaler
 Mike Trompower
 Mark-Rene Uchida
 Emmanuel Van Lil
 Sarosh N. Vesuna
 James Vorhies
 Barry M. Vornbrock
 Qian-li Yang
 Oren Yuen
 Chris Zegelin
 Jonathan M. Zweig

At the time the draft of the 1997 version of this standard was sent to sponsor ballot, the IEEE 802.11 working group had the following voting members:

Victor Hayes, Chair

Stuart J. Kerry and Chris Zegelin, Vice Chairs

Bob O'Hara and Greg Ennis, Chief Technical Editors

George Fishel and Carolyn L. Heide, Secretaries

David Bagby, MAC Group Chair

C. Thomas Baumgartner, Infrared Chair and Editor

Jan Boer, Direct Sequence Chair

Michael Fischer, State Diagram Editor

Dean M. Kawaguchi, PHY Group and FH Chair

Mike Trompower, Direct Sequence Editor

Jeff Abramowitz
 Keith B. Amundsen
 Phil Belanger
 Manuel J. Betancor*
 John Biddick
 Simon Black
 Alessandro M. Bolea
 Pablo Brenner
 Peter E. Chadwick
 Naftali Chayat
 Jonathon Y. Cheah
 Hae Wook Choi
 Wim Diepstraten

Robert J. Egan
 Darwin Engwer
 John Fakatselis
 Matthew Fischer
 Keith S. Furuya
 Rich Gardner
 Ian Gifford
 Howard J. Hall
 Bill Huhn
 Donald C. Johnson
 Mikio Kiyono
 Joseph J. Kubler
 Arthur Lashbrook

F. J. Lopez-Hernandez
 Ronald Mahany
 Bob Marshall
 Jim McDonald
 Akira Miura
 Wayne D. Moyers
 Ravi P. Nalamati
 Mitsuji Okada
 Al Petrick
 Miri Ratner
 James A. Renfro
 William Roberts
 Jon Walter Rosdahl

Michael Rothenberg
 Chandos A. Rypinski
 Anil K. Sanwalka
 Roy Sebring
 Glen Sherwood
 Thomas Siep
 Nathan Silberman

Don Sloan
 Greg Smith
 Marvin L. Sojka
 Dave Strohschein
 Bert Sullam
 Mack Sullivan

Tom Tsoulogiannis
 Jeanine Valadez
 Sarosh Vesuna
 Richard E. White
 Donna A. Woznicki
 Timothy M. Zimmerman
 Johnny Zweig

Major contributions to the 1997 version were received from the following individuals:

Robert Achatz
 Ken Biba
 Paul Eastman
 Ed Geiger
 Larry van der Jagt

Richard Lee
 Kerry Lynn
 Michael Masleid
 John McKown
 K. S. Natarajan
 Jim Neally

Richard Ozer
 Thomas Phinney
 Leon S. Scaldeferri*
 Jim Schuessler
 François Y. Simon

*Deceased

The following persons were on the balloting committee for the 1997 version of this standard:

Bernhard Albert
 Jon M. Allingham
 Jack S. Andresen
 Kit Athul
 Anthony L. Barnes
 Robert T. Bell
 Manuel J. Betancor
 Simon Black
 Alan L. Bridges
 Graham Campbell
 James T. Carlo
 David E. Carlson
 Peter E. Chadwick
 Naftali Chayat
 Alan J. Chwick
 Ken Clements
 Robert S. Crowder
 Rifaat Dayem
 Wim Diepstraten
 Edward A. Dunlop
 Sourav K. Dutta
 Paul S. Eastman
 Peter Ecclesine
 Gregory Elkmann
 John E. Emrich
 Philip H. Enslow
 Changxin Fan
 Michael A. Fischer
 Harvey A. Freeman
 Robert J. Gagliano
 Patrick S. Gonia
 N. Douglas Grant
 Govert M. Griffioen
 Joseph L. Hammond
 Victor Hayes
 Kenneth C. Heck
 Jan Hoogendoorn
 Russell D. Housley
 Walter K. Hurwitz

Richard J. Iliff
 Tomoaki Ishifuji
 Carlos Islas-Perez
 Raj Jain
 A. Kamerman
 Peter M. Kelly
 Yongbum Kim
 Mikio Kiyono
 Thaddeus Kobylarz
 Stephen B. Kruger
 Joseph J. Kubler
 David J. Law
 Jai Yong Lee
 Jungtae Lee
 Daniel E. Lewis
 Randolph S. Little
 Ming T. Liu
 Joseph C. J. Loo
 Donald C. Loughry
 Robert D. Love
 Ronald Mahany
 Jim L. Mangin
 Peter Martini
 P. Takis Mathiopoulos
 Steve Messenger
 Bennett Meyer
 Ann Miller
 David S. Millman
 Hiroshi Miyano
 Stig Frode Mjolsnes
 W. Melody Moh
 John E. Montague
 Wayne D. Moyers
 Paul Nikolich
 Ellis S. Nolley
 Robert O'Hara
 Donal O'Mahony
 Roger Pandanda
 Lalit Mohan Patnaik
 Lucy W. Person

Thomas L. Phinney
 Vikram Prabhu
 Alberto Profumo
 David L. Propp
 Vikram Punj
 Andris Putnins
 Fernando Ramos
 James W. Romlein
 Floyd E. Ross
 Michael Rothenberg
 Christoph Ruland
 Chandos A. Rypinski
 Anil K. Sanwalka
 Gregory D. Schumacher
 Rich Seifert
 Lee A. Sendelbach
 Michael Serrone
 Adarshpal S. Sethi
 Donald A. Sheppard
 Nathan Silberman
 Joseph S. Skorupa
 Michael A. Smith
 Marvin L. Sojka
 Efsthathios D. Sykas
 Geoffrey O. Thompson
 Robert C. Tripi
 Mike Trompower
 David B. Turner
 Mark-Rene Uchida
 James Vorhies
 Yun-Che Wang
 Raymond P. Wenig
 Earl J. Whitaker
 David W. Wilson
 Jerry A. Wyatt
 Qian-Li Yang
 Iwen Yao
 Oren Yuen
 Jonathan M. Zweig

When the IEEE-SA Standards Board approved this standard on 18 March 1999, it had the following membership:

Richard J. Holleman, *Chair*

Donald N. Heirman, *Vice Chair*

Judith Gorman, *Secretary*

Satish K. Aggarwal
Dennis Bodson
Mark D. Bowman
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Ruben D. Garzon

James H. Gurney
Lowell G. Johnson
Robert J. Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
L. Bruce McClung
Daleep C. Mohla
Robert F. Munzner

Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Akio Tojo
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Also included is the following nonvoting IEEE-SA Standards Board liaison:

Robert E. Hebner

Valerie E. Zelenty
IEEE Standards Project Editor

Contents

1.	Overview	1
1.1	Scope	1
1.2	Purpose	1
2.	Normative references	2
3.	Definitions	3
4.	Abbreviations and acronyms	6
5.	General description	9
5.1	General description of the architecture	9
5.1.1	How wireless LAN systems are different	9
5.2	Components of the IEEE 802.11 architecture	10
5.2.1	The independent BSS as an ad hoc network	10
5.2.2	Distribution system concepts	11
5.2.3	Area concepts	12
5.2.4	Integration with wired LANs	14
5.3	Logical service interfaces	14
5.3.1	Station service (SS)	15
5.3.2	Distribution system service (DSS)	15
5.3.3	Multiple logical address spaces	16
5.4	Overview of the services	17
5.4.1	Distribution of messages within a DS	17
5.4.2	Services that support the distribution service	18
5.4.3	Access and confidentiality control services	19
5.5	Relationships between services	21
5.6	Differences between ESS and IBSS LANs	23
5.7	Message information contents that support the services	24
5.7.1	Data	25
5.7.2	Association	25
5.7.3	Reassociation	25
5.7.4	Disassociation	26
5.7.5	Privacy	26
5.7.6	Authentication	26
5.7.7	Deauthentication	27
5.8	Reference model	27
6.	MAC service definition	29
6.1	Overview of MAC services	29
6.1.1	Asynchronous data service	29
6.1.2	Security services	29
6.1.3	MSDU ordering	29
6.2	Detailed service specification	30
6.2.1	MAC data services	30
7.	Frame formats	34
7.1	MAC frame formats	34

7.1.1	Conventions	34
7.1.2	General frame format	34
7.1.3	Frame fields	35
7.2	Format of individual frame types.....	41
7.2.1	Control frames	41
7.2.2	Data frames	43
7.2.3	Management frames.....	45
7.3	Management frame body components	50
7.3.1	Fixed fields.....	50
7.3.2	Information elements	55
8.	Authentication and privacy	59
8.1	Authentication services.....	59
8.1.1	Open System authentication	59
8.1.2	Shared Key authentication	60
8.2	The Wired Equivalent Privacy (WEP) algorithm	61
8.2.1	Introduction.....	61
8.2.2	Properties of the WEP algorithm	62
8.2.3	WEP theory of operation	62
8.2.4	WEP algorithm specification	64
8.2.5	WEP Frame Body expansion.....	64
8.3	Security-Related MIB attributes	65
8.3.1	Authentication-Related MIB attributes.....	65
8.3.2	Privacy-Related MIB attributes	65
9.	MAC sublayer functional description.....	70
9.1	MAC architecture.....	70
9.1.1	Distributed coordination function (DCF).....	70
9.1.2	Point coordination function (PCF).....	70
9.1.3	Coexistence of DCF and PCF	71
9.1.4	Fragmentation/defragmentation overview	71
9.1.5	MAC data service	72
9.2	DCF.....	72
9.2.1	Carrier-sense mechanism	73
9.2.2	MAC-Level acknowledgments	73
9.2.3	Interframe space (IFS)	74
9.2.4	Random backoff time.....	75
9.2.5	DCF access procedure.....	76
9.2.6	Directed MPDU transfer procedure	82
9.2.7	Broadcast and multicast MPDU transfer procedure	83
9.2.8	ACK procedure	83
9.2.9	Duplicate detection and recovery.....	83
9.2.10	DCF timing relations.....	84
9.3	PCF	86
9.3.1	CFP structure and timing	87
9.3.2	PCF access procedure	88
9.3.3	PCF transfer procedure	89
9.3.4	Contention-Free polling list.....	92
9.4	Fragmentation	93
9.5	Defragmentation	94
9.6	Multirate support.....	95
9.7	Frame exchange sequences	95

9.8	MSDU transmission restrictions	97
10.	Layer management.....	98
10.1	Overview of management model.....	98
10.2	Generic management primitives	98
10.3	MLME SAP interface	100
10.3.1	Power management.....	100
10.3.2	Scan.....	101
10.3.3	Synchronization	103
10.3.4	Authenticate	105
10.3.5	De-authenticate	107
10.3.6	Associate	109
10.3.7	Reassociate.....	111
10.3.8	Disassociate.....	113
10.3.9	Reset.....	114
10.3.10	Start.....	116
10.4	PLME SAP interface.....	118
10.4.1	PLME-RESET.request.....	118
10.4.2	PLME-CHARACTERISTICS.request.....	118
10.4.3	PLME-CHARACTERISTICS.confirm	119
10.4.4	PLME-DSSSTESTMODE.request	121
10.4.5	PLME-DSSSTESTOUTPUT.request	122
11.	MAC sublayer management entity	123
11.1	Synchronization	123
11.1.1	Basic approach.....	123
11.1.2	Maintaining synchronization	123
11.1.3	Acquiring synchronization, scanning.....	125
11.1.4	Adjusting STA timers	127
11.1.5	Timing synchronization for frequency-hopping (FH) PHYs.....	128
11.2	Power management.....	128
11.2.1	Power management in an infrastructure network	128
11.2.2	Power management in an IBSS.....	133
11.3	Association and reassociation	136
11.3.1	STA association procedures.....	136
11.3.2	AP association procedures	136
11.3.3	STA reassociation procedures.....	136
11.3.4	AP reassociation procedures	137
11.4	Management information base (MIB) definitions	137
12.	Physical layer (PHY) service specification.....	138
12.1	Scope.....	138
12.2	PHY functions.....	138
12.3	Detailed PHY service specifications.....	138
12.3.1	Scope and field of application.....	138
12.3.2	Overview of the service	138
12.3.3	Overview of interactions.....	138
12.3.4	Basic service and options.....	139
12.3.5	PHY-SAP detailed service specification	140
13.	PHY management	147

14.	Frequency-Hopping spread spectrum (FHSS) PHY specification for the 2.4 GHz Industrial, Scientific, and Medical (ISM) band.....	148
14.1	Overview.....	148
14.1.1	Overview of FHSS PHY.....	148
14.1.2	FHSS PHY functions.....	148
14.1.3	Service specification method and notation.....	148
14.2	FHSS PHY-specific service parameter lists.....	149
14.2.1	Overview.....	149
14.2.2	TXVECTOR parameters.....	149
14.2.3	RXVECTOR parameters.....	150
14.3	FHSS PLCP sublayer.....	150
14.3.1	Overview.....	150
14.3.2	PLCP frame format.....	151
14.3.3	PLCP state machines.....	154
14.4	PLME SAP layer management.....	163
14.4.1	Overview.....	163
14.4.2	FH PHY specific MAC sublayer management entity (MLME) procedures.....	163
14.4.3	FH PHY layer management entity state machines.....	163
14.5	FHSS PMD sublayer services.....	166
14.5.1	Scope and field of application.....	166
14.5.2	Overview of services.....	166
14.5.3	Overview of interactions.....	166
14.5.4	Basic service and options.....	166
14.5.5	PMD_SAP detailed service specification.....	167
14.6	FHSS PMD sublayer, 1.0 Mbit/s.....	172
14.6.1	1 Mbit/s PMD operating specifications, general.....	172
14.6.2	Regulatory requirements.....	172
14.6.3	Operating frequency range.....	173
14.6.4	Number of operating channels.....	174
14.6.5	Operating channel center frequency.....	174
14.6.6	Occupied channel bandwidth.....	176
14.6.7	Minimum hop rate.....	176
14.6.8	Hop sequences.....	177
14.6.9	Unwanted emissions.....	179
14.6.10	Modulation.....	179
14.6.11	Channel data rate.....	180
14.6.12	Channel switching/settling time.....	180
14.6.13	Receive to transmit switch time.....	180
14.6.14	PMD transmit specifications.....	181
14.6.15	PMD receiver specifications.....	182
14.6.16	Operating temperature range.....	183
14.7	FHSS PMD sublayer, 2.0 Mbit/s.....	183
14.7.1	Overview.....	183
14.7.2	Four-Level GFSK modulation.....	184
14.7.3	Channel data rate.....	185
14.8	FHSS PHY management information base (MIB).....	186
14.8.1	Overview.....	186
14.8.2	FH PHY attributes.....	187
14.9	FH PHY characteristics.....	194
15.	Direct sequence spread spectrum (DSSS) PHY specification for the 2.4 GHz band designated for ISM applications.....	195

15.1	Overview.....	195
15.1.1	Scope.....	195
15.1.2	DSSS PHY functions.....	195
15.1.3	Service specification method and notation.....	196
15.2	DSSS PLCP sublayer.....	196
15.2.1	Overview.....	196
15.2.2	PLCP frame format.....	196
15.2.3	PLCP field definitions.....	196
15.2.4	PLCP/DSSS PHY data scrambler and descrambler.....	199
15.2.5	PLCP data modulation and modulation rate change.....	199
15.2.6	PLCP transmit procedure.....	199
15.2.7	PLCP receive procedure.....	200
15.3	DSSS physical layer management entity (PLME).....	203
15.3.1	PLME_SAP sublayer management primitives.....	203
15.3.2	DSSS PHY MIB.....	204
15.3.3	DS PHY characteristics.....	205
15.4	DSSS PMD sublayer.....	205
15.4.1	Scope and field of application.....	205
15.4.2	Overview of service.....	206
15.4.3	Overview of interactions.....	206
15.4.4	Basic service and options.....	206
15.4.5	PMD_SAP detailed service specification.....	208
15.4.6	PMD operating specifications, general.....	215
15.4.7	PMD transmit specifications.....	218
15.4.8	PMD receiver specifications.....	222
16.	Infrared (IR) PHY specification.....	224
16.1	Overview.....	224
16.1.1	Scope.....	225
16.1.2	IR PHY functions.....	225
16.1.3	Service specification method and notation.....	225
16.2	IR PLCP sublayer.....	226
16.2.1	Overview.....	226
16.2.2	PLCP frame format.....	226
16.2.3	PLCP modulation and rate change.....	226
16.2.4	PLCP field definitions.....	227
16.2.5	PLCP procedures.....	228
16.3	IR PMD sublayer.....	230
16.3.1	Overview.....	230
16.3.2	PMD operating specifications, general.....	230
16.3.3	PMD transmit specifications.....	233
16.3.4	PMD receiver specifications.....	236
16.3.5	Energy Detect, Carrier Sense, and CCA definitions.....	237
16.4	PHY attributes.....	239
Annex A	(normative) Protocol Implementation Conformance Statement (PICS) proforma.....	241
A.1	Introduction.....	241
A.2	Abbreviations and special symbols.....	241
A.2.1	Status symbols.....	241
A.2.2	General abbreviations.....	241
A.3	Instructions for completing the PICS proforma.....	242
A.3.1	General structure of the PICS proforma.....	242

A.3.2 Additional information.....	242
A.3.3 Exception information.....	243
A.3.4 Conditional status.....	243
A.4 PICS proforma—ISO/IEC 8802-11: 1999.....	244
A.4.1 Implementation identification.....	244
A.4.2 Protocol summary, ISO/IEC 8802-11: 1999.....	244
A.4.3 IUT configuration.....	245
A.4.4 MAC protocol.....	245
A.4.5 Frequency-Hopping PHY functions.....	250
A.4.6 Direct sequence PHY functions.....	252
A.4.7 Infrared baseband PHY functions.....	255
 Annex B (informative) Hopping sequences.....	 259
 Annex C (normative) Formal description of MAC operation.....	 272
C.1 Introduction to the MAC formal description.....	275
C.2 Data type and operator definitions for the MAC state machines.....	277
C.3 State Machines for MAC stations.....	324
C.4 State machines for MAC access point.....	400
 Annex D (normative) ASN.1 encoding of the MAC and PHY MIB.....	 469
 Annex E (informative) Bibliography.....	 512
E.1 General.....	512
E.2 Specification and description language (SDL) documentation.....	512

**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**

1. Overview

1.1 Scope

The scope of this standard is to develop a medium access control (MAC) and physical layer (PHY) specification for wireless connectivity for fixed, portable, and moving stations within a local area.

1.2 Purpose

The purpose of this standard is to provide wireless connectivity to automatic machinery, equipment, or stations that require rapid deployment, which may be portable or hand-held, or which may be mounted on moving vehicles within a local area. This standard also offers regulatory bodies a means of standardizing access to one or more frequency bands for the purpose of local area communication.

Specifically, this standard

- Describes the functions and services required by an IEEE 802.11 compliant device to operate within ad hoc and infrastructure networks as well as the aspects of station mobility (transition) within those networks.
- Defines the MAC procedures to support the asynchronous MAC service data unit (MSDU) delivery services.
- Defines several PHY signaling techniques and interface functions that are controlled by the IEEE 802.11 MAC.
- Permits the operation of an IEEE 802.11 conformant device within a wireless local area network (LAN) that may coexist with multiple overlapping IEEE 802.11 wireless LANs.
- Describes the requirements and procedures to provide privacy of user information being transferred over the wireless medium (WM) and authentication of IEEE 802.11 conformant devices.

2. Normative references

The following standards contain provisions which, through references in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

IEEE Std 802-1990, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.¹

IEEE Std C95.1-1991 (Reaff 1997), IEEE Standard Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3 kHz to 300 GHz.

ISO/IEC 7498-1: 1994, Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model.²

ISO/IEC 8802-2: 1998, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

ISO/IEC 8824-1: 1995, Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation.

ISO/IEC 8824-2: 1995, Information technology—Abstract Syntax Notation One (ASN.1): Information object specification.

ISO/IEC 8824-3: 1995, Information technology—Abstract Syntax Notation One (ASN.1): Constraint specification.

ISO/IEC 8824-4: 1995, Information technology—Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.

ISO/IEC 8825-1: 1995, Information technology—ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

ISO/IEC 8825-2: 1996, Information technology—ASN.1 encoding rules: Specification of Packed Encoding Rules (PER).

ISO/IEC 15802-1: 1995, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.

ITU Radio Regulations, volumes 1–4.³

ITU-T Recommendation X.210 (11/93), Information technology—Open systems interconnection—Basic Reference Model: Conventions for the definition of OSI services (*common text with ISO/IEC*).

ITU-T Recommendation Z.100 (03/93), CCITT specification and description language (SDL).

ITU-T Recommendation Z.105 (03/95), SDL combined with ASN.1 (SDL/ASN.1).

¹IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://www.standards.ieee.org/>).

²ISO and ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Genève 20, Switzerland/Suisse (<http://www.iso.ch/>). They are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).

³ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (<http://www.itu.int/>). They are also available in the United States from the U.S. Department of Commerce, Technology Administration, National Technical Information Service (NTIS), Springfield, VA 22161, USA.

3. Definitions

3.1 access control: The prevention of unauthorized usage of resources.

3.2 access point (AP): Any entity that has station functionality and provides access to the distribution services, via the wireless medium (WM) for associated stations.

3.3 ad hoc network: A network composed solely of stations within mutual communication range of each other via the wireless medium (WM). An ad hoc network is typically created in a spontaneous manner. The principal distinguishing characteristic of an ad hoc network is its limited temporal and spatial extent. These limitations allow the act of creating and dissolving the ad hoc network to be sufficiently straightforward and convenient so as to be achievable by nontechnical users of the network facilities; i.e., no specialized “technical skills” are required and little or no investment of time or additional resources is required beyond the stations that are to participate in the ad hoc network. The term *ad hoc* is often used as slang to refer to an independent basic service set (IBSS).

3.4 association: The service used to establish access point/station (AP/STA) mapping and enable STA invocation of the distribution system services (DSSs).

3.5 authentication: The service used to establish the identity of one station as a member of the set of stations authorized to associate with another station.

3.6 basic service area (BSA): The conceptual area within which members of a basic service set (BSS) may communicate.

3.7 basic service set (BSS): A set of stations controlled by a single coordination function.

3.8 basic service set (BSS) basic rate set: The set of data transfer rates that all the stations in a BSS will be capable of using to receive frames from the wireless medium (WM). The BSS basic rate set data rates are preset for all stations in the BSS.

3.9 broadcast address: A unique multicast address that specifies all stations.

3.10 channel: An instance of medium use for the purpose of passing protocol data units (PDUs) that may be used simultaneously, in the same volume of space, with other instances of medium use (on other channels) by other instances of the same physical layer (PHY), with an acceptably low frame error ratio due to mutual interference. Some PHYs provide only one channel, whereas others provide multiple channels. Examples of channel types are as shown in the following table:

Single channel	n-channel
Narrowband radio-frequency (RF) channel	Frequency division multiplexed channels
Baseband infrared	Direct sequence spread spectrum (DSSS) with code division multiple access

3.11 clear channel assessment (CCA) function: That logical function in the physical layer (PHY) that determines the current state of use of the wireless medium (WM).

3.12 confidentiality: The property of information that is not made available or disclosed to unauthorized individuals, entities, or processes.

3.13 coordination function: The logical function that determines when a station operating within a basic service set (BSS) is permitted to transmit and may be able to receive protocol data units (PDUs) via the wireless medium (WM). The coordination function within a BSS may have one point coordination function (PCF) and will have one distributed coordination function (DCF).

3.14 coordination function pollable: A station able to (1) respond to a coordination function poll with a data frame, if such a frame is queued and able to be generated, and (2) interpret acknowledgments in frames sent to or from the point coordinator.

3.15 deauthentication: The service that voids an existing authentication relationship.

3.16 directed address: *See: unicast frame.*

3.17 disassociation: The service that removes an existing association.

3.18 distributed coordination function (DCF): A class of coordination function where the same coordination function logic is active in every station in the basic service set (BSS) whenever the network is in operation.

3.19 distribution: The service that, by using association information, delivers medium access control (MAC) service data units (MSDUs) within the distribution system (DS).

3.20 distribution system (DS): A system used to interconnect a set of basic service sets (BSSs) and integrated local area networks (LANs) to create an extended service set (ESS).

3.21 distribution system medium (DSM): The medium or set of media used by a distribution system (DS) for communications between access points (APs) and portals of an extended service set (ESS).

3.22 distribution system service (DSS): The set of services provided by the distribution system (DS) that enable the medium access control (MAC) to transport MAC service data units (MSDUs) between stations that are not in direct communication with each other over a single instance of the wireless medium (WM). These services include transport of MSDUs between the access points (APs) of basic service sets (BSSs) within an extended service set (ESS), transport of MSDUs between portals and BSSs within an ESS, and transport of MSDUs between stations in the same BSS in cases where the MSDU has a multicast or broadcast destination address or where the destination is an individual address, but the station sending the MSDU chooses to involve DSS. DSSs are provided between pairs of IEEE 802.11 MACs.

3.23 extended rate set (ERS): The set of data transfer rates supported by a station (if any) beyond the extended service set (ESS) basic rate set. This set may include data transfer rates that will be defined in future physical layer (PHY) standards.

3.24 extended service area (ESA): The conceptual area within which members of an extended service set (ESS) may communicate. An ESA is larger than or equal to a basic service area (BSA) and may involve several basic service sets (BSSs) in overlapping, disjointed, or both configurations.

3.25 extended service set (ESS): A set of one or more interconnected basic service sets (BSSs) and integrated local area networks (LANs) that appears as a single BSS to the logical link control layer at any station associated with one of those BSSs.

3.26 Gaussian frequency shift keying (GFSK): A modulation scheme in which the data is first filtered by a Gaussian filter in the baseband and then modulated with a simple frequency modulation.

3.27 independent basic service set (IBSS): A BSS that forms a self-contained network, and in which no access to a distribution system (DS) is available.

3.28 infrastructure: The infrastructure includes the distribution system medium (DSM), access point (AP), and portal entities. It is also the logical location of distribution and integration service functions of an extended service set (ESS). An infrastructure contains one or more APs and zero or more portals in addition to the distribution system (DS).

3.29 integration: The service that enables delivery of medium access control (MAC) service data units (MSDUs) between the distribution system (DS) and an existing, non-IEEE 802.11 local area network (via a portal).

3.30 medium access control (MAC) management protocol data unit (MMPDU): The unit of data exchanged between two peer MAC entities to implement the MAC management protocol.

3.31 medium access control (MAC) protocol data unit (MPDU): The unit of data exchanged between two peer MAC entities using the services of the physical layer (PHY).

3.32 medium access control (MAC) service data unit (MSDU): Information that is delivered as a unit between MAC service access points (SAPs).

3.33 minimally conformant network: An IEEE 802.11 network in which two stations in a single basic service area (BSA) are conformant with ISO/IEC 8802-11: 1999.

3.34 mobile station: A type of station that uses network communications while in motion.

3.35 multicast: A medium access control (MAC) address that has the group bit set. A multicast MAC service data unit (MSDU) is one with a multicast destination address. A multicast MAC protocol data unit (MPDU) or control frame is one with a multicast receiver address.

3.36 network allocation vector (NAV): An indicator, maintained by each station, of time periods when transmission onto the wireless medium (WM) will not be initiated by the station whether or not the station's clear channel assessment (CCA) function senses that the WM is busy.

3.37 point coordination function (PCF): A class of possible coordination functions in which the coordination function logic is active in only one station in a basic service set (BSS) at any given time that the network is in operation.

3.38 portable station: A type of station that may be moved from location to location, but that only uses network communications while at a fixed location.

3.39 portal: The logical point at which medium access control (MAC) service data units (MSDUs) from a non-IEEE 802.11 local area network (LAN) enter the distribution system (DS) of an extended service set (ESS).

3.40 privacy: The service used to prevent the content of messages from being read by other than the intended recipients.

3.41 reassociation: The service that enables an established association [between access point (AP) and station (STA)] to be transferred from one AP to another (or the same) AP.

3.42 station (STA): Any device that contains an IEEE 802.11 conformant medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM).

3.43 station basic rate: A data transfer rate belonging to the extended service set (ESS) basic rate set that is used by a station for specific transmissions. The station basic rate may change dynamically as frequently as

each medium access control (MAC) protocol data unit (MPDU) transmission attempt, based on local considerations at that station.

3.44 station service (SS): The set of services that support transport of medium access control (MAC) service data units (MSDUs) between stations within a basic service set (BSS).

3.45 time unit (TU): A measurement of time equal to 1024 μ s.

3.46 unauthorized disclosure: The process of making information available to unauthorized individuals, entities, or processes.

3.47 unauthorized resource use: Use of a resource not consistent with the defined security policy.

3.48 unicast frame: A frame that is addressed to a single recipient, not a broadcast or multicast frame. *Syn:* directed address.

3.49 wired equivalent privacy (WEP): The optional cryptographic confidentiality algorithm specified by IEEE 802.11 used to provide data confidentiality that is subjectively equivalent to the confidentiality of a wired local area network (LAN) medium that does not employ cryptographic techniques to enhance privacy.

3.50 wireless medium (WM): The medium used to implement the transfer of protocol data units (PDUs) between peer physical layer (PHY) entities of a wireless local area network (LAN).

4. Abbreviations and acronyms

ACK	acknowledgment
AID	association identifier
AP	access point
ATIM	announcement traffic indication message
BSA	basic service area
BSS	basic service set
BSSID	basic service set identification
CCA	clear channel assessment
CF	contention free
CFP	contention-free period
CID	connection identifier
CP	contention period
CRC	cyclic redundancy code
CS	carrier sense
CTS	clear to send
CW	contention window
DA	destination address
DBPSK	differential binary phase shift keying
DCE	data communication equipment
DCF	distributed coordination function
DCLA	direct current level adjustment
DIFS	distributed (coordination function) interframe space
DLL	data link layer
Dp	desensitization
DQPSK	differential quadrature phase shift keying
DS	distribution system
DSAP	destination service access point
DSM	distribution system medium

DSS	distribution system service
DSSS	direct sequence spread spectrum
DTIM	delivery traffic indication message
ED	energy detection
EIFS	extended interframe space
EIRP	equivalent isotropically radiated power
ERS	extended rate set
ESA	extended service area
ESS	extended service set
FC	frame control
FCS	frame check sequence
FER	frame error ratio
FH	frequency hopping
FHSS	frequency-hopping spread spectrum
FIFO	first in first out
GFSK	Gaussian frequency shift keying
IBSS	independent basic service set
ICV	integrity check value
IDU	interface data unit
IFS	interframe space
IMp	intermodulation protection
IR	infrared
ISM	industrial, scientific, and medical
IV	initialization vector
LAN	local area network
LLC	logical link control
LME	layer management entity
LRC	long retry count
lsb	least significant bit
MAC	medium access control
MDF	management-defined field
MIB	management information base
MLME	MAC sublayer management entity
MMPDU	MAC management protocol data unit
MPDU	MAC protocol data unit
msb	most significant bit
MSDU	MAC service data unit
N/A	not applicable
NAV	network allocation vector
PC	point coordinator
PCF	point coordination function
PDU	protocol data unit
PHY	physical (layer)
PHY-SAP	physical layer service access point
PIFS	point (coordination function) interframe space
PLCP	physical layer convergence protocol
PLME	physical layer management entity
PMD	physical medium dependent
PMD-SAP	physical medium dependent service access point
PN	pseudo-noise (code sequence)
PPDU	PLCP protocol data unit
ppm	parts per million
PPM	pulse position modulation
PRNG	pseudo-random number generator

PS	power save (mode)
PSDU	PLCP SDU
RA	receiver address
RF	radio frequency
RSSI	received signal strength indication
RTS	request to send
RX	receive or receiver
SA	source address
SAP	service access point
SDU	service data unit
SFD	start frame delimiter
SIFS	short interframe space
SLRC	station long retry count
SME	station management entity
SMT	station management
SQ	signal quality (PN code correlation strength)
SRC	short retry count
SS	station service
SSAP	source service access point
SSID	service set identifier
SSRC	station short retry count
STA	station
TA	transmitter address
TBTT	target beacon transmission time
TIM	traffic indication map
TSF	timing synchronization function
TU	time unit
TX	transmit or transmitter
TXE	transmit enable
UCT	unconditional transition
WAN	wide area network
WDM	wireless distribution media
WDS	wireless distribution system
WEP	wired equivalent privacy
WM	wireless medium

5. General description

5.1 General description of the architecture

This subclause presents the concepts and terminology used within the ISO/IEC 8802-11: 1999 document (referred to throughout the text as IEEE 802.11). Specific terms are defined in Clause 3. Illustrations convey key IEEE 802.11 concepts and the interrelationships of the architectural components. IEEE 802.11 uses an architecture to describe functional components of an IEEE 802.11 LAN. The architectural descriptions are not intended to represent any specific physical implementation of IEEE 802.11.

5.1.1 How wireless LAN systems are different

Wireless networks have fundamental characteristics that make them significantly different from traditional wired LANs. Some countries impose specific requirements for radio equipment in addition to those specified in this standard.

5.1.1.1 Destination address does not equal destination location

In wired LANs, an address is equivalent to a physical location. This is implicitly assumed in the design of wired LANs. In IEEE 802.11, the addressable unit is a station (STA). The STA is a message destination, but not (in general) a fixed location.

5.1.1.2 The media impact the design

The physical layers used in IEEE 802.11 are fundamentally different from wired media. Thus IEEE 802.11 PHYs

- a) Use a medium that has neither absolute nor readily observable boundaries outside of which stations with conformant PHY transceivers are known to be unable to receive network frames.
- b) Are unprotected from outside signals.
- c) Communicate over a medium significantly less reliable than wired PHYs.
- d) Have dynamic topologies.
- e) Lack full connectivity, and therefore the assumption normally made that every STA can hear every other STA is invalid (i.e., STAs may be "hidden" from each other).
- f) Have time-varying and asymmetric propagation properties.

Because of limitations on wireless PHY ranges, wireless LANs intended to cover reasonable geographic distances may be built from basic coverage building blocks.

5.1.1.3 The impact of handling mobile stations

One of the requirements of IEEE 802.11 is to handle *mobile* as well as *portable* stations. A *portable* station is one that is moved from location to location, but that is only used while at a fixed location. *Mobile* stations actually access the LAN while in motion.

For technical reasons, it is not sufficient to handle only portable stations. Propagation effects blur the distinction between portable and mobile stations; stationary stations often appear to be mobile due to propagation effects.

Another aspect of mobile stations is that they may often be battery powered. Hence power management is an important consideration. For example, it cannot be presumed that a station's receiver will always be powered on.

5.1.1.4 Interaction with other IEEE 802 layers

IEEE 802.11 is required to appear to higher layers [logical link control (LLC)] as a current style IEEE 802 LAN. This requires that the IEEE 802.11 network handle station mobility within the MAC sublayer. To meet reliability assumptions (that LLC makes about lower layers), it is necessary for IEEE 802.11 to incorporate functionality that is untraditional for MAC sublayers.

5.2 Components of the IEEE 802.11 architecture

The IEEE 802.11 architecture consists of several components that interact to provide a wireless LAN that supports station mobility transparently to upper layers.

The basic service set (BSS) is the basic building block of an IEEE 802.11 LAN. Figure 1 shows two BSSs, each of which has two stations that are members of the BSS.

It is useful to think of the ovals used to depict a BSS as the coverage area within which the member stations of the BSS may remain in communication. (The concept of area, while not precise, is often good enough.) If a station moves out of its BSS, it can no longer directly communicate with other members of the BSS.

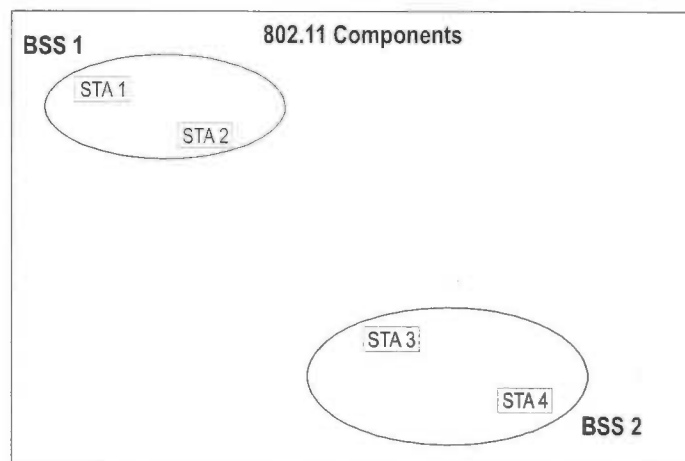


Figure 1—Basic service sets

5.2.1 The independent BSS as an ad hoc network

The independent BSS (IBSS) is the most basic type of IEEE 802.11 LAN. A minimum IEEE 802.11 LAN may consist of only two stations.

Figure 1 shows two IBSSs. This mode of operation is possible when IEEE 802.11 stations are able to communicate directly. Because this type of IEEE 802.11 LAN is often formed without pre-planning, for only as long as the LAN is needed, this type of operation is often referred to as an *ad hoc network*.

5.2.1.1 STA to BSS association is dynamic

The association between a STA and a BSS is dynamic (STAs turn on, turn off, come within range, and go out of range). To become a member of an infrastructure BSS, a station shall become “associated.” These associations are dynamic and involve the use of the distribution system service (DSS), which is described in 5.3.2.

5.2.2 Distribution system concepts

PHY limitations determine the direct station-to-station distance that may be supported. For some networks this distance is sufficient; for other networks, increased coverage is required.

Instead of existing independently, a BSS may also form a component of an extended form of network that is built with multiple BSSs. The architectural component used to interconnect BSSs is the *distribution system* (DS).

IEEE 802.11 logically separates the wireless medium (WM) from the distribution system medium (DSM). Each logical medium is used for different purposes, by a different component of the architecture. The IEEE 802.11 definitions neither preclude, nor demand, that the multiple media be either the same or different.

Recognizing that the multiple media are *logically* different is key to understanding the flexibility of the architecture. The IEEE 802.11 LAN architecture is specified independently of the physical characteristics of any specific implementation.

The DS enables mobile device support by providing the logical services necessary to handle address to destination mapping and seamless integration of multiple BSSs.

An access point (AP) is a STA that provides access to the DS by providing DS services in addition to acting as a STA.

Figure 2 adds the DS and AP components to the IEEE 802.11 architecture picture.

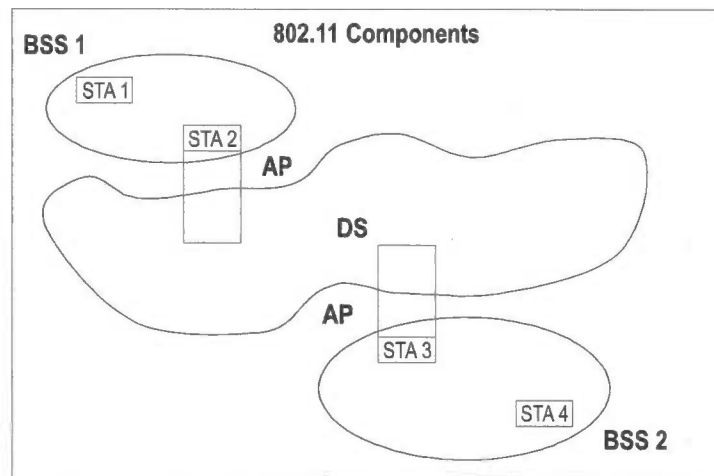


Figure 2—Distribution systems and access points

Data move between a BSS and the DS via an AP. Note that all APs are also STAs; thus they are addressable entities. The addresses used by an AP for communication on the WM and on the DSM are not necessarily the same.

5.2.2.1 Extended service set (ESS): The large coverage network

The DS and BSSs allow IEEE 802.11 to create a wireless network of arbitrary size and complexity. IEEE 802.11 refers to this type of network as the *extended service set* network.

The key concept is that the ESS network appears the same to an LLC layer as an IBSS network. Stations within an ESS may communicate and mobile stations may move from one BSS to another (within the same ESS) transparently to LLC.

Nothing is assumed by IEEE 802.11 about the relative physical locations of the BSSs in Figure 3.

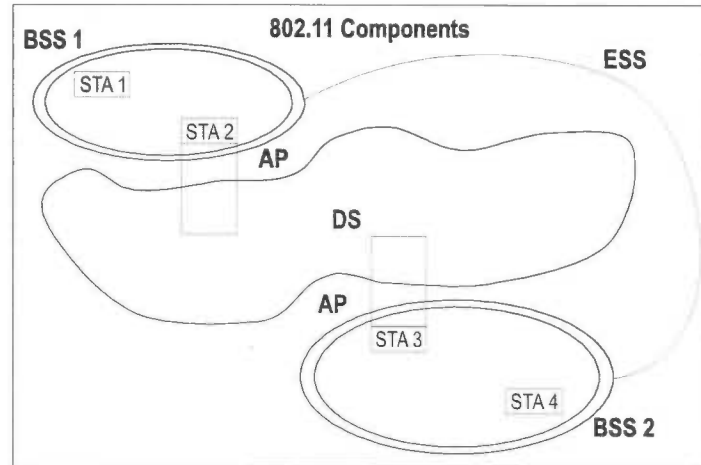


Figure 3—Extended service set

All of the following are possible:

- The BSSs may partially overlap. This is commonly used to arrange contiguous coverage within a physical volume.
- The BSSs could be physically disjointed. Logically there is no limit to the distance between BSSs.
- The BSSs may be physically collocated. This may be done to provide redundancy.
- One (or more) IBSS or ESS networks may be physically present in the same space as one (or more) ESS networks. This may arise for a number of reasons. Two of the most common are when an ad hoc network is operating in a location that also has an ESS network, and when physically overlapping IEEE 802.11 networks have been set up by different organizations.

5.2.3 Area concepts

For wireless PHYs, well-defined coverage areas simply do not exist. Propagation characteristics are dynamic and unpredictable. Small changes in position or direction may result in dramatic differences in signal strength. Similar effects occur whether a STA is stationary or mobile (as moving objects may impact station-to-station propagation).

Figure 4 shows a signal strength map for a simple square room with a standard metal desk and an open doorway. Figure 4 is a static snapshot; the propagation patterns change dynamically as stations and objects in the environment move. In Figure 4 the dark (solid) blocks in the lower left are a metal desk and there is a doorway at the top right of the figure. The figure indicates relative differences in field strength with different intensities and indicates the variability of field strength even in a static environment.

While the architecture diagrams show sharp boundaries for BSSs, this is an artifact of the pictorial representation, not a physical reality. Since dynamic three-dimensional field strength pictures are difficult to draw, well-defined shapes are used by IEEE 802.11 architectural diagrams to represent the coverage of a BSS.

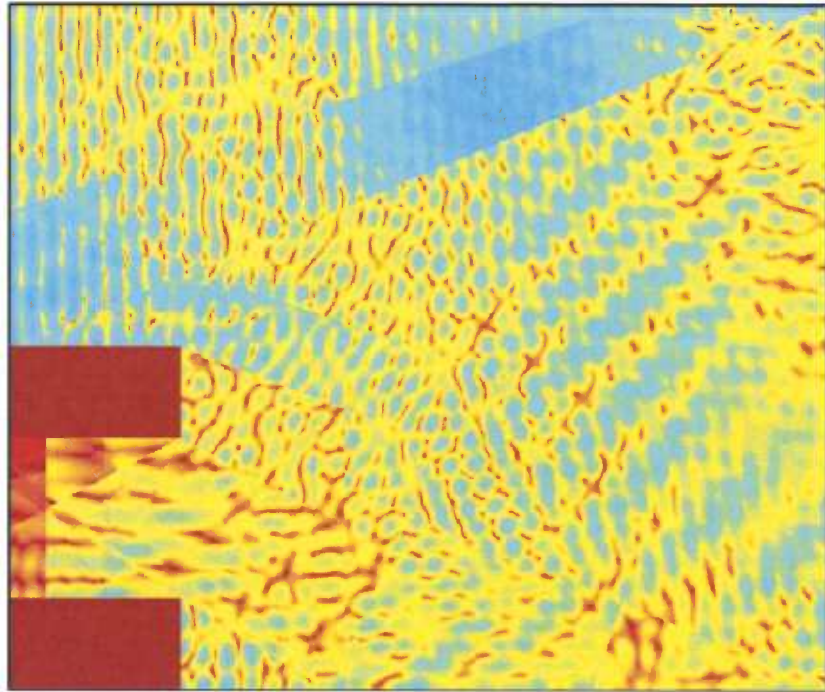


Figure 4—A representative signal intensity map

Further description difficulties arise when attempting to describe collocated coverage areas. Consider Figure 5, in which STA 6 could belong to BSS 2 or BSS 3.

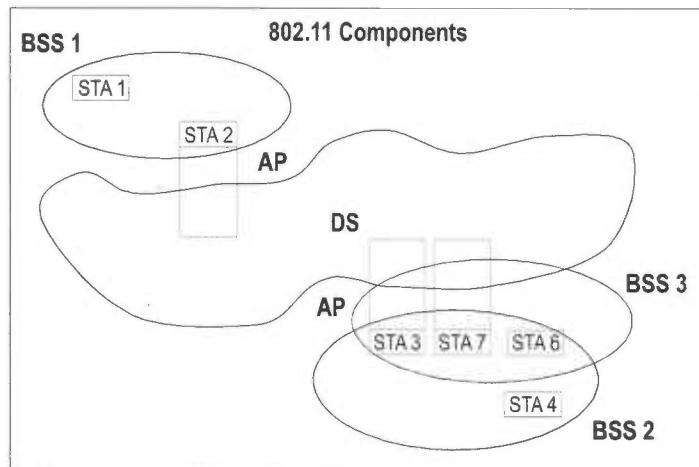


Figure 5—Collocated coverage areas

While the concept of sets of stations is correct, it is often convenient to talk about areas. For many topics the concept of area is sufficient. *Volume* is a more precise term than area, though still not technically correct. For historical reasons and convenience, this standard uses the common term *area*.

5.2.4 Integration with wired LANs

To integrate the IEEE 802.11 architecture with a traditional wired LAN, a final *logical* architectural component is introduced—a *portal*.

A portal is the logical point at which MSDUs from an integrated non-IEEE 802.11 LAN enter the IEEE 802.11 DS. For example, a portal is shown in Figure 6 connecting to a wired IEEE 802 LAN.

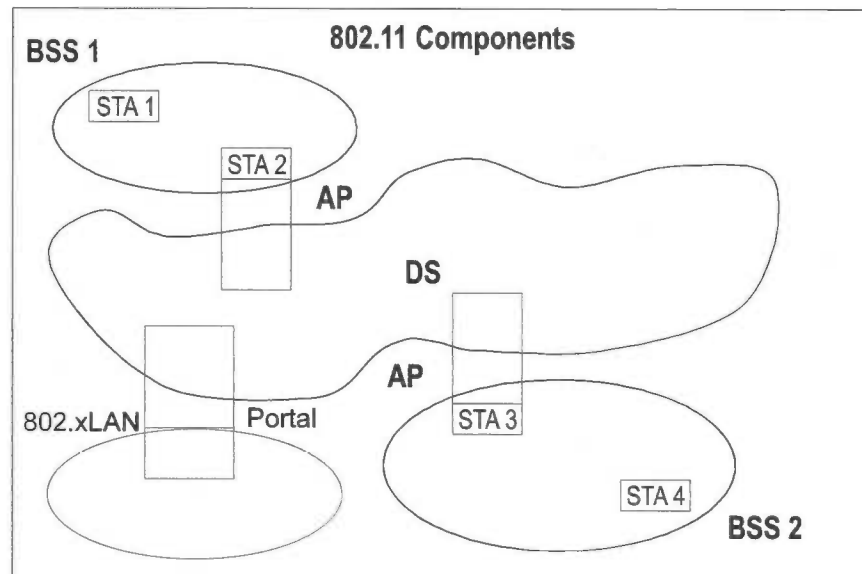


Figure 6—Connecting to other IEEE 802 LANs

All data from non-IEEE 802.11 LANs enter the IEEE 802.11 architecture via a portal. The portal provides logical integration between the IEEE 802.11 architecture and existing wired LANs. It is possible for one device to offer both the functions of an AP and a portal; this could be the case when a DS is implemented from IEEE 802 LAN components.

In IEEE 802.11, the ESS architecture (APs and the DS) provides traffic segmentation and range extension. Logical connections between IEEE 802.11 and other LANs are via the portal. Portals connect between the DSM and the LAN medium that is to be integrated.

5.3 Logical service interfaces

The IEEE 802.11 architecture allows for the possibility that the DS may not be identical to an existing wired LAN. A DS may be created from many different technologies including current IEEE 802 wired LANs. IEEE 802.11 does not constrain the DS to be either data link or network layer based. Nor does IEEE 802.11 constrain a DS to be either centralized or distributed in nature.

IEEE 802.11 explicitly does not specify the details of DS implementations. Instead, IEEE 802.11 specifies *services*. The services are associated with different components of the architecture. There are two categories of IEEE 802.11 service—the station service (SS) and the distribution system service (DSS). Both categories of service are used by the IEEE 802.11 MAC sublayer.

The complete set of IEEE 802.11 architectural services are as follows:

- a) Authentication
- b) Association
- c) Deauthentication
- d) Disassociation
- e) Distribution
- f) Integration
- g) Privacy
- h) Reassociation
- i) MSDU delivery

This set of services is divided into two groups: those that are part of every STA, and those that are part of a DS.

5.3.1 Station service (SS)

The service provided by stations is known as the *station service*.

The SS is present in every IEEE 802.11 station (including APs, as APs include station functionality). The SS is specified for use by MAC sublayer entities. All conformant stations provide SS.

The SS is as follows:

- a) Authentication
- b) Deauthentication
- c) Privacy
- d) MSDU delivery

5.3.2 Distribution system service (DSS)

The service provided by the DS is known as the *distribution system service*.

These services are represented in the IEEE 802.11 architecture by arrows within the APs, indicating that the services are used to cross media and address space logical boundaries. This is the convenient place to show the services in the picture. The physical embodiment of various services may or may not be within a physical AP.

The DSSs are provided by the DS. They are accessed via a STA that also provides DSSs. A STA that is providing access to DSS is an AP.

The DSSs are as follows:

- a) Association
- b) Disassociation
- c) Distribution
- d) Integration
- e) Reassociation

DSSs are specified for use by MAC sublayer entities.

Figure 7 combines the components from previous figures with both types of services to show the complete IEEE 802.11 architecture.

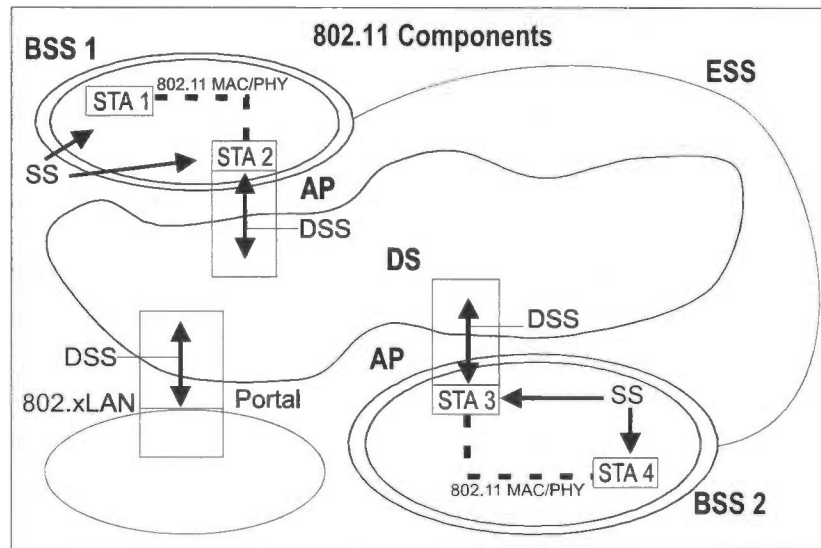


Figure 7—Complete IEEE 802.11 architecture

5.3.3 Multiple logical address spaces

Just as the IEEE 802.11 architecture allows for the possibility that the WM, DSM, and an integrated wired LAN may all be different physical media, it also allows for the possibility that each of these components may be operating within different address spaces. IEEE 802.11 only uses and specifies the use of the WM address space.

Each IEEE 802.11 PHY operates in a single medium—the WM. The IEEE 802.11 MAC operates in a single address space. MAC addresses are used on the WM in the IEEE 802.11 architecture. Therefore, it is unnecessary for the standard to explicitly specify that its addresses are “WM addresses.” This is assumed throughout this standard.

IEEE 802.11 has chosen to use the IEEE 802 48-bit address space (see 7.1.3.3.1). Thus IEEE 802.11 addresses are compatible with the address space used by the IEEE 802 LAN family.

The IEEE 802.11 choice of address space implies that for many instantiations of the IEEE 802.11 architecture, the wired LAN MAC address space and the IEEE 802.11 MAC address space may be the same. In those situations where a DS that uses MAC level IEEE 802 addressing is appropriate, all three of the logical address spaces used within a system could be identical. While this is a common case, it is not the only combination allowed by the architecture. The IEEE 802.11 architecture allows for all three logical address spaces to be distinct.

A multiple address space example is one in which the DS implementation uses network layer addressing. In this case, the WM address space and the DS address space would be different.

The ability of the architecture to handle multiple logical media and address spaces is key to the ability of IEEE 802.11 to be independent of the DS implementation and to interface cleanly with network layer mobility approaches. The implementation of the DS is unspecified and is beyond the scope of this standard.

5.4 Overview of the services

There are nine services specified by IEEE 802.11. Six of the services are used to support MSDU delivery between STAs. Three of the services are used to control IEEE 802.11 LAN access and confidentiality.

This subclause presents the services, an overview of how each service is used, and a description of how each service relates to other services and the IEEE 802.11 architecture. The services are presented in an order designed to help build an understanding of the operation of an IEEE 802.11 ESS network. As a result, the SSSs and DSSs are intermixed in order (rather than being grouped by category).

Each of the services is supported by one or more MAC frame types. Some of the services are supported by MAC management messages and some by MAC data messages. All of the messages gain access to the WM via the IEEE 802.11 MAC sublayer medium access method specified in Clause 9.

The IEEE 802.11 MAC sublayer uses three types of messages—*data*, *management*, and *control* (see Clause 7). The data messages are handled via the MAC data service path.

MAC management messages are used to support the IEEE 802.11 services and are handled via the MAC management service data path.

MAC control messages are used to support the delivery of IEEE 802.11 data and management messages.

The examples here assume an ESS network environment. The differences between the ESS and the IBSS network environments are discussed separately in 5.6.

5.4.1 Distribution of messages within a DS

5.4.1.1 Distribution

This is the primary service used by IEEE 802.11 STAs. It is conceptually invoked by every data message to or from an IEEE 802.11 STA operating in an ESS (when the frame is sent via the DS). Distribution is via a DSS.

Refer to the ESS network in Figure 7 and consider a data message being sent from STA 1 to STA 4. The message is sent from STA 1 and received by STA 2 (the “input” AP). The AP gives the message to the distribution service of the DS. It is the job of the distribution service to deliver the message within the DS in such a way that it arrives at the appropriate DS destination for the intended recipient. In this example, the message is distributed to STA 3 (the “output” AP) and STA 3 accesses the WM to send the message to STA 4 (the intended destination).

How the message is distributed within the DS is not specified by IEEE 802.11. All IEEE 802.11 is required to do is to provide the DS with enough information for the DS to be able to determine the “output” point that corresponds to the desired recipient. The necessary information is provided to the DS by the three association related services (association, reassociation, and disassociation).

The previous example was a case in which the AP that invoked the distribution service was different from the AP that received the distributed message. If the message had been intended for a station that was a member of the same BSS as the sending station, then the “input” and “output” APs for the message would have been the same.

In either example, the distribution service was logically invoked. Whether the message actually had to traverse the physical DSM or not is a DS implementation matter and is not specified by this standard.

While IEEE 802.11 does not specify DS implementations, it does recognize and support the use of the WM as the DSM. This is specifically supported by the IEEE 802.11 frame formats. (Refer to Clause 7 for details.)

5.4.1.2 Integration

If the distribution service determines that the intended recipient of a message is a member of an integrated LAN, the “output” point of the DS would be a portal instead of an AP.

Messages that are distributed to a portal cause the DS to invoke the Integration function (conceptually after the distribution service). The Integration function is responsible for accomplishing whatever is needed to deliver a message from the DSM to the integrated LAN media (including any required media or address space translations). Integration is a DSS.

Messages received from an integrated LAN (via a portal) by the DS for an IEEE 802.11 STA will invoke the Integration function before the message is distributed by the distribution service.

The details of an Integration function are dependent on a specific DS implementation and are outside the scope of this standard.

5.4.2 Services that support the distribution service

The primary purpose of a MAC sublayer is to transfer MSDUs between MAC sublayer entities. The information required for the distribution service to operate is provided by the association services. Before a data message can be handled by the distribution service, a STA shall be “associated.”

To understand the concept of association, it is necessary first to understand the concept of mobility.

5.4.2.1 Mobility types

The three transition types of significance to this standard that describe the mobility of stations within a network are as follows:

- a) **No-transition:** In this type, two subclasses that are usually indistinguishable are identified:
 - 1) Static—no motion.
 - 2) Local movement—movement within the PHY range of the communicating STAs [i.e., movement within a basic service area (BSA)].
- b) **BSS-transition:** This type is defined as a station movement from one BSS in one ESS to another BSS within the same ESS.
- c) **ESS-transition:** This type is defined as station movement from a BSS in one ESS to a BSS in a different ESS. This case is supported only in the sense that the STA may move. Maintenance of upper-layer connections cannot be guaranteed by IEEE 802.11; in fact, disruption of service is likely to occur.

The different association services support the different categories of mobility.

5.4.2.2 Association

To deliver a message within a DS, the distribution service needs to know which AP to access for the given IEEE 802.11 STA. This information is provided to the DS by the concept of association. Association is necessary, but not sufficient, to support BSS-transition mobility. Association is sufficient to support no-transition mobility. Association is a DSS.

Before a STA is allowed to send a data message via an AP, it shall first become associated with the AP. The act of becoming associated invokes the association service, which provides the STA to AP mapping to the

DS. The DS uses this information to accomplish its message distribution service. How the information provided by the association service is stored and managed within the DS is not specified by this standard.

At any given instant, a STA may be associated with no more than one AP. This ensures that the DS may determine a unique answer to the question, "Which AP is serving STA X?" Once an association is completed, a STA may make full use of a DS (via the AP) to communicate. Association is always initiated by the mobile STA, not the AP.

An AP may be associated with many STAs at one time.

A STA learns what APs are present and then requests to establish an association by invoking the association service. For details of how a station learns about what APs are present, see 11.1.3.

5.4.2.3 Reassociation

Association is sufficient for no-transition message delivery between IEEE 802.11 stations. Additional functionality is needed to support BSS-transition mobility. The additional required functionality is provided by the reassociation service. Reassociation is a DSS.

The reassociation service is invoked to "move" a current association from one AP to another. This keeps the DS informed of the current mapping between AP and STA as the station moves from BSS to BSS within an ESS. Reassociation also enables changing association attributes of an established association while the STA remains associated with the same AP. Reassociation is always initiated by the mobile STA.

5.4.2.4 Disassociation

The disassociation service is invoked whenever an existing association is to be terminated. Disassociation is a DSS.

In an ESS, this tells the DS to void existing association information. Attempts to send messages via the DS to a disassociated STA will be unsuccessful.

The disassociation service may be invoked by either party to an association (non-AP STA or AP). Disassociation is a notification, not a request. Disassociation cannot be refused by either party to the association.

APs may need to disassociate STAs to enable the AP to be removed from a network for service or for other reasons.

STAs shall attempt to disassociate whenever they leave a network. However, the MAC protocol does not depend on STAs invoking the disassociation service. (MAC management is designed to accommodate loss of an associated STA.)

5.4.3 Access and confidentiality control services

Two services are required for IEEE 802.11 to provide functionality equivalent to that which is inherent to wired LANs. The design of wired LANs assumes the physical attributes of wire. In particular, wired LAN design assumes the physically closed and controlled nature of wired media. The physically open medium nature of an IEEE 802.11 LAN violates those assumptions.

Two services are provided to bring the IEEE 802.11 functionality in line with wired LAN assumptions; authentication and privacy. Authentication is used instead of the wired media physical connection. Privacy is used to provide the confidential aspects of closed wired media.

5.4.3.1 Authentication

In wired LANs, physical security can be used to prevent unauthorized access. This is impractical in wireless LANs since they have a medium without precise bounds.

IEEE 802.11 provides the ability to control LAN access via the authentication service. This service is used by all stations to establish their identity to stations with which they will communicate. This is true for both ESS and IBSS networks. If a mutually acceptable level of authentication has not been established between two stations, an association shall not be established. Authentication is an SS.

IEEE 802.11 supports several authentication processes. The IEEE 802.11 authentication mechanism also allows expansion of the supported authentication schemes. IEEE 802.11 does not mandate the use of any particular authentication scheme.

IEEE 802.11 provides link-level authentication between IEEE 802.11 STAs. IEEE 802.11 does not provide either end-to-end (message origin to message destination) or user-to-user authentication. IEEE 802.11 authentication is used simply to bring the wireless link up to the assumed physical standards of a wired link. (This use of authentication is independent of any authentication process that may be used in higher levels of a network protocol stack.) If authentication other than that described here is desired, it is recommended that IEEE Std 802.10-1992 [B3]⁴ be implemented.

If desired, an IEEE 802.11 network may be operated using Open System authentication (see 8.1.1). This may violate implicit assumptions made by higher network layers. In an Open System, any station may become authenticated.

IEEE 802.11 also supports Shared Key authentication. Use of this authentication mechanism requires implementation of the wired equivalent privacy (WEP) option (see 8.2). In a Shared Key authentication system, identity is demonstrated by knowledge of a shared, secret, WEP encryption key.

Management information base (MIB) functions are provided to support the standardized authentication schemes.

IEEE 802.11 requires mutually acceptable, successful, authentication.

A STA may be authenticated with many other STAs at any given instant.

5.4.3.1.1 Preauthentication

Because the authentication process could be time-consuming (depending on the authentication protocol in use), the authentication service can be invoked independently of the association service.

Preauthentication is typically done by a STA while it is already associated with an AP (with which it previously authenticated). IEEE 802.11 does not require that STAs preauthenticate with APs. However, authentication is required before an association can be established.

If the authentication is left until reassociation time, this may impact the speed with which a STA can reassociate between APs, limiting BSS-transition mobility performance. The use of preauthentication takes the authentication service overhead out of the time-critical reassociation process.

⁴The numbers in brackets correspond to those of the bibliography in Annex E.

5.4.3.2 Deauthentication

The deauthentication service is invoked whenever an existing authentication is to be terminated. Deauthentication is an SS.

In an ESS, since authentication is a prerequisite for association, the act of deauthentication shall cause the station to be disassociated. The deauthentication service may be invoked by either authenticated party (non-AP STA or AP). Deauthentication is not a request; it is a notification. Deauthentication shall not be refused by either party. When an AP sends a deauthentication notice to an associated STA, the association shall also be terminated.

5.4.3.3 Privacy

In a wired LAN, only those stations physically connected to the wire may hear LAN traffic. With a wireless shared medium, this is not the case. Any IEEE 802.11-compliant STA may hear all like-PHY IEEE 802.11 traffic that is within range. Thus the connection of a single wireless link (without privacy) to an existing wired LAN may seriously degrade the security level of the wired LAN.

To bring the functionality of the wireless LAN up to the level implicit in wired LAN design, IEEE 802.11 provides the ability to encrypt the contents of messages. This functionality is provided by the privacy service. Privacy is an SS.

IEEE 802.11 specifies an optional privacy algorithm, WEP, that is designed to satisfy the goal of wired LAN “equivalent” privacy. The algorithm is not designed for ultimate security but rather to be “at least as secure as a wire.” See Clause 8 for more details.

IEEE 802.11 uses the WEP mechanism (see Clause 8) to perform the actual encryption of messages. MIB functions are provided to support WEP.

Note that privacy may only be invoked for data frames and some Authentication Management frames. All stations initially start “in the clear” in order to set up the authentication and privacy services.

The default privacy state for all IEEE 802.11 STAs is “in the clear.” If the privacy service is not invoked, all messages shall be sent unencrypted. If this default is not acceptable to one party or the other, data frames shall not be successfully communicated between the LLC entities. Unencrypted data frames received at a station configured for mandatory privacy, as well as encrypted data frames using a key not available at the receiving station, are discarded without an indication to LLC (or without indication to distribution services in the case of “To DS” frames received at an AP). These frames are acknowledged on the WM [if received without frame check sequence (FCS) error] to avoid wasting WM bandwidth on retries.

5.5 Relationships between services

A STA keeps two state variables for each STA with which direct communication via the WM is needed:

- *Authentication state*: The values are unauthenticated and authenticated.
- *Association state*: The values are unassociated and associated.

These two variables create three local states for each remote STA:

- *State 1*: Initial start state, unauthenticated, unassociated.
- *State 2*: Authenticated, not associated.
- *State 3*: Authenticated and associated.

The relationships between these station state variables and the services are given in Figure 8.

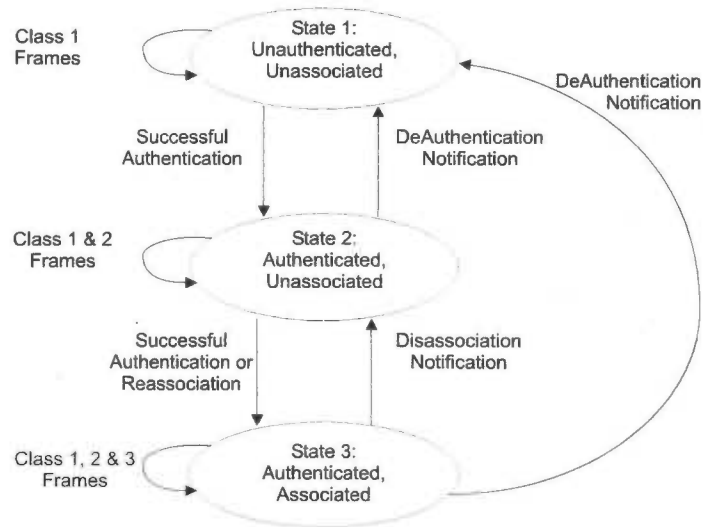


Figure 8—Relationship between state variables and services

The current state existing between the source and destination station determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs (see Clause 7). The state of the sending STA given by Figure 8 is with respect to the intended receiving STA. The allowed frame types are grouped into classes and the classes correspond to the station state. In State 1, only Class 1 frames are allowed. In State 2, either Class 1 or Class 2 frames are allowed. In State 3, all frames are allowed (Classes 1, 2, and 3). The frame classes are defined as follows:

- a) Class 1 frames (permitted from within States 1, 2, and 3):
 - 1) Control frames
 - i) Request to send (RTS)
 - ii) Clear to send (CTS)
 - iii) Acknowledgment (ACK)
 - iv) Contention-Free (CF)-End+ACK
 - v) CF-End
 - 2) Management frames
 - i) Probe request/response
 - ii) Beacon
 - iii) Authentication: Successful authentication enables a station to exchange Class 2 frames. Unsuccessful authentication leaves the STA in State 1.
 - iv) Deauthentication: Deauthentication notification when in State 2 or State 3 changes the STA's state to State 1. The STA shall become authenticated again prior to sending Class 2 frames.
 - v) Announcement traffic indication message (ATIM)
 - 3) Data frames
 - i) Data: Data frames with frame control (FC) bits "To DS" and "From DS" both false.
- b) Class 2 frames (if and only if authenticated; allowed from within States 2 and 3 only):
 - 1) Management frames:
 - i) Association request/response
 - Successful association enables Class 3 frames.
 - Unsuccessful association leaves STA in State 2.
 - ii) Reassociation request/response
 - Successful reassociation enables Class 3 frames.

- Unsuccessful reassociation leaves the STA in State 2 (with respect to the STA that was sent the reassociation message). Reassociation frames shall only be sent if the sending STA is already associated in the same ESS.
- iii) Disassociation
 - Disassociation notification when in State 3 changes a Station's state to State 2. This station shall become associated again if it wishes to utilize the DS.

If STA A receives a Class 2 frame with a unicast address in the Address 1 field from STA B that is not authenticated with STA A, STA A shall send a deauthentication frame to STA B.

- c) Class 3 frames (if and only if associated; allowed only from within State 3):
- 1) Data frames
 - Data subtypes: Data frames allowed. That is, either the "To DS" or "From DS" FC bits may be set to true to utilize DSSs.
 - 2) Management frames
 - Deauthentication: Deauthentication notification when in State 3 implies disassociation as well, changing the STA's state from 3 to 1. The station shall become authenticated again prior to another association.
 - 3) Control frames
 - PS-Poll

If STA A receives a Class 3 frame with a unicast address in the Address 1 field from STA B that is authenticated but not associated with STA A, STA A shall send a disassociation frame to STA B.

If STA A receives a Class 3 frame with a unicast address in the Address 1 field from STA B that is not authenticated with STA A, STA A shall send a deauthentication frame to STA B.

(The use of the word "receive" in this subclause refers to a frame that meets all of the filtering criteria specified in Clauses 8 and 9.)

5.6 Differences between ESS and IBSS LANs

In 5.2.1 the concept of the IBSS LAN was introduced. It was noted that an IBSS is often used to support an ad hoc network. In an IBSS network, a STA communicates directly with one or more other STAs.

Consider the full IEEE 802.11 architecture as shown in Figure 9.

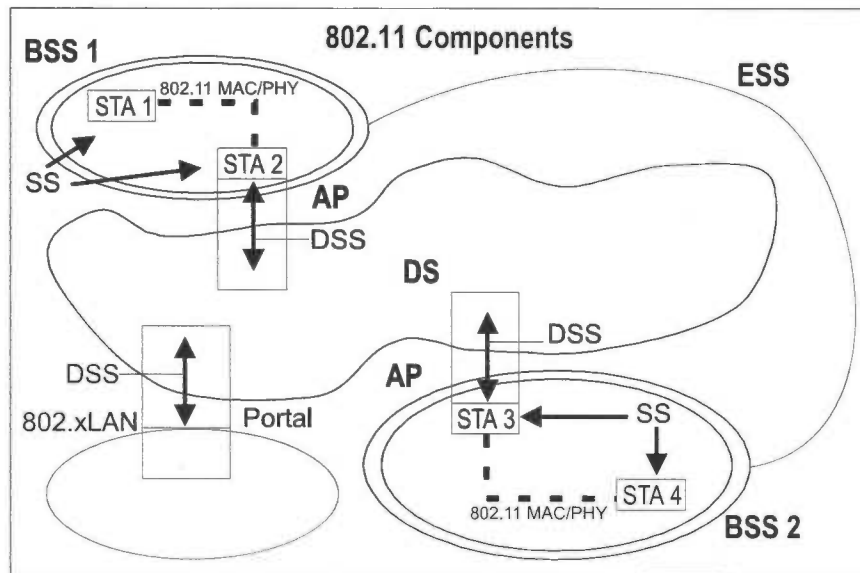


Figure 9—IEEE 802.11 architecture (again)

An IBSS consists of STAs that are directly connected. Thus there is (by definition) only one BSS. Further, since there is no physical DS, there cannot be a portal, an integrated wired LAN, or the DSSs. The logical picture reduces to Figure 10.

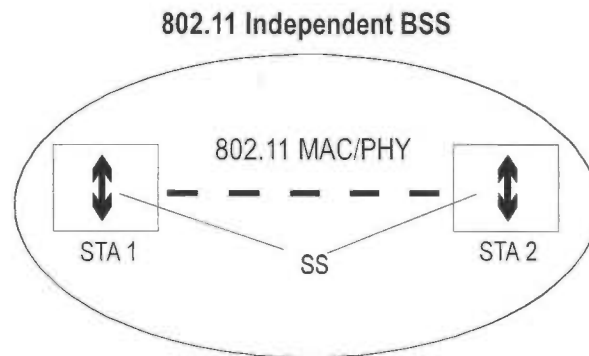


Figure 10—Logical architecture of an IBSS

Only the minimum two stations are shown in Figure 10. An IBSS may have an arbitrary number of members. In an IBSS, only Class 1 and Class 2 frames are allowed since there is no DS in an IBSS.

The services that apply to an IBSS are the SSs.

5.7 Message information contents that support the services

Each service is supported by one or more IEEE 802.11 messages. Information items are given by name; for corresponding values, see Clause 7.

5.7.1 Data

For a STA to send data to another STA, it sends a data message, as shown below:

Data messages

- Message type: Data
- Message subtype: Data
- Information items:
 - IEEE source address of message
 - IEEE destination address of message
 - BSS ID
- Direction of message: From STA to STA

5.7.2 Association

For a STA to associate, the association service causes the following messages to occur:

Association request

- Message type: Management
- Message subtype: Association request
- Information items:
 - IEEE address of the STA initiating the association
 - IEEE address of the AP with which the initiating station will associate
 - ESS ID
- Direction of message: From STA to AP

Association response

- Message type: Management
- Message subtype: Association response
- Information items:
 - Result of the requested association. This is an item with values “successful” and “unsuccessful.”
 - If the association is successful, the response shall include the association identifier (AID).
- Direction of message: From AP to STA

5.7.3 Reassociation

For a STA to reassociate, the reassociation service causes the following message to occur:

Reassociation request

- Message type: Management
- Message subtype: Reassociation request
- Information items:
 - IEEE address of the STA initiating the reassociation
 - IEEE address of the AP with which the initiating station will reassociate
 - IEEE address of the AP with which the initiating station is currently associated
 - ESS ID
- Direction of message:
 - From STA to AP (The AP with which the STA is requesting reassociation)

The address of the current AP is included for efficiency. The inclusion of the current AP address facilitates MAC reassociation to be independent of the DS implementation.

Reassociation response

- Message type: Management
- Message subtype: Reassociation response
- Information items:
 - Result of the requested reassociation. This is an item with values “successful” and “unsuccessful.”
 - If the reassociation is successful, the response shall include the AID.
- Direction of message: From AP to STA

5.7.4 Disassociation

For a STA to terminate an active association, the disassociation service causes the following message to occur:

Disassociation

- Message type: Management
- Message subtype: Disassociation
- Information items:
 - IEEE address of the station that is being disassociated. This shall be the broadcast address in the case of an AP disassociating with all associated stations.
 - IEEE address of the AP with which the station is currently associated.
- Direction of message: From STA to STA (e.g., STA to AP or AP to STA)

5.7.5 Privacy

For a STA to invoke the WEP privacy algorithm (as controlled by the related MIB attributes, see Clause 11), the privacy service causes MPDU encryption and sets the WEP frame header bit appropriately (see Clause 7).

5.7.6 Authentication

For a STA to authenticate with another STA, the authentication service causes one or more authentication management frames to be exchanged. The exact sequence of frames and their content is dependent on the authentication scheme invoked. For all authentication schemes, the authentication algorithm is identified within the management frame body.

In an IBSS environment, either station may be the initiating STA (STA 1). In an ESS environment, STA 1 is the mobile STA, and STA 2 is the AP.

Authentication (first frame of sequence)

- Message type: Management
- Message subtype: Authentication
- Information items:
 - Authentication algorithm identification
 - Station identity assertion
 - Authentication transaction sequence number
 - Authentication algorithm dependent information
- Direction of message: First frame in the transaction sequence is always from STA 1 to STA 2.

The first frame in an authentication sequence shall always be unencrypted.

Authentication (intermediate sequence frames)

- Message type: Management
- Message subtype: Authentication
- Information items:
 - Authentication algorithm identification
 - Authentication transaction sequence number
 - Authentication algorithm dependent information
- Direction of message:
 - Even transaction sequence numbers: From STA 2 to STA 1
 - Odd transaction sequence numbers: From STA 1 to STA 2

Authentication (final frame of sequence)

- Message type: Management
- Message subtype: Authentication
- Information items:
 - Authentication algorithm identification
 - Authentication transaction sequence number
 - Authentication algorithm dependent information
 - The result of the requested authentication. This is an item with values “successful” and “unsuccessful.”
- Direction of message: From STA 2 to STA 1

5.7.7 Deauthentication

For a STA to invalidate an active authentication, the following message is sent:

Deauthentication

- Message type: Management
- Message subtype: Deauthentication
- Information items:
 - IEEE address of the STA that is being deauthenticated
 - IEEE address of the STA with which the STA is currently authenticated
 - This shall be the broadcast address in the case of a STA deauthenticating all STAs currently authenticated.
- Direction of message: From STA to STA

5.8 Reference model

This standard presents the architectural view, emphasizing the separation of the system into two major parts: the MAC of the data link layer and the PHY. These layers are intended to correspond closely to the lowest

layers of the ISO/IEC basic reference model of Open Systems Interconnection (OSI) (ISO/IEC 7498-1: 1994⁵). The layers and sublayers described in this standard are shown in Figure 11.

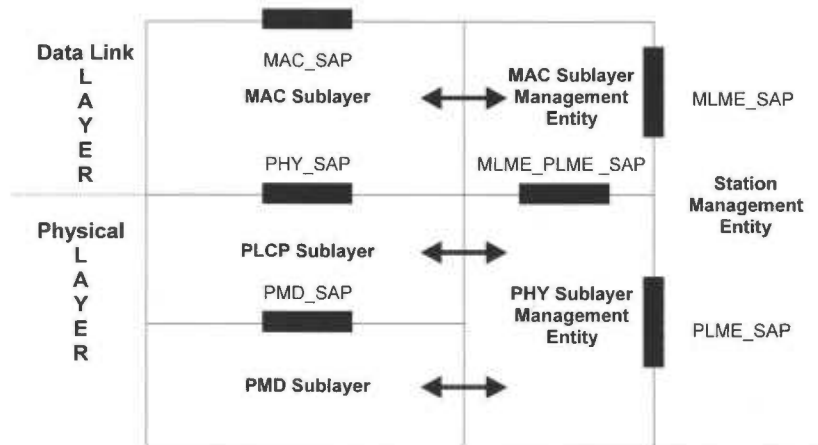


Figure 11—Portion of the ISO/IEC basic reference model covered in this standard

⁵Information on normative references can be found in Clause 2.

6. MAC service definition

6.1 Overview of MAC services

6.1.1 Asynchronous data service

This service provides peer LLC entities with the ability to exchange MAC service data units (MSDUs). To support this service, the local MAC uses the underlying PHY-level services to transport an MSDU to a peer MAC entity, where it will be delivered to the peer LLC. Such asynchronous MSDU transport is performed on a best-effort connectionless basis. There are no guarantees that the submitted MSDU will be delivered successfully. Broadcast and multicast transport is part of the asynchronous data service provided by the MAC. Due to the characteristics of the WM, broadcast and multicast MSDUs may experience a lower quality of service, compared to that of unicast MSDUs. All STAs will support the asynchronous data service. Because operation of certain functions of the MAC may cause reordering of some MSDUs, as discussed in more detail below, there are two service classes within the asynchronous data service. By selecting the desired service class, each LLC entity initiating the transfer of MSDUs is able to control whether MAC entities are or are not allowed to reorder those MSDUs.

6.1.2 Security services

Security services in IEEE 802.11 are provided by the authentication service and the WEP mechanism. The scope of the security services provided is limited to station-to-station data exchange. The privacy service offered by an IEEE 802.11 WEP implementation is the encryption of the MSDU. For the purposes of this standard, WEP is viewed as a logical service located within the MAC sublayer as shown in the reference model, Figure 11. Actual implementations of the WEP service are transparent to the LLC and other layers above the MAC sublayer.

The security services provided by the WEP in IEEE 802.11 are as follows:

- a) Confidentiality;
- b) Authentication; and
- c) Access control in conjunction with layer management.

During the authentication exchange, parties A and B exchange authentication information as described in Clause 8.

The MAC sublayer security services provided by WEP rely on information from non-layer 2 management or system entities. Management entities communicate information to WEP through a set of MIB attributes.

6.1.3 MSDU ordering

The services provided by the MAC sublayer permit, and may in certain cases require, the reordering of MSDUs. The MAC does not intentionally reorder MSDUs except as may be necessary to improve the likelihood of successful delivery based on the current operational (“power management”) mode of the designated recipient station(s). The sole effect of this reordering (if any), for the set of MSDUs received at the MAC service interface of any single station, is a change in the delivery order of broadcast and multicast MSDUs, relative to directed MSDUs, originating from a single source station address. If a higher-layer protocol using the asynchronous data service cannot tolerate this possible reordering, the optional StrictlyOrdered service class should be used. MSDUs transferred between any pair of stations using the StrictlyOrdered service class are not subject to the relative reordering that is possible when the ReorderableMulticast service class is used. However, the desire to receive MSDUs sent using the StrictlyOrdered service class at a station precludes simultaneous use of the MAC power management facilities at that station.

In order for the MAC to operate properly, the DS must meet the requirements of ISO/IEC 15802-1: 1995.

Operational restrictions that ensure the appropriate ordering of MSDUs are specified in 9.8.

6.2 Detailed service specification

6.2.1 MAC data services

The IEEE 802.11 MAC supports the following service primitives as defined in ISO/IEC 8802-2: 1998:

- MA-UNITDATA.request
- MA-UNITDATA.indication
- MA-UNITDATA-STATUS.indication

The LLC definitions of the primitives and specify parameter value restrictions imposed by IEEE 802.11 are given in 6.2.1.1 through 6.2.1.3.

6.2.1.1 MA-UNITDATA.request

6.2.1.1.1 Function

This primitive requests a transfer of an MSDU from a local LLC sublayer entity to a single peer LLC sublayer entity, or multiple peer LLC sublayer entities in the case of group addresses.

6.2.1.1.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MA-UNITDATA.request    (
                        source address,
                        destination address,
                        routing information,
                        data,
                        priority,
                        service class
                        )
```

The source address (SA) parameter specifies an individual MAC sublayer address of the sublayer entity to which the MSDU is being transferred.

The destination address (DA) parameter specifies either an individual or a group MAC sublayer entity address.

The routing information parameter specifies the route desired for the data transfer (a null value indicates source routing is not to be used). For IEEE 802.11, the routing information parameter must be null.

The data parameter specifies the MSDU to be transmitted by the MAC sublayer entity. For IEEE 802.11, the length of the MSDU must be less than or equal to 2304 octets.

The priority parameter specifies the priority desired for the data unit transfer. IEEE 802.11 allows two values: Contention or ContentionFree.

The service class parameter specifies the service class desired for the data unit transfer. IEEE 802.11 allows two values: ReorderableMulticast or StrictlyOrdered.

6.2.1.1.3 When generated

This primitive is generated by the LLC sublayer entity whenever an MSDU is to be transferred to a peer LLC sublayer entity or entities.

6.2.1.1.4 Effect of receipt

The receipt of this primitive causes the MAC sublayer entity to append all MAC specified fields, including DA, SA, and all fields that are unique to IEEE 802.11, and pass the properly formatted frame to the lower layers for transfer to a peer MAC sublayer entity or entities.

6.2.1.2 MA-UNITDATA.indication

6.2.1.2.1 Function

This primitive defines the transfer of an MSDU from the MAC sublayer entity to the LLC sublayer entity, or entities in the case of group addresses. In the absence of error, the contents of the data parameter are logically complete and unchanged relative to the data parameter in the associated MA-UNITDATA.request primitive.

6.2.1.2.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MA-UNITDATA.indication (
    source address,
    destination address,
    routing information,
    data,
    reception status,
    priority,
    service class
)
```

The SA parameter is an individual address as specified by the SA field of the incoming frame.

The DA parameter is either an individual or a group address as specified by the DA field of the incoming frame.

The routing information parameter specifies the route that was used for the data transfer. IEEE 802.11 will always set this field to null.

The data parameter specifies the MSDU as received by the local MAC entity.

The reception status parameter indicates the success or failure of the received frame for those frames that IEEE 802.11 reports via an MA-UNITDATA.indication. This MAC only reports "success" when all failures of reception are discarded without generating MA-UNITDATA.indication.

The priority parameter specifies the receive processing priority that was used for the data unit transfer. IEEE 802.11 allows two values: Contention or ContentionFree.

The service class parameter specifies the receive service class that was used for the data unit transfer. IEEE 802.11 allows two values: `ReorderableMulticast` or `StrictlyOrdered`.

6.2.1.2.3 When generated

The `MA-UNITDATA.indication` primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities to indicate the arrival of a frame at the local MAC sublayer entity. Frames are reported only if they are validly formatted at the MAC sublayer, received without error, received with valid (or null) WEP encryption, and their destination address designates the local MAC sublayer entity.

6.2.1.2.4 Effect of receipt

The effect of receipt of this primitive by the LLC sublayer is dependent on the validity and content of the frame.

6.2.1.3 MA-UNITDATA-STATUS.indication

6.2.1.3.1 Function

This primitive has local significance and provides the LLC sublayer with status information for the corresponding preceding `MA-UNITDATA.request` primitive.

6.2.1.3.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MA-UNITDATA-STATUS.indication (
    source address,
    destination address,
    transmission status,
    provided priority,
    provided service class
)
```

The SA parameter is an individual MAC sublayer entity address as specified in the associated `MA-UNITDATA.request` primitive.

The DA parameter is either an individual or group MAC sublayer entity address as specified in the associated `MA-UNITDATA.request` primitive.

The transmission status parameter will be used to pass status information back to the local requesting LLC sublayer entity. IEEE 802.11 specifies the following values for transmission status:

- a) Successful;
- b) Undeliverable (for unacknowledged directed MSDUs when the `aShortRetryMax` or `aLongRetryMax` retry limit would otherwise be exceeded);
- c) Excessive data length;
- d) Non-null source routing;
- e) Unsupported priority (for priorities other than `Contention` or `ContentionFree`);
- f) Unsupported service class (for service classes other than `ReorderableMulticast` or `StrictlyOrdered`);
- g) Unavailable priority (for `ContentionFree` when no point coordinator is available, in which case the MSDU is transmitted with a provided priority of `Contention`);

- h) Unavailable service class (for StrictlyOrdered service when the station's power management mode is other than "active");
- i) Undeliverable (TransmitMSDUTimer reached aMaxTransmitMSDULifetime before successful delivery);
- j) Undeliverable (no BSS available);
- k) Undeliverable (cannot encrypt with a null key).

The provided priority parameter specifies the priority that was used for the associated data unit transfer (Contention or ContentionFree).

The provided service class parameter specifies the class of service used for the associated data unit transfer (ReorderableMulticast or StrictlyOrdered).

6.2.1.3.3 When generated

The MA-UNITDATA-STATUS.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity to indicate the status of the service provided for the corresponding MA-UNITDATA.request primitive.

6.2.1.3.4 Effect of receipt

The effect of receipt of this primitive by the LLC sublayer is dependent upon the type of operation employed by the LLC sublayer entity.

7. Frame formats

The format of the MAC frames is specified in this clause. All stations shall be able to properly construct frames for transmission and decode frames upon reception, as specified in this clause.

7.1 MAC frame formats

Each frame consists of the following basic components:

- a) A *MAC header*, which comprises frame control, duration, address, and sequence control information;
- b) A variable length *frame body*, which contains information specific to the frame *type*;
- c) A *frame check sequence (FCS)*, which contains an IEEE 32-bit cyclic redundancy code (CRC).

7.1.1 Conventions

The MAC protocol data units (MPDUs) or frames in the MAC sublayer are described as a sequence of fields in specific order. Each figure in Clause 7 depicts the fields/subfields as they appear in the MAC frame and in the order in which they are passed to the physical layer convergence protocol (PLCP), from left to right.

In figures, all bits within fields are numbered, from 0 to k , where the length of the field is $k + 1$ bit. The octet boundaries within a field can be obtained by taking the bit numbers of the field modulo 8. Octets within numeric fields that are longer than a single octet are depicted in increasing order of significance, from lowest numbered bit to highest numbered bit. The octets in fields longer than a single octet are sent to the PLCP in order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits.

Any field containing a CRC is an exception to this convention and is transmitted commencing with the coefficient of the highest-order term.

MAC addresses are assigned as ordered sequences of bits. The Individual/Group bit is always transferred first and is bit 0 of the first octet.

Values specified in decimal are coded in natural binary unless otherwise stated. The values in Table 1 are in binary, with the bit assignments shown in the table. Values in other tables are shown in decimal notation.

Reserved fields and subfields are set to 0 upon transmission and are ignored upon reception.

7.1.2 General frame format

The MAC frame format comprises a set of fields that occur in a fixed order in all frames. Figure 12 depicts the general MAC frame format. The fields Address 2, Address 3, Sequence Control, Address 4, and Frame Body are only present in certain frame types. Each field is defined in 7.1.3. The format of each of the individual frame types is defined in 7.2.

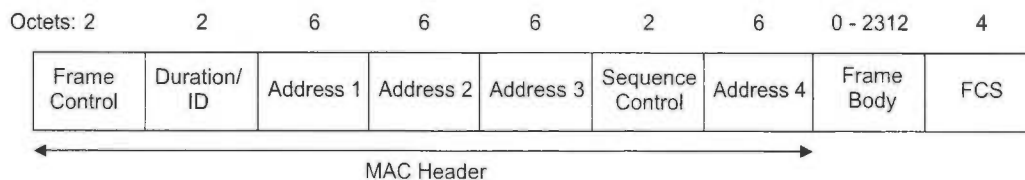


Figure 12—MAC frame format

7.1.3 Frame fields

7.1.3.1 Frame Control field

The Frame Control field consists of the following subfields: Protocol Version, Type, Subtype, To DS, From DS, More Fragments, Retry, Power Management, More Data, Wired Equivalent Privacy (WEP), and Order. The format of the Frame Control field is illustrated in Figure 13.

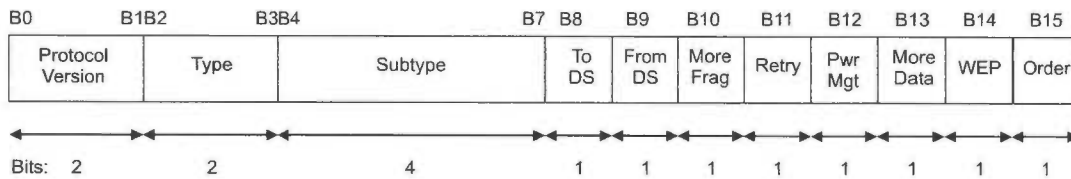


Figure 13—Frame Control field

7.1.3.1.1 Protocol Version field

The Protocol Version field is 2 bits in length and is invariant in size and placement across all revisions of this standard. For this standard, the value of the protocol version is 0. All other values are reserved. The revision level will be incremented only when a fundamental incompatibility exists between a new revision and the prior edition of the standard. A device that receives a frame with a higher revision level than it supports will discard the frame without indication to the sending station or to LLC.

7.1.3.1.2 Type and Subtype fields

The Type field is 2 bits in length, and the Subtype field is 4 bits in length. The Type and Subtype fields together identify the function of the frame. There are three frame types: control, data, and management. Each of the frame types have several defined subtypes. Table 1 defines the valid combinations of type and subtype.

7.1.3.1.3 To DS field

The To DS field is 1 bit in length and is set to 1 in data type frames destined for the DS. This includes all data type frames sent by STAs associated with an AP. The To DS field is set to 0 in all other frames.

7.1.3.1.4 From DS field

The From DS field is 1 bit in length and is set to 1 in data type frames exiting the DS. It is set to 0 in all other frames.

The permitted To/From DS bit combinations and their meanings are given in Table 2.

7.1.3.1.5 More Fragments field

The More Fragments field is 1 bit in length and is set to 1 in all data or management type frames that have another fragment of the current MSDU or current MMPDU to follow. It is set to 0 in all other frames.

Table 1—Valid type and subtype combinations

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0000	Association request
00	Management	0001	Association response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	0110–0111	Reserved
00	Management	1000	Beacon
00	Management	1001	Announcement traffic indication message (ATIM)
00	Management	1010	Disassociation
00	Management	1011	Authentication
00	Management	1100	Deauthentication
00	Management	1101–1111	Reserved
01	Control	0000–1001	Reserved
01	Control	1010	Power Save (PS)-Poll
01	Control	1011	Request To Send (RTS)
01	Control	1100	Clear To Send (CTS)
01	Control	1101	Acknowledgment (ACK)
01	Control	1110	Contention-Free (CF)-End
01	Control	1111	CF-End + CF-Ack
10	Data	0000	Data
10	Data	0001	Data + CF-Ack
10	Data	0010	Data + CF-Poll
10	Data	0011	Data + CF-Ack + CF-Poll
10	Data	0100	Null function (no data)
10	Data	0101	CF-Ack (no data)
10	Data	0110	CF-Poll (no data)
10	Data	0111	CF-Ack + CF-Poll (no data)
10	Data	1000–1111	Reserved
11	Reserved	0000–1111	Reserved

Table 2—To/From DS combinations in data type frames

To/From DS values	Meaning
To DS = 0 From DS = 0	A data frame direct from one STA to another STA within the same IBSS, as well as all management and control type frames.
To DS = 1 From DS = 0	Data frame destined for the DS.
To DS = 0 From DS = 1	Data frame exiting the DS.
To DS = 1 From DS = 1	Wireless distribution system (WDS) frame being distributed from one AP to another AP.

7.1.3.1.6 Retry field

The Retry field is 1 bit in length and is set to 1 in any data or management type frame that is a retransmission of an earlier frame. It is set to 0 in all other frames. A receiving station uses this indication to aid in the process of eliminating duplicate frames.

7.1.3.1.7 Power Management field

The Power Management field is 1 bit in length and is used to indicate the power management mode of a STA. The value of this field remains constant in each frame from a particular STA within a frame exchange sequence defined in 9.7. The value indicates the mode in which the station will be after the successful completion of the frame exchange sequence.

A value of 1 indicates that the STA will be in power-save mode. A value of 0 indicates that the STA will be in active mode. This field is always set to 0 in frames transmitted by an AP.

7.1.3.1.8 More Data field

The More Data field is 1 bit in length and is used to indicate to a STA in power-save mode that more MSDUs, or MMPDUs are buffered for that STA at the AP. The More Data field is valid in directed data or management type frames transmitted by an AP to an STA in power-save mode. A value of 1 indicates that at least one additional buffered MSDU, or MMPDU, is present for the same STA.

The More Data field may be set to 1 in directed data type frames transmitted by a contention-free (CF)-Pollable STA to the point coordinator (PC) in response to a CF-Poll to indicate that the STA has at least one additional buffered MSDU available for transmission in response to a subsequent CF-Poll.

The More Data field is set to 0 in all other directed frames.

The More Data field is set to 1 in broadcast/multicast frames transmitted by the AP, when additional broadcast/multicast MSDUs, or MMPDUs, remain to be transmitted by the AP during this beacon interval. The More Data field is set to 0 in broadcast/multicast frames transmitted by the AP when no more broadcast/multicast MSDUs, or MMPDUs, remain to be transmitted by the AP during this beacon interval and in all broadcast/multicast frames transmitted by non-AP stations.

7.1.3.1.9 WEP field

The WEP field is 1 bit in length. It is set to 1 if the Frame Body field contains information that has been processed by the WEP algorithm. The WEP field is only set to 1 within frames of type Data and frames of

type Management, subtype Authentication. The WEP field is set to 0 in all other frames. When the WEP bit is set to 1, the Frame Body field is expanded as defined in 8.2.5.

7.1.3.1.10 Order field

The Order field is 1 bit in length and is set to 1 in any data type frame that contains an MSDU, or fragment thereof, which is being transferred using the StrictlyOrdered service class. This field is set to 0 in all other frames.

7.1.3.2 Duration/ID field

The Duration/ID field is 16 bits in length. The contents of this field are as follows:

- a) In control type frames of subtype Power Save (PS)-Poll, the Duration/ID field carries the association identity (AID) of the station that transmitted the frame in the 14 least significant bits (lsb), with the 2 most significant bits (msb) both set to 1. The value of the AID is in the range 1–2007.
- b) In all other frames, the Duration/ID field contains a duration value as defined for each frame type in 7.2. For frames transmitted during the contention-free period (CFP), the duration field is set to 32 768.

Whenever the contents of the Duration/ID field are less than 32 768, the duration value is used to update the network allocation vector (NAV) according to the procedures defined in Clause 9.

The encoding of the Duration/ID field is given in Table 3.

Table 3—Duration/ID field encoding

Bit 15	Bit 14	Bits 13–0	Usage
0		0–32 767	Duration
1	0	0	Fixed value within frames transmitted during the CFP
1	0	1–16 383	Reserved
1	1	0	Reserved
1	1	1–2 007	AID in PS-Poll frames
1	1	2 008–16 383	Reserved

7.1.3.3 Address fields

There are four address fields in the MAC frame format. These fields are used to indicate the BSSID, source address, destination address, transmitting station address, and receiving station address. The usage of the four address fields in each frame type is indicated by the abbreviations BSSID, DA, SA, RA, and TA, indicating basic service set identifier (BSSID), Destination Address, Source Address, Receiver Address, and Transmitter Address, respectively. Certain frames may not contain some of the address fields.

Certain address field usage is specified by the relative position of the address field (1–4) within the MAC header, independent of the type of address present in that field. For example, receiver address matching is always performed on the contents of the Address 1 field in received frames, and the receiver address of CTS and ACK frames is always obtained from the Address 2 field in the corresponding RTS frame, or from the frame being acknowledged.

7.1.3.3.1 Address representation

Each Address field contains a 48-bit address as defined in 5.2 of IEEE Std 802-1990.

7.1.3.3.2 Address designation

A MAC sublayer address is one of the following two types:

- a) *Individual address.* The address associated with a particular station on the network.
- b) *Group address.* A multideestination address, associated with one or more stations on a given network. The two kinds of group addresses are as follows:
 - 1) *Multicast-group address.* An address associated by higher-level convention with a group of logically related stations.
 - 2) *Broadcast address.* A distinguished, predefined multicast address that always denotes the set of all stations on a given LAN. All 1s in the Destination Address field are interpreted to be the broadcast address. This group is predefined for each communication medium to consist of all stations actively connected to that medium; it is used to broadcast to all the active stations on that medium. All stations are able to recognize the broadcast address. It is not necessary that a station be capable of generating the broadcast address.

The address space is also partitioned into locally administered and universal (globally administered) addresses. The nature of a body and the procedures by which it administers these universal (globally administered) addresses is beyond the scope of this standard. See IEEE Std 802-1990 for more information.

7.1.3.3.3 BSSID field

The BSSID field is a 48-bit field of the same format as an IEEE 802 MAC address. This field uniquely identifies each BSS. The value of this field, in an infrastructure BSS, is the MAC address currently in use by the STA in the AP of the BSS.

The value of this field in an IBSS is a locally administered IEEE MAC address formed from a 46-bit random number generated according to the procedure defined in 11.1.3. The individual/group bit of the address is set to 0. The universal/local bit of the address is set to 1. This mechanism is used to provide a high probability of selecting a unique BSSID.

The value of all 1s is used to indicate the broadcast BSSID. A broadcast BSSID may only be used in the BSSID field of management frames of subtype probe request.

7.1.3.3.4 Destination Address (DA) field

The DA field contains an IEEE MAC individual or group address that identifies the MAC entity or entities intended as the final recipient(s) of the MSDU (or fragment thereof) contained in the frame body field.

7.1.3.3.5 Source Address (SA) field

The SA field contains an IEEE MAC individual address that identifies the MAC entity from which the transfer of the MSDU (or fragment thereof) contained in the frame body field was initiated. The individual/group bit is always transmitted as a zero in the source address.

7.1.3.3.6 Receiver Address (RA) field

The RA field contains an IEEE MAC individual or group address that identifies the intended immediate recipient STA(s), on the WM, for the information contained in the frame body field.

7.1.3.3.7 Transmitter Address (TA) field

The TA field contains an IEEE MAC individual address that identifies the STA that has transmitted, onto the WM, the MPDU contained in the frame body field. The Individual/Group bit is always transmitted as a zero in the transmitter address.

7.1.3.4 Sequence Control field

The Sequence Control field is 16 bits in length and consists of two subfields, the Sequence Number and the Fragment Number. The format of the Sequence Control field is illustrated in Figure 14.

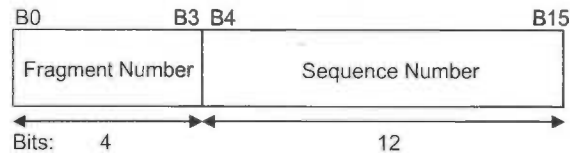


Figure 14—Sequence Control field

7.1.3.4.1 Sequence Number field

The Sequence Number field is a 12-bit field indicating the sequence number of an MSDU or MMPDU. Each MSDU or MMPDU transmitted by a STA is assigned a sequence number. Sequence numbers are assigned from a single modulo 4096 counter, starting at 0 and incrementing by 1 for each MSDU or MMPDU. Each fragment of an MSDU or MMPDU contains the assigned sequence number. The sequence number remains constant in all retransmissions of an MSDU, MMPDU, or fragment thereof.

7.1.3.4.2 Fragment Number field

The Fragment Number field is a 4-bit field indicating the number of each fragment of an MSDU or MMPDU. The fragment number is set to zero in the first or only fragment of an MSDU or MMPDU and is incremented by one for each successive fragment of that MSDU or MMPDU. The fragment number remains constant in all retransmissions of the fragment.

7.1.3.5 Frame Body field

The Frame Body is a variable length field that contains information specific to individual frame types and subtypes. The minimum frame body is 0 octets. The maximum length frame body is defined by the maximum length (MSDU + ICV + IV), where ICV and IV are the WEP fields defined in 8.2.5.

7.1.3.6 FCS field

The FCS field is a 32-bit field containing a 32-bit CRC. The FCS is calculated over all the fields of the MAC header and the Frame Body field. These are referred to as the *calculation fields*.

The FCS is calculated using the following standard generator polynomial of degree 32:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

The FCS is the 1's complement of the sum (modulo 2) of the following:

- a) The remainder of $x^k \times (x^{31} + x^{30} + x^{29} + \dots + x^2 + x + 1)$ divided (modulo 2) by $G(x)$, where k is the number of bits in the calculation fields, and

- b) The remainder after multiplication of the contents (treated as a polynomial) of the calculation fields by x^{32} and then division by $G(x)$.

The FCS field is transmitted commencing with the coefficient of the highest-order term.

As a typical implementation, at the transmitter, the initial remainder of the division is preset to all 1's and is then modified by division of the calculation fields by the generator polynomial $G(x)$. The 1's complement of this remainder is transmitted, with the highest-order bit first, as the FCS field.

At the receiver, the initial remainder is preset to all 1's and the serial incoming bits of the calculation fields and FCS, when divided by $G(x)$, results in the absence of transmission errors, in a unique nonzero remainder value. The unique remainder value is the polynomial:

$$x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$$

7.2 Format of individual frame types

7.2.1 Control frames

In the following descriptions, "immediately previous" frame means a frame whose reception concluded within the prior short interframe space (SIFS) interval.

The subfields within the Frame Control field of control frames are set as illustrated in Figure 15.

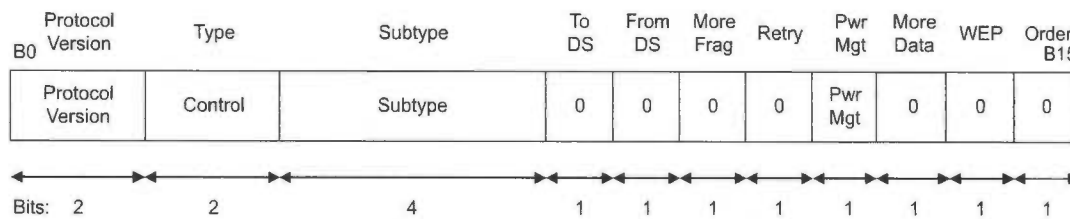


Figure 15—Frame Control field subfield values within control frames

7.2.1.1 Request To Send (RTS) frame format

The frame format for the RTS frame is as defined in Figure 16.

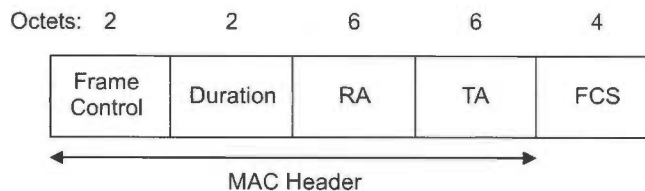


Figure 16—RTS frame

The RA of the RTS frame is the address of the STA, on the WM, that is the intended immediate recipient of the pending directed data or management frame.

The TA is the address of the STA transmitting the RTS frame.

The duration value is the time, in microseconds, required to transmit the pending data or management frame, plus one CTS frame, plus one ACK frame, plus three SIFS intervals. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

7.2.1.2 Clear To Send (CTS) frame format

The frame format for the CTS frame is as defined in Figure 17.

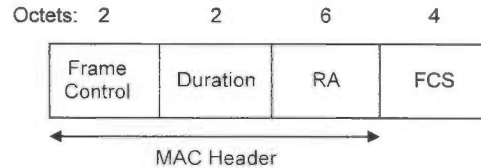


Figure 17—CTS frame

The RA of the CTS frame is copied from the TA field of the immediately previous RTS frame to which the CTS is a response.

The duration value is the value obtained from the Duration field of the immediately previous RTS frame, minus the time, in microseconds, required to transmit the CTS frame and its SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

7.2.1.3 Acknowledgment (ACK) frame format

The frame format for the ACK frame is as defined in Figure 18.

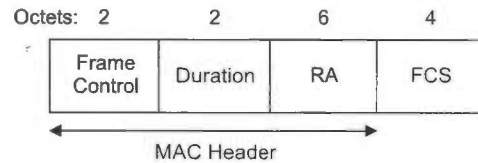


Figure 18—ACK frame

The RA of the ACK frame is copied from the Address 2 field of the immediately previous directed data, management, or PS-Poll control frame.

If the More Fragment bit was set to 0 in the Frame Control field of the immediately previous directed data or management frame, the duration value is set to 0. If the More Fragment bit was set to 1 in the Frame Control field of the immediately previous directed data or management frame, the duration value is the value obtained from the Duration field of the immediately previous data or management frame, minus the time, in microseconds, required to transmit the ACK frame and its SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

7.2.1.4 Power-Save Poll (PS-Poll) frame format

The frame format for the PS-Poll frame is as defined in Figure 19.

The BSSID is the address of the STA contained in the AP. The TA is the address of the STA transmitting the frame. The AID is the value assigned to the STA transmitting the frame by the AP in the association response frame that established that STA's current association.

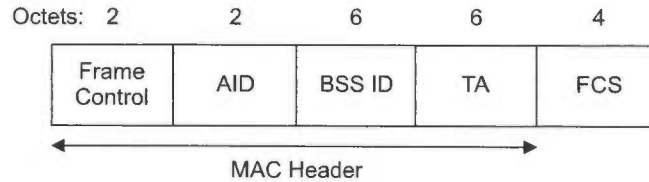


Figure 19—PS-Poll frame

The AID value always has its two most significant bits each set to 1. All STAs, upon receipt of a PS-Poll frame, update their NAV settings as appropriate under the coordination function rules using a duration value equal to the time, in microseconds, required to transmit one ACK frame plus one SIFS interval.

7.2.1.5 CF-End frame format

The frame format for the CF-End frame is as defined in Figure 20.

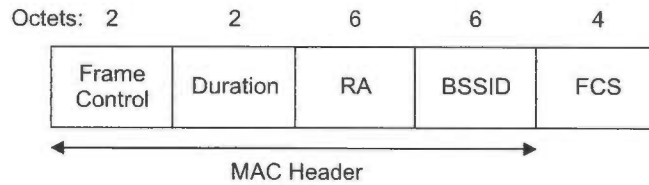


Figure 20—CF-End frame

The BSSID is the address of the STA contained in the AP. The RA is the broadcast group address.

The Duration field is set to 0.

7.2.1.6 CF-End + CF-Ack frame format

The frame format for the contention-free-end acknowledge (CF-End + CF-Ack) frame is as defined in Figure 21.

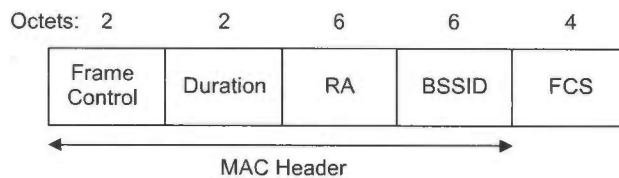


Figure 21—CF-End + CF-Ack Frame

The BSSID is the address of the STA contained in the AP. The RA is the broadcast group address.

The Duration field is set to 0.

7.2.2 Data frames

The frame format for a Data frame is independent of subtype and is as defined in Figure 22.

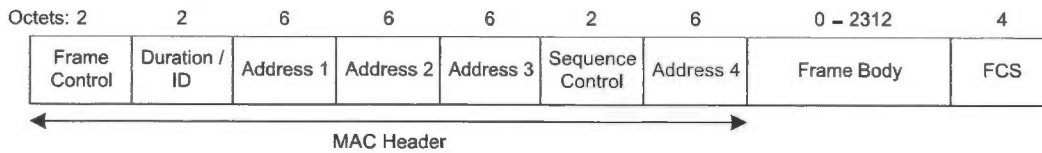


Figure 22—Data frame

The content of the Address fields of the data frame is dependent upon the values of the To DS and From DS bits and is defined in Table 4. Where the content of a field is shown as not applicable (N/A), the field is omitted. Note that Address 1 always holds the receiver address of the intended receiver (or, in the case of multi-cast frames, receivers), and that Address 2 always holds the address of the station that is transmitting the frame.

Table 4 —Address field contents

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	DA	SA	BSSID	N/A
0	1	DA	BSSID	SA	N/A
1	0	BSSID	SA	DA	N/A
1	1	RA	TA	DA	SA

A station uses the contents of the Address 1 field to perform address matching for receive decisions. In cases where the Address 1 field contains a group address, the BSSID also is validated to ensure that the broadcast or multicast originated in the same BSS.

A station uses the contents of the Address 2 field to direct the acknowledgment if an acknowledgment is necessary.

The DA is the destination of the MSDU (or fragment thereof) in the frame body field.

The SA is the address of the MAC entity that initiated the MSDU (or fragment thereof) in the frame body field.

The RA is the address of the STA contained in the AP in the wireless distribution system that is the next immediate intended recipient of the frame.

The TA is the address of the STA contained in the AP in the wireless distribution system that is transmitting the frame.

The BSSID of the Data frame is determined as follows:

- a) If the station is an AP or is associated with an AP, the BSSID is the address currently in use by the STA contained in the AP.
- b) If the station is a member of an IBSS, the BSSID is the BSSID of the IBSS.

The frame body consists of the MSDU or a fragment thereof, and a WEP IV and ICV (if and only if the WEP subfield in the frame control field is set to 1). The frame body is null (0 octets in length) in data frames of Subtype Null function (no data), CF-Ack (no data), CF-Poll (no data), and CF-Ack+CF-Poll (no data).

Within all data type frames sent during the CFP, the Duration field is set to the value 32 768. Within all data type frames sent during the contention period, the Duration field is set according to the following rules:

- If the Address 1 field contains a group address, the duration value is set to 0.
- If the More Fragments bit is set to 0 in the Frame Control field of a frame and the Address 1 field contains an individual address, the duration value is set to the time, in microseconds, required to transmit one ACK frame, plus one SIFS interval.
- If the More Fragments bit is set to 1 in the Frame Control field of a frame, and the Address 1 field contains an individual address, the duration value is set to the time, in microseconds, required to transmit the next fragment of this data frame, plus two ACK frames, plus three SIFS intervals.

The duration value calculation for the data frame is based on the rules in 9.6 that determine the data rate at which the control frames in the frame exchange sequence are transmitted. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. All stations process Duration field values less than or equal to 32 767 from valid data frames to update their NAV settings as appropriate under the coordination function rules.

7.2.3 Management frames

The frame format for a Management frame is independent of frame subtype and is as defined in Figure 23.

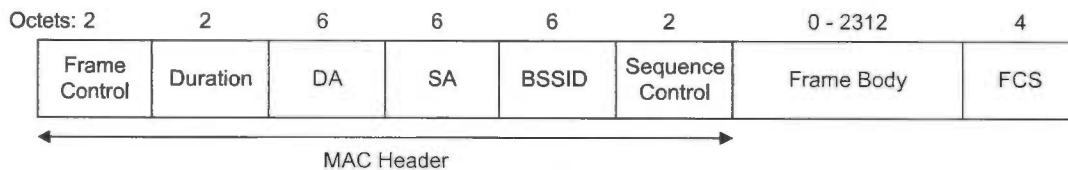


Figure 23—Management frame format

A STA uses the contents of the Address 1 field to perform the address matching for receive decisions. In the case where the Address 1 field contains a group address and the frame type is other than Beacon, the BSSID also is validated to ensure that the broadcast or multicast originated in the same BSS. If the frame type is Beacon, other address matching rules apply, as specified in 11.1.2.3.

The address fields for management frames do not vary by frame subtype.

The BSSID of the management frame is determined as follows:

- a) If the station is an AP or is associated with an AP, the BSSID is the address currently in use by the STA contained in the AP.
- b) If the station is a member of an IBSS, the BSSID is the BSSID of the IBSS.
- c) In Management frames of subtype Probe Request, the BSSID is either a specific BSSID, or the broadcast BSSID as defined in the procedures specified in Clause 10.

The DA is the destination of the frame.

The SA is the address of the station transmitting the frame.

Within all management type frames sent during the CFP, the Duration field is set to the value 32 768. Within all management type frames sent during the contention period, the Duration field is set according to the following rules:

- If the DA field contains a group address, the duration value is set to 0.
- If the More Fragments bit is set to 0 in the Frame Control field of a frame and the DA contains an individual address, the duration value is set to the time, in microseconds, required to transmit one ACK frame, plus one SIFS interval.
- If the More Fragments bit is set to 1 in the Frame Control field of a frame, and the DA contains an individual address, the duration value is the time, in microseconds, required to transmit the next fragment of this management frame, plus two ACK frames, plus three SIFS intervals.

The duration value calculation for the management frame is based on the rules in 9.6 that determine the data rate at which the control frames in the frame exchange sequence are transmitted. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. All stations process Duration field values less than or equal to 32 767 from valid management frames to update their NAV settings as appropriate under the coordination function rules.

The frame body consists of the fixed fields and information elements defined for each management frame subtype. All fixed fields and information elements are mandatory unless stated otherwise, and they can appear only in the specified order. Stations encountering an element type they do not understand ignore that element. Element type codes not explicitly defined in this standard are reserved, and do not appear in any frames.

7.2.3.1 Beacon frame format

The frame body of a management frame of subtype Beacon contains the information shown in Table 5.

Table 5—Beacon frame body

Order	Information	Notes
1	Timestamp	
2	Beacon interval	
3	Capability information	
4	SSID	
5	Supported rates	
6	FH Parameter Set	The FH Parameter Set information element is present within Beacon frames generated by STAs using frequency-hopping PHYs.
7	DS Parameter Set	The DS Parameter Set information element is present within Beacon frames generated by STAs using direct sequence PHYs.
8	CF Parameter Set	The CF Parameter Set information element is only present within Beacon frames generated by APs supporting a PCF.
9	IBSS Parameter Set	The IBSS Parameter Set information element is only present within Beacon frames generated by STAs in an IBSS.
10	TIM	The TIM information element is only present within Beacon frames generated by APs.

7.2.3.2 IBSS Announcement Traffic Indication Message (ATIM) frame format

The frame body of a management frame of subtype ATIM is null.

7.2.3.3 Disassociation frame format

The frame body of a management frame of subtype Disassociation contains the information shown in Table 6.

Table 6—Disassociation frame body

Order	Information
1	Reason code

7.2.3.4 Association Request frame format

The frame body of a management frame of subtype Association Request contains the information shown in Table 7.

Table 7 —Association Request frame body

Order	Information
1	Capability information
2	Listen interval
3	SSID
4	Supported rates

7.2.3.5 Association Response frame format

The frame body of a management frame of subtype Association Response contains the information shown in Table 8.

Table 8—Association Response frame body

Order	Information
1	Capability information
2	Status code
3	Association ID (AID)
4	Supported rates

7.2.3.6 Reassociation Request frame format

The frame body of a management frame of subtype Reassociation Request contains the information shown in Table 9.

Table 9—Reassociation Request frame body

Order	Information
1	Capability information
2	Listen interval
3	Current AP address
4	SSID
5	Supported rates

7.2.3.7 Reassociation Response frame format

The frame body of a management frame of subtype Reassociation Response contains the information shown in Table 10.

Table 10—Reassociation Response frame body

Order	Information
1	Capability information
2	Status code
3	Association ID (AID)
4	Supported rates

7.2.3.8 Probe Request frame format

The frame body of a management frame of subtype Probe Request contains the information shown in Table 11.

Table 11—Probe Request frame body

Order	Information
1	SSID
2	Supported rates

7.2.3.9 Probe Response frame format

The frame body of a management frame of subtype Probe Response contains the information shown in Table 12.

Table 12—Probe Response frame body

Order	Information	Notes
1	Timestamp	
2	Beacon interval	
3	Capability information	
4	SSID	
5	Supported rates	
6	FH Parameter Set	The FH Parameter Set information element is present within Probe Response frames generated by STAs using frequency-hopping PHYs.
7	DS Parameter Set	The DS Parameter Set information element is present within Probe Response frames generated by STAs using direct sequence PHYs.
8	CF Parameter Set	The CF Parameter Set information element is only present within Probe Response frames generated by APs supporting a PCF.
9	IBSS Parameter Set	The IBSS Parameter Set information element is only present within Probe Response frames generated by STAs in an IBSS.

7.2.3.10 Authentication frame format

The frame body of a management frame of subtype Authentication contains the information shown in Table 13.

Table 13—Authentication frame body

Order	Information	Notes
1	Authentication algorithm number	
2	Authentication transaction sequence number	
3	Status code	The status code information is reserved and set to 0 in certain Authentication frames as defined in Table 14.
4	Challenge text	The challenge text information is only present in certain Authentication frames as defined in Table 14.

Table 14—Presence of challenge text information

Authentication algorithm	Authentication transaction sequence no.	Status code	Challenge text
Open System	1	Reserved	Not present
Open System	2	Status	Not present
Shared Key	1	Reserved	Not present
Shared Key	2	Status	Present
Shared Key	3	Reserved	Present
Shared Key	4	Status	Not present

7.2.3.11 Deauthentication

The frame body of a management frame of subtype Deauthentication contains the information shown in Table 15.

Table 15—Deauthentication frame body

Order	Information
1	Reason code

7.3 Management frame body components

Within management frames, fixed-length mandatory frame body components are defined as fixed fields; variable length mandatory and all optional frame body components are defined as information elements.

7.3.1 Fixed fields

7.3.1.1 Authentication Algorithm Number field

The Authentication Algorithm Number field indicates a single authentication algorithm. The length of the Authentication Algorithm Number field is 2 octets. The Authentication Algorithm Number field is illustrated in Figure 24. The following values are defined for authentication algorithm number:

- Authentication algorithm number = 0: Open System
- Authentication algorithm number = 1: Shared Key
- All other values of authentication number are reserved.

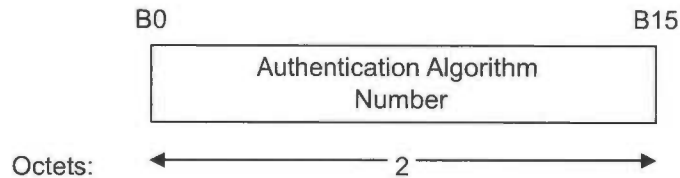


Figure 24—Authentication Algorithm Number fixed field

7.3.1.2 Authentication Transaction Sequence Number field

The Authentication Transaction Sequence Number field indicates the current state of progress through a multistep transaction. The length of the Authentication Transaction Sequence Number field is 2 octets. The Authentication Transaction Sequence Number field is illustrated in Figure 25.

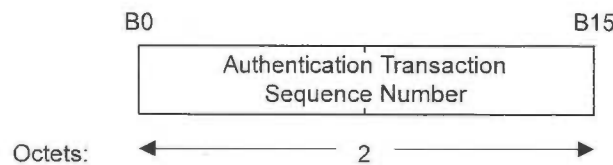


Figure 25—Authentication Transaction Sequence Number fixed field

7.3.1.3 Beacon Interval field

The Beacon Interval field represents the number of time units (TUs) between target beacon transmission times (TBTTs). The length of the Beacon Interval field is 2 octets. The Beacon Interval field is illustrated in Figure 26.

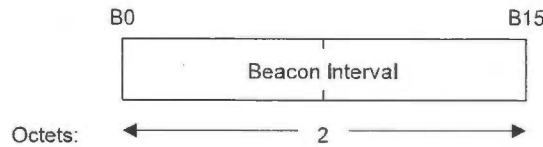


Figure 26—Beacon Interval fixed field

7.3.1.4 Capability Information field

The Capability Information field contains a number of subfields that are used to indicate requested or advertised capabilities. The length of the Capability Information field is 2 octets. The Capability Information field consists of the following subfields: ESS, IBSS, CF-Pollable, CF-Poll Request, and Privacy. The remaining part of the Capability Information field is reserved. The format of the Capability Information field is as illustrated in Figure 27.

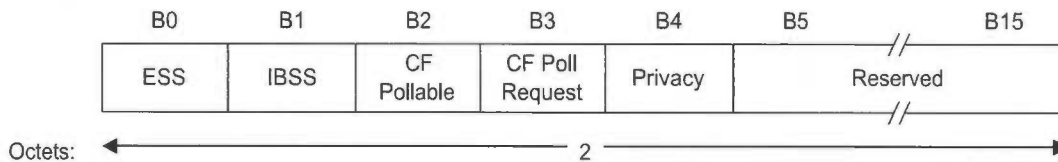


Figure 27—Capability Information fixed field

Each Capability Information subfield is interpreted only in the management frame subtypes for which the transmission rules are defined.

APs set the ESS subfield to 1 and the IBSS subfield to 0 within transmitted Beacon or Probe Response management frames. STAs within an IBSS set the ESS subfield to 0 and the IBSS subfield to 1 in transmitted Beacon or Probe Response management frames.

STAs set the CF-Pollable and CF-Poll Request subfields in Association and Reassociation Request management frames according to Table 16.

Table 16—STA usage of CF-Pollable and CF-Poll Request

CF-Pollable	CF-Poll request	Meaning
0	0	STA is not CF-Pollable
0	1	STA is CF-Pollable, not requesting to be placed on the CF-Polling list
1	0	STA is CF-Pollable, requesting to be placed on the CF-Polling list
1	1	STA is CF-Pollable, requesting never to be polled

APs set the CF-Pollable and CF-Poll Request subfields in Beacon, Probe Response, Association Response, and Reassociation Response management frames according to Table 17. An AP sets the CF-Pollable and CF-Poll Request subfield values in Association Response and Reassociation Response management frames equal to the values in the last Beacon or Probe Response frame that it transmitted.

Table 17—AP usage of CF-Pollable and CF-Poll Request

CF-Pollable	CF-Poll Request	Meaning
0	0	No point coordinator at AP
0	1	Point coordinator at AP for delivery only (no polling)
1	0	Point coordinator at AP for delivery and polling
1	1	Reserved

APs set the Privacy subfield to 1 within transmitted Beacon, Probe Response, Association Response, and Reassociation Response management frames if WEP encryption is required for all data type frames exchanged within the BSS. If WEP encryption is not required, the Privacy subfield is set to 0.

STAs within an IBSS set the Privacy subfield to 1 in transmitted Beacon or Probe Response management frames if WEP encryption is required for all data type frames exchanged within the IBSS. If WEP encryption is not required, the Privacy subfield is set to 0.

7.3.1.5 Current AP Address field

The Current AP Address field is the MAC address of the AP with which the station is currently associated. The length of the Current AP Address field is 6 octets. The Current AP Address field is illustrated in Figure 28.

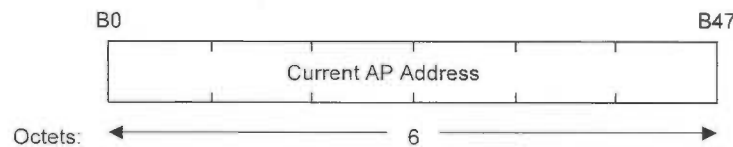


Figure 28—Current AP Address fixed field

7.3.1.6 Listen Interval field

The Listen Interval field is used to indicate to the AP how often an STA wakes to listen to Beacon management frames. The value of this parameter is the STA's Listen Interval parameter of the MLME-Associate.request primitive and is expressed in units of Beacon Interval. The length of the Listen Interval field is 2 octets. The Listen Interval field is illustrated in Figure 29.

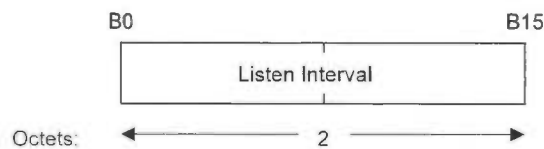


Figure 29—Listen Interval fixed field

An AP may use the Listen Interval information in determining the lifetime of frames that it buffers for an STA.

7.3.1.7 Reason Code field

This Reason Code field is used to indicate the reason that an unsolicited notification management frame of type Disassociation or Deauthentication was generated. The length of the Reason Code field is 2 octets. The Reason Code field is illustrated in Figure 30.

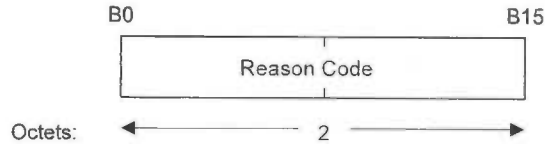


Figure 30—Reason Code fixed field

The reason codes are defined in Table 18.

Table 18—Reason codes

Reason code	Meaning
0	Reserved
1	Unspecified reason
2	Previous authentication no longer valid
3	Deauthenticated because sending station is leaving (or has left) IBSS or ESS
4	Disassociated due to inactivity
5	Disassociated because AP is unable to handle all currently associated stations
6	Class 2 frame received from nonauthenticated station
7	Class 3 frame received from nonassociated station
8	Disassociated because sending station is leaving (or has left) BSS
9	Station requesting (re)association is not authenticated with responding station
10–65 535	Reserved

7.3.1.8 Association ID (AID) field

The AID field is a value assigned by an AP during association that represents the 16-bit ID of a STA. The length of the AID field is 2 octets. The AID field is illustrated in Figure 31.

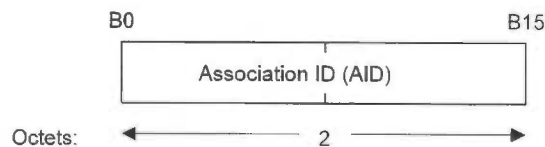


Figure 31—AID fixed field

The value assigned as the Association ID is in the range 1–2007 and is placed in the 14 least significant bits of the AID field, with the two most significant bits of the AID field each set to 1 (see 7.1.3.2).

7.3.1.9 Status Code field

The Status Code field is used in a response management frame to indicate the success or failure of a requested operation. The length of the Status Code field is 2 octets. The Status Code field is illustrated in Figure 32.

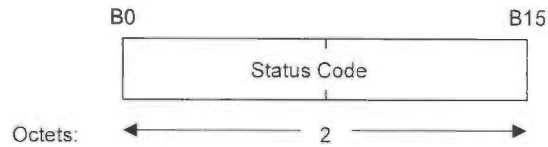


Figure 32—Status Code fixed field

If an operation is successful, then the status code is set to 0. If an operation results in failure, the status code indicates a failure cause. The failure cause codes are defined in Table 19.

Table 19—Status codes

Status code	Meaning
0	Successful
1	Unspecified failure
2–9	Reserved
10	Cannot support all requested capabilities in the Capability Information field
11	Reassociation denied due to inability to confirm that association exists
12	Association denied due to reason outside the scope of this standard
13	Responding station does not support the specified authentication algorithm
14	Received an Authentication frame with authentication transaction sequence number out of expected sequence
15	Authentication rejected because of challenge failure
16	Authentication rejected due to timeout waiting for next frame in sequence
17	Association denied because AP is unable to handle additional associated stations
18	Association denied due to requesting station not supporting all of the data rates in the BSSBasicRateSet parameter
19–65 535	Reserved

7.3.1.10 Timestamp field

This field represents the value of the TSFTIMER (see 11.1) of a frame’s source. The length of the Timestamp field is 8 octets. The Timestamp field is illustrated in Figure 33.

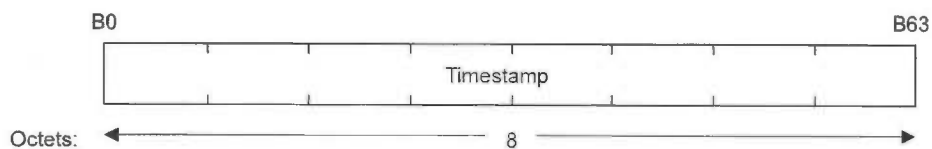


Figure 33—Timestamp fixed field

7.3.2 Information elements

Elements are defined to have a common general format consisting of a 1 octet Element ID field, a 1 octet length field, and a variable-length element-specific information field. Each element is assigned a unique Element ID as defined in this standard. The Length field specifies the number of octets in the Information field. See Figure 34.

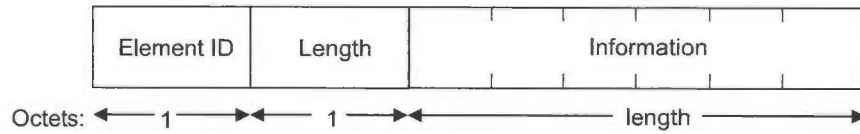


Figure 34—Element format

The set of valid elements is defined in Table 20.

Table 20—Element IDs

Information element	Element ID
SSID	0
Supported rates	1
FH Parameter Set	2
DS Parameter Set	3
CF Parameter Set	4
TIM	5
IBSS Parameter Set	6
Reserved	7–15
Challenge text	16
Reserved for challenge text extension	17–31
Reserved	32–255

7.3.2.1 Service Set Identity (SSID) element

The SSID element indicates the identity of an ESS or IBSS. See Figure 35.

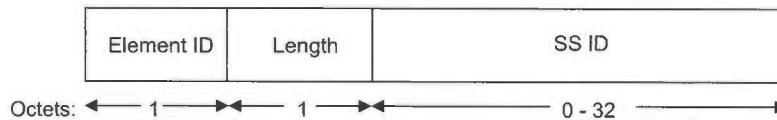


Figure 35—SSID element format

The length of the SSID information field is between 0 and 32 octets. A 0 length information field indicates the broadcast SSID.

7.3.2.2 Supported Rates element

The Supported Rates element specifies the rates in the Operational Rate Set as described in the MLME_Join.request and MLME_Start.request primitives. The information field is encoded as 1 to 8 octets where each octet describes a single supported rate in units of 500 kbit/s.

Within Beacon, Probe Response, Association Response, and Reassociation Response management frames, each supported rate belonging to the BSSBasicRateSet, as defined in 10.3.10.1, is encoded as an octet with the msb (bit 7) set to 1 (e.g., a 1 Mbit/s rate belonging to the BSSBasicRateSet is encoded as X'82'). Rates not belonging to the BSSBasicRateSet are encoded with the msb set to 0 (e.g., a 2 Mbit/s rate not belonging to the BSSBasicRate Set is encoded as X'04'). The msb of each Supported Rate octet in other management frame types is ignored by receiving STAs.

BSSBasicRateSet information in Beacon and Probe Response management frames is used by STAs in order to avoid associating with a BSS if they do not support all the data rates in the BSSBasicRateSet. See Figure 36.

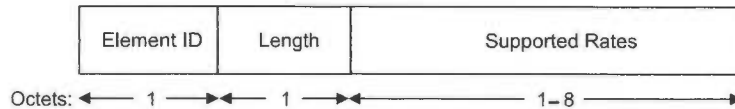


Figure 36—Supported rates element format

7.3.2.3 FH Parameter Set element

The FH Parameter Set element contains the set of parameters necessary to allow synchronization for STAs using a frequency-hopping (FH) PHY. The information field contains Dwell Time, Hop Set, Hop Pattern, and Hop Index parameters. The total length of the information field is 5 octets. See Figure 37.

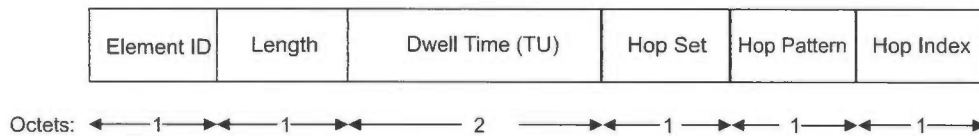


Figure 37—FH Parameter Set element format

The Dwell Time field is 2 octets in length and contains the dwell time in TU.

The Hop Set field identifies the current set (dot11CurrentSet) of hop patterns and is a single octet.

The Hop Pattern field identifies the current pattern (dot11CurrentPattern) within a set of hop patterns and is a single octet.

The Hop Index field selects the current index (dot11CurrentIndex) within a pattern and is a single octet.

The description of the attributes used in this subclause can be found in 14.8.2.1.

7.3.2.4 DS Parameter Set element

The DS Parameter Set element contains information to allow channel number identification for STAs using a direct sequence spread spectrum (DSSS) PHY. The information field contains a single parameter containing the dot11CurrentChannelNumber (see 15.4.6.2 for values). The length of the dot11CurrentChannelNumber parameter is 1 octet. See Figure 38.

7.3.2.5 CF Parameter Set element

The CF Parameter Set element contains the set of parameters necessary to support the PCF. The information field contains the CFPCount, CFPPeriod, CFPPMaxDuration, and CFPPDurRemaining fields. The total length of the information field is 6 octets. See Figure 39.

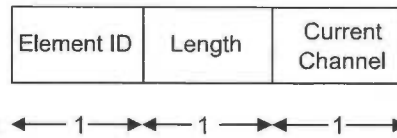


Figure 38—DS Parameter Set element format

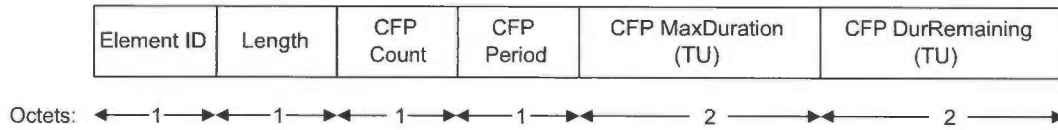


Figure 39—CF Parameter Set element format

CFPCount indicates how many DTIMs (including the current frame) appear before the next CFP start. A CFPCount of 0 indicates that the current DTIM marks the start of the CFP.

CFPPeriod indicates the number of DTIM intervals between the start of CFPs. The value is an integral number of DTIM intervals.

CFPMaxDuration indicates the maximum duration, in TU, of the CFP that may be generated by this PCF. This value is used by STAs to set their NAV at the TBTT of beacons that begin CFPs.

CFPDurRemaining indicates the maximum time, in TU, remaining in the present CFP, and is set to zero in CFP Parameter elements of beacons transmitted during the contention period. The value of CFPDurRemaining is referenced to the immediately previous TBTT. This value is used by all STAs to update their NAVs during CFPs.

7.3.2.6 TIM

The TIM element contains four fields: DTIM Count, DTIM Period, Bitmap Control, and Partial Virtual Bitmap. See Figure 40.

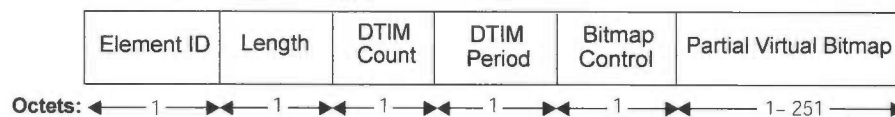


Figure 40—TIM element format

The Length field for this element indicates the length of the information field, which is constrained as described below.

The DTIM Count field indicates how many beacons (including the current frame) appear before the next DTIM. A DTIM Count of 0 indicates that the current TIM is a DTIM. The DTIM count field is a single octet.

The DTIM Period field indicates the number of Beacon intervals between successive DTIMs. If all TIMs are DTIMs, the DTIM Period field has the value 1. The DTIM Period value 0 is reserved. The DTIM period field is a single octet.

The Bitmap Control field is a single octet. Bit 0 of the field contains the Traffic Indicator bit associated with Association ID 0. This bit is set to 1 in TIM elements with a value of 0 in the DTIM Count field when one or

more broadcast or multicast frames are buffered at the AP. The remaining 7 bits of the field form the Bitmap Offset.

The traffic-indication virtual bitmap, maintained by the AP that generates a TIM, consists of 2008 bits, and is organized into 251 octets such that bit number N ($0 \leq N \leq 2007$) in the bitmap corresponds to bit number $(N \bmod 8)$ in octet number $\lfloor N / 8 \rfloor$ where the low-order bit of each octet is bit number 0, and the high order bit is bit number 7. Each bit in the traffic-indication virtual bitmap corresponds to traffic buffered for a specific station within the BSS that the AP is prepared to deliver at the time the beacon frame is transmitted. Bit number N is 0 if there are no directed frames buffered for the station whose Association ID is N . If any directed frames for that station are buffered and the AP is prepared to deliver them, bit number N in the traffic-indication virtual bitmap is 1. A PC may decline to set bits in the TIM for CF-Pollable stations it does not intend to poll (see 11.2.1.5).

The Partial Virtual Bitmap field consists of octets numbered $N1$ through $N2$ of the traffic indication virtual bitmap, where $N1$ is the largest even number such that bits numbered 1 through $(N1 \times 8) - 1$ in the bitmap are all 0 and $N2$ is the smallest number such that bits numbered $(N2 + 1) \times 8$ through 2007 in the bitmap are all 0. In this case, the Bitmap Offset subfield value contains the number $\lfloor N1/2 \rfloor$, and the Length field will be set to $(N2 - N1) + 4$.

In the event that all bits other than bit 0 in the virtual bitmap are 0, the Partial Virtual Bitmap field is encoded as a single octet equal to 0, and the Bitmap Offset subfield is 0.

7.3.2.7 IBSS Parameter Set element

The IBSS Parameter Set element contains the set of parameters necessary to support an IBSS. The information field contains the ATIM Window parameter. See Figure 41.

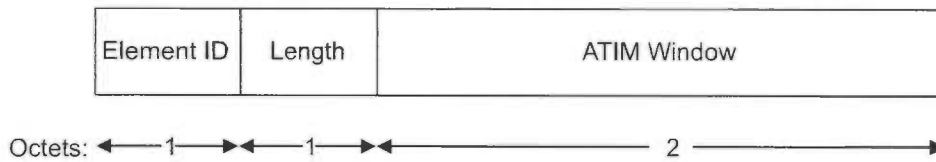


Figure 41—IBSS Parameter Set element format

The ATIM Window field is 2 octets in length and contains the ATIM Window length in TU.

7.3.2.8 Challenge Text element

The Challenge Text element contains the challenge text within Authentication exchanges. The element information field length is dependent upon the authentication algorithm and the transaction sequence number as specified in 8.1. See Figure 42.

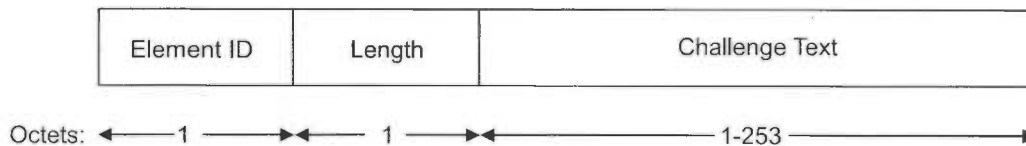


Figure 42—Challenge Text element format

8. Authentication and privacy

8.1 Authentication services

IEEE 802.11 defines two subtypes of authentication service: *Open System* and *Shared Key*. The subtype invoked is indicated in the body of authentication management frames. Thus authentication frames are self-identifying with respect to authentication algorithm. All management frames of subtype Authentication shall be unicast frames as authentication is performed between pairs of stations (i.e., multicast authentication is not allowed). Management frames of subtype Deauthentication are advisory, and may therefore be sent as group-addressed frames.

A mutual authentication relationship shall exist between two stations following a successful authentication exchange as described below. Authentication shall be used between stations and the AP in an infrastructure BSS. Authentication may be used between two STAs in an IBSS.

8.1.1 Open System authentication

Open System authentication is the simplest of the available authentication algorithms. Essentially it is a null authentication algorithm. Any STA that requests authentication with this algorithm may become authenticated if dot11AuthenticationType at the recipient station is set to Open System authentication. Open System authentication is not required to be successful as a STA may decline to authenticate with any particular other STA. Open System authentication is the default authentication algorithm.

Open System authentication involves a two-step authentication transaction sequence. The first step in the sequence is the identity assertion and request for authentication. The second step in the sequence is the authentication result. If the result is “successful,” the STAs shall be mutually authenticated.

8.1.1.1 Open System authentication (first frame)

- Message type: Management
- Message subtype: Authentication
- Information items:
 - Authentication Algorithm Identification = “Open System”
 - Station Identity Assertion (in SA field of header)
 - Authentication transaction sequence number = 1
 - Authentication algorithm dependent information (none)
- Direction of message: From authentication initiating STA to authenticating STA

8.1.1.2 Open System authentication (final frame)

- Message type: Management
- Message subtype: Authentication
- Information items:
 - Authentication Algorithm Identification = “Open System”
 - Authentication transaction sequence number = 2
 - Authentication algorithm dependent information (none)
 - The result of the requested authentication as defined in 7.3.1.9
- Direction of message: From authenticating STA to initiating STA

If dot11AuthenticationType does not include the value “Open System,” the result code shall not take the value “successful.”

8.1.2 Shared Key authentication

Shared Key authentication supports authentication of STAs as either a member of those who know a shared secret key or a member of those who do not. IEEE 802.11 Shared Key authentication accomplishes this without the need to transmit the secret key in the clear; however, it does require the use of the WEP privacy mechanism. Therefore, this authentication scheme is only available if the WEP option is implemented. Additionally, the Shared Key authentication algorithm shall be implemented as one of the dot11AuthenticationAlgorithms at any STA where WEP is implemented.

The required secret, shared key is presumed to have been delivered to participating STAs via a secure channel that is independent of IEEE 802.11. This shared key is contained in a write-only MIB attribute via the MAC management path. The attribute is write-only so that the key value remains internal to the MAC.

During the Shared Key authentication exchange, both the challenge and the encrypted challenge are transmitted. This facilitates unauthorized discovery of the pseudorandom number (PRN) sequence for the key/IV pair used for the exchange. Implementations should therefore avoid using the same key/IV pair for subsequent frames.

A STA shall not initiate a Shared Key authentication exchange unless its dot11PrivacyOptionImplemented attribute is "true."

In the following description, the STA initiating the authentication exchange is referred to as the *requester*, and the STA to which the initial frame in the exchange is addressed is referred to as the *responder*.

8.1.2.1 Shared Key authentication (first frame)

- Message type: Management
- Message subtype: Authentication
- Information Items:
 - Station Identity Assertion (in SA field of header)
 - Authentication Algorithm Identification = "Shared Key"
 - Authentication transaction sequence number = 1
 - Authentication algorithm dependent information (none)
- Direction of message: From requester to responder

8.1.2.2 Shared Key authentication (second frame)

Before sending the second frame in the Shared Key authentication sequence, the responder shall use WEP to generate a string of octets that shall be used as the authentication challenge text.

- Message type: Management
- Message subtype: Authentication
- Information Items:
 - Authentication Algorithm Identification = "Shared Key"
 - Authentication transaction sequence number = 2
 - Authentication algorithm dependent information = the authentication result.
 - The result of the requested authentication as defined in 7.3.1.9

If the status code is not "successful," this shall be the last frame of the transaction sequence. If the status code is not "successful," the content of the challenge text field is unspecified.

If the status code is "successful," the following additional information items shall have valid contents:

Authentication algorithm dependent information = challenge text.

This field shall be of fixed length of 128 octets. The field shall be filled with octets generated by the WEP pseudo-random number generator (PRNG). The actual value of the challenge field is unimportant, but the value shall not be a single static value. The key and IV used when generating the challenge text are unspecified because this key/IV value does not have to be shared and does not affect interoperability.

- Direction of message: From responder to requester

8.1.2.3 Shared Key authentication (third frame)

The requester shall copy the challenge text from the second frame into the third frame. The third frame shall be transmitted after encryption by WEP, as defined in 8.2.3, using the shared secret key.

- Message type: Management
- Message subtype: Authentication
- Information Items:
 - Authentication Algorithm Identification = “Shared Key”
 - Authentication transaction sequence number = 3
 - Authentication algorithm dependent information = challenge text from sequence two frame
- Direction of message: From requester to responder

This frame shall be encrypted as described below.

8.1.2.4 Shared Key authentication (final frame)

The responder shall attempt to decrypt the contents of the third frame in the authentication sequence as described below. If the WEP ICV check is successful, the responder shall then compare the decrypted contents of the Challenge Text field to the challenge text that was sent in Frame 2 of the sequence. If they are the same, then the responder shall respond with a successful status code in Frame 4 of the sequence. If the WEP ICV check fails, the responder shall respond with an unsuccessful status code in Frame 4 of the sequence as described below.

- Message type: Management
- Message subtype: Authentication
- Information Items:
 - Authentication Algorithm Identification = “Shared Key”
 - Authentication transaction sequence number = 4
 - Authentication algorithm dependent information = the authentication result
 - The result of the requested authentication.
 - This is a fixed length item with values “successful” and “unsuccessful.”
- Direction of message: From responder to requester

8.2 The Wired Equivalent Privacy (WEP) algorithm

8.2.1 Introduction

Eavesdropping is a familiar problem to users of other types of wireless technology. IEEE 802.11 specifies a wired LAN equivalent data confidentiality algorithm. *Wired equivalent privacy* is defined as protecting authorized users of a wireless LAN from casual eavesdropping. This service is intended to provide functionality for the wireless LAN equivalent to that provided by the physical security attributes inherent to a wired medium.

Data confidentiality depends on an external key management service to distribute data enciphering/deciphering keys. The IEEE 802.11 standards committee specifically recommends against running an IEEE 802.11

LAN with privacy but without authentication. While this combination is possible, it leaves the system open to significant security threats.

8.2.2 Properties of the WEP algorithm

The WEP algorithm has the following properties:

- *It is reasonably strong:* The security afforded by the algorithm relies on the difficulty of discovering the secret key through a brute-force attack. This in turn is related to the length of the secret key and the frequency of changing keys. WEP allows for the changing of the key (k) and frequent changing of the IV.
- *It is self-synchronizing:* WEP is self-synchronizing for each message. This property is critical for a data-link level encryption algorithm, where “best effort” delivery is assumed and packet loss rates may be high.
- *It is efficient:* The WEP algorithm is efficient and may be implemented in either hardware or software.
- *It may be exportable:* Every effort has been made to design the WEP system operation so as to maximize the chances of approval, by the U.S. Department of Commerce, of export from the U.S. of products containing a WEP implementation. However, due to the legal and political climate toward cryptography at the time of publication, no guarantee can be made that any specific IEEE 802.11 implementations that use WEP will be exportable from the USA.
- *It is optional:* The implementation and use of WEP is an IEEE 802.11 option.

8.2.3 WEP theory of operation

The process of disguising (binary) data in order to hide its information content is called *encryption* (denoted by E) (see [B4]). Data that is not enciphered is called *plaintext* (denoted by P) and data that is enciphered is called *ciphertext* (denoted by C). The process of turning ciphertext back into plaintext is called *decryption* (denoted by D). A *cryptographic algorithm*, or cipher, is a mathematical function used for enciphering or deciphering data. Modern cryptographic algorithms use a key sequence (denoted by k) to modify their output. The encryption function E operates on P to produce C :

$$E_k(P) = C$$

In the reverse process, the decryption function D operates on C to produce P :

$$D_k(C) = P$$

As illustrated in Figure 43, note that if the same key can be used for encryption and decryption then

$$D_k(E_k(P)) = P$$

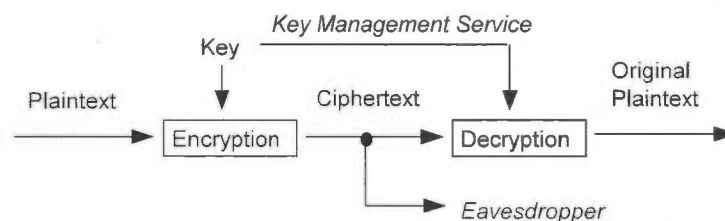


Figure 43—A confidential data channel

The WEP algorithm is a form of electronic code book in which a block of plaintext is bitwise XORed with a pseudorandom key sequence of equal length. The key sequence is generated by the WEP algorithm.

Referring to Figure 44 and viewing from left to right, encipherment begins with a *secret key* that has been distributed to cooperating STAs by an external key management service. WEP is a symmetric algorithm in which the same key is used for encipherment and decipherment.

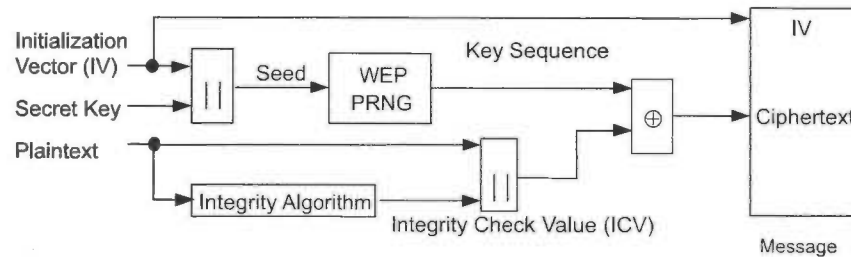


Figure 44—WEP encipherment block diagram

The secret key is concatenated with an *initialization vector (IV)* and the resulting *seed* is input to a PRNG. The PRNG outputs a *key sequence* k of pseudorandom octets equal in length to the number of data octets that are to be transmitted in the expanded MPDU plus 4 [since the key sequence is used to protect the *integrity check value (ICV)* as well as the data]. Two processes are applied to the plaintext MPDU. To protect against unauthorized data modification, an integrity algorithm operates on P to produce an ICV. Encipherment is then accomplished by mathematically combining the key sequence with the plaintext concatenated with the ICV. The output of the process is a *message* containing the IV and ciphertext.

The WEP PRNG is the critical component of this process, since it transforms a relatively short secret key into an arbitrarily long key sequence. This greatly simplifies the task of key distribution, as only the secret key needs to be communicated between STAs. The IV extends the useful lifetime of the secret key and provides the self-synchronous property of the algorithm. The secret key remains constant while the IV changes periodically. Each new IV results in a new seed and key sequence, thus there is a one-to-one correspondence between the IV and k . The IV may be changed as frequently as every MPDU and, since it travels with the message, the receiver will always be able to decipher any message. The IV is transmitted in the clear since it does not provide an attacker with any information about the secret key, and since its value must be known by the recipient in order to perform the decryption.

When choosing how often to change IV values, implementors should consider that the contents of some fields in higher-layer protocol headers, as well as certain other higher-layer information, is constant or highly predictable. When such information is transmitted while encrypting with a particular key and IV, an eavesdropper can readily determine portions of the key sequence generated by that (key, IV) pair. If the same (key, IV) pair is used for successive MPDUs, this effect may substantially reduce the degree of privacy conferred by the WEP algorithm, allowing an eavesdropper to recover a subset of the user data without any knowledge of the secret key. Changing the IV after each MPDU is a simple method of preserving the effectiveness of WEP in this situation.

The WEP algorithm is applied to the frame body of an MPDU. The (IV, frame body, ICV) triplet forms the actual data to be sent in the data frame.

For WEP protected frames, the first four octets of the frame body contain the IV field for the MPDU. This field is defined in 8.2.5. The PRNG seed is 64 bits. Bits 0 through 23 of the IV correspond to bits 0 through 23 of the PRNG seed, respectively. Bits 0 through 39 of the secret key correspond to bits 24 through 63 of the PRNG seed, respectively. The bit and octet numbering conventions in 7.1.1 apply to the PRNG seed, secret key, and IV. The numbering of the octets of the PRNG seed corresponds to that of the RC4 key. The IV is followed by the MPDU, which is followed by the ICV. The WEP ICV is 32 bits. The WEP Integrity Check algorithm is CRC-32, as defined in 7.1.3.6.

As stated previously, WEP combines k with P using bitwise XOR.

Referring to Figure 45 and viewing from left to right, decipherment begins with the arrival of a message. The IV of the incoming message shall be used to generate the key sequence necessary to decipher the incoming message. Combining the ciphertext with the proper key sequence yields the original plaintext and ICV. Correct decipherment shall be verified by performing the integrity check algorithm on the recovered plaintext and comparing the output ICV' to the ICV transmitted with the message. If ICV' is not equal to ICV, the received MPDU is in error and an error indication is sent to MAC management. MSDUs with erroneous MPDUs (due to inability to decrypt) shall not be passed to LLC.

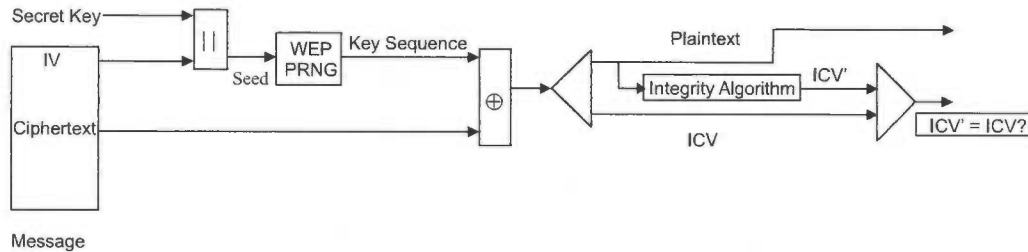


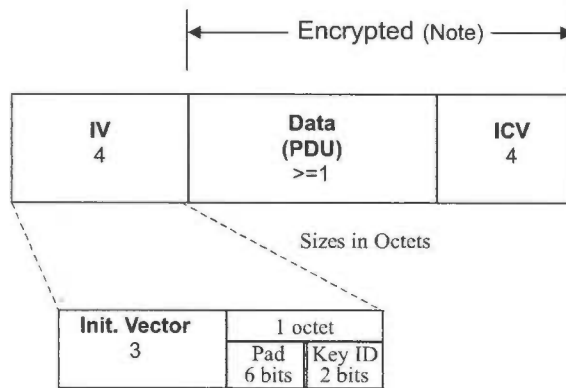
Figure 45—WEP decipherment block diagram

8.2.4 WEP algorithm specification

WEP uses the RC4 PRNG algorithm from RSA Data Security, Inc.⁶

8.2.5 WEP Frame Body expansion

Figure 46 shows the encrypted Frame Body as constructed by the WEP algorithm.



NOTE – The encypherment process has expanded the original Frame Body by 8 octets, 4 for the IV field and 4 for the ICV. The ICV is calculated on the data field only.

Figure 46—Construction of expanded WEP Frame Body

The WEP ICV shall be a 32-bit field containing the CRC-32, as defined in 7.1.3.6 calculated over the Data (PDU) field as depicted in Figure 46. The expanded Frame Body shall include a 32-bit IV field immediately preceding the original Frame Body. This field shall contain three subfields: a three-octet field that contains the initialization vector, a 2-bit key ID field, and a 6-bit pad field. The ordering conventions defined in 7.1.1 apply to the IV fields and its subfields and to the ICV field. The key ID subfield contents select one of four

⁶Details of the RC4 algorithm are available from RSA. Please contact RSA for algorithm details and the uniform RC4 licensee terms that RSA offers to anyone wishing to use RC4 for the purpose of implementing the IEEE 802.11 WEP option. If necessary, contact the IEEE Standards Department Intellectual Property Rights Administrator for details on how to communicate with RSA.

possible secret key values for use in decrypting this Frame Body. Interpretation of these bits is discussed further in 8.3.2. The contents of the pad subfield shall be zero. The key ID occupies the two msb of the last octet of the IV field, while the pad occupies the six lsb of this octet.

The WEP mechanism is invisible to entities outside the IEEE 802.11 MAC data path.

8.3 Security-Related MIB attributes

The IEEE 802.11 security mechanisms are controlled via the MAC management path and related MIB attributes. This subclause gives an overview of the security-related MIB attributes and how they are used. For details of the MIB attribute definitions, refer to 11.4.

8.3.1 Authentication-Related MIB attributes

The type of authentication invoked when authentication is attempted is controlled by the AuthenticationType parameter to the MLME-AUTHENTICATE.request primitive. The type of authentication request that may be accepted by a STA is controlled by the MIB attribute dot11AuthenticationType. The type of authentication is selected from the following set of values:

- Open System
- Shared Key

All other values are reserved. The numeric encoding of these values is given in 7.3.1.1.

8.3.2 Privacy-Related MIB attributes

WEP invocation is controlled by the parameters passed to the MLME-AUTHENTICATE.request primitive as well as a number of MIB attributes. An overview of the attributes and their usage is given in this subclause. All MIB attributes that hold WEP keys are externally write-only; the contents shall not be readable via MAC management SAPs. See 11.4 for the formal MIB attribute definitions.

The boolean variable dot11PrivacyInvoked shall be set to “false” to prevent the STA from transmitting MPDUs of type Data with the WEP subfield of the Frame Control field set to 1. It does not affect MPDU or MMPDU reception.

The default value for all WEP keys shall be null. Note that encrypting a frame using WEP with a null key is not the same as failing to encrypt the frame. Any request to encrypt a frame with a null key shall result in the MSDU being discarded and an MA-UNIDATA-STATUS.indication with a transmission status indicating that the frame may not be encrypted with a null key. Decrypting a frame whose WEP subfield is set to 1 involves stripping the IV, and checking the ICV against the calculated ICV' value computed over the data contained in the MPDU.

To support shared key configurations, the MIB contains a four-element vector called “dot11WEPDefaultKeys.” The default value for each element of this vector is null. These elements contain the default keys to be used with WEP.

An additional attribute called “dot11WEPDefaultKeyID” is an integer. When set to a value of 0, 1, 2, or 3, MPDUs transmitted with the WEP subfield of the Frame Control field set to 1 shall be encrypted using the first, second, third, or fourth element, respectively, from dot11WEPDefaultKeys, unless the frame has an individual RA and a key mapping exists for the RA of the frame. On receive, the incoming MPDU shall be decrypted using the element from dot11WEPDefaultKeys specified by the received key ID field, unless the frame has an individual RA and a key mapping exists for the TA of the frame. The value in the transmitted key ID field shall be zero in all cases except when dot11WEPDefaultKeyID is used to encrypt a frame and is

set to a value of 1, 2, or 3, in which case the transmitted key ID field shall contain the value of dot11WEPDefaultKeyID.

When the boolean attribute aExcludeUnencrypted is set to True, MPDUs of type Data received by the STA with the WEP subfield of the Frame Control field equal to zero shall not be indicated at the MAC service interface. When aExcludeUnencrypted is set to True, only MSDUs that have been decrypted successfully shall be indicated at the MAC service interface.

IEEE 802.11 does not require that the same WEP key be used for all STAs. The MIB supports the ability to share a separate WEP key for each RA/TA pair. Key mapping is supported by a MIB attribute that is an array called "dot11WEPKeyMappings." dot11WEPKeyMappings contains zero or one entry for each MAC address, up to an implementation-defined maximum number of entries identified by dot11WEPKeyMappingLength, and contains two fields for each entry: a boolean "WEPOn" and the corresponding WEPKey. In an infrastructure BSS, the AP's WEPOn value in the entry in its dot11WEPKeyMapping table corresponding to a STA's MAC address shall not be set to True for a STA if that STA has not successfully initiated and completed an authentication sequence using an authentication type other than "Open System." The default value for all WEPOn fields is False. dot11WEPKeyMappings shall be indexed by either RA or TA addresses (since WEP is applied only to the wireless link), as described below. When an entry in the table exists for a particular MAC address, the values in the dot11WEPKeyMappings attribute shall be used instead of the dot11WEPDefaultKeyID and dot11WEPDefaultKeys variables.

The minimal value of dot11WEPKeyMappingLength shall be 10. This value represents a minimum capability that may be assumed for any STA implementing the WEP option.

When transmitting a frame of type Data, the values of dot11PrivacyInvoked, dot11WEPKeyMappings, dot11WEPDefaultKeys, and dot11WEPDefaultKeyID in effect at an unspecified time between receipt by the MAC of the MAUNITDATA.request primitive and the time of transmission of that frame shall be used according to the following decision tree:

```

if dot11PrivacyInvoked is "false"
    the MPDU is transmitted without encryption
else
    if (the MPDU has an individual RA and
        there is an entry in dot11WEPKeyMappings for that RA)
        if that entry has WEPOn set to "false"
            the MPDU is transmitted without encryption
        else
            if that entry contains a key that is null
                discard the entire MSDU and generate an
                MA-UNITDATA-STATUS.indication primitive to
                notify LLC that the MSDU was undeliverable due to
                a null WEP key
            else
                encrypt the MPDU using that entry's key, setting the keyID
                subfield of the IV field to zero
    else
        if (the MPDU has a group RA and the Privacy subfield
            of the Capability Information field in this BSS is set to 0)
            the MPDU is transmitted without encryption
        else
            if dot11WEPDefaultKeys[dot11WEPDefaultKeyID] is null
                discard the MSDU and generate an
                MA-UNITDATA-STATUS.indication primitive to

```

```

        notify LLC that the entire MSDU was undeliverable
        due to a null WEP key
    else
        encrypt the MPDU using
        dot11WEPDefaultKeys[dot11WEPDefaultKeyID],
        setting the KeyID subfield of the IV field to
        dot11WEPDefaultKeyID

```

When receiving a frame of type Data, the values of dot11PrivacyOptionImplemented, dot11WEPKeyMappings, dot11WEPDefaultKeys, dot11WEPDefaultKeyID, and aExcludeUnencrypted in effect at the time the PHY-RXSTART.indication primitive is received by the MAC shall be used according to the following decision tree:

```

    if the WEP subfield of the Frame Control Field is zero
        if aExcludeUnencrypted is "true"
            discard the frame body without indication to LLC and increment
            dot11WEPExcludedCount
        else
            receive the frame without decryption
    else
        if dot11PrivacyOptionImplemented is "true"
            if (the MPDU has individual RA and
                there is an entry in dot11WEPKeyMappings matching the MPDU's TA)
                if that entry has WEPOn set to "false"
                    discard the frame body and increment
                    dot11WEPUndecryptableCount
                else
                    if that entry contains a key that is null
                        discard the frame body and increment
                        dot11WEPUndecryptableCount
                    else
                        attempt to decrypt with that key, incrementing
                        dot11WEPICVErrorCount if the ICV check fails
            else
                if dot11WEPDefaultKeys[keyID] is null
                    discard the frame body and increment
                    dot11WEPUndecryptableCount
                else
                    attempt to decrypt with dot11WEPDefaultKeys[keyID],
                    incrementing dot11WEPICVErrorCount if the ICV check fails
        else
            discard the frame body and increment dot11WEPUndecryptableCount

```

When transmitting a frame of type Management, subtype Authentication with an Authentication Transaction Sequence Number field value of 2, the MAC shall operate according to the following decision tree:

```

    if dot11PrivacyOptionImplemented is "false"
        the MMPDU is transmitted with a sequence
        of zero octets in the Challenge Text field and a Status Code value of 13
    else
        the MMPDU is transmitted with a sequence of 128 octets generated using the
        WEP PRNG and a key whose value is unspecified and beyond the scope of this
        standard and a randomly chosen IV value (note that this will typically be selected
        by the same mechanism for choosing IV values for transmitted data MPDUs)

```

in the Challenge Text field and a status code value of 0 (the IV used is immaterial and is not transmitted). Note that there are cryptographic issues involved in the choice of key/IV for this process as the challenge text is sent unencrypted and therefore provides a known output sequence from the PRNG.

When receiving a frame of type Management, subtype Authentication with an Authentication Transaction Sequence Number field value of 2, the MAC shall operate according to the following decision tree:

```

if the WEP subfield of the Frame Control field is 1
    respond with a status code value of 15
else
    if dot11PrivacyOptionImplemented is "true"
        if there is a mapping in dot11WEPKeyMappings matching the MSDU's TA
            if that key is null
                respond with a frame whose Authentication Transaction
                Sequence Number field is 3 that contains the appropriate
                Authentication Algorithm Number, a status code value of
                15 and no Challenge Text field, without encrypting the
                contents of the frame
            else
                respond with a frame whose Authentication Transaction
                Sequence Number field is 3 that contains the appropriate
                Authentication Algorithm Number, a status code value of
                0 and the identical Challenge Text field, encrypted using
                that key, and setting the key ID subfield in the IV field to 0
        else
            if dot11WEPDefaultKeys[dot11WEPDefaultKeyID] is null
                respond with a frame whose Authentication Transaction
                Sequence Number field is 3 that contains the appropriate
                Authentication Algorithm Number, a status code value of
                15 and no Challenge Text field, without encrypting the
                contents of the frame
            else
                respond with a frame whose Authentication Transaction
                Sequence Number field is 3 that contains the appropriate
                Authentication Algorithm Number, a status code value of 0
                and the identical Challenge Text field, encrypted using
                dot11WEPDefaultKeys[dot11WEPDefaultKeyID], setting the
                key ID subfield in the IV field to dot11WEPDefaultKeyID
    else
        respond with a frame whose Authentication Transaction
        Sequence Number field is 3 that contains the appropriate Authentication
        Algorithm Number, a status code value of 13 and no Challenge Text
        field, without encrypting the contents of the frame
  
```

When receiving a frame of type Management, subtype Authentication with an Authentication Transaction Sequence Number field value of 3, the MAC shall operate according to the following decision tree:

```

if the WEP subfield of the Frame Control field is zero
    respond with a status code value of 15
else
    if dot11PrivacyOptionImplemented is "true"
        if there is a mapping in dot11WEPKeyMappings matching the MSDU's TA
  
```

```

    if that key is null
        respond with a frame whose Authentication Transaction
        Sequence Number field is 4 that contains the appropriate
        Authentication Algorithm Number, and a status code value
        of 15 without encrypting the contents of the frame
    else
        attempt to decrypt with that key, incrementing
        dot11WEPICVErrorCount and responding with a status code value
        of 15 if the ICV check fails
    else
        if dot11WEPDefaultKeys[keyID] is null
            respond with a frame whose Authentication Transaction
            Sequence Number field is 4 that contains the appropriate
            Authentication Algorithm Number, and a status code value
            of 15 without encrypting the contents of the frame
        else
            attempt to decrypt with dot11WEPDefaultKeys[keyID],
            incrementing dot11WEPICVErrorCount and responding with
            a status code value of 15 if the ICV check fails
    else
        respond with a frame whose Authentication Transaction Sequence Number
        field is 4 that contains the appropriate Authentication Algorithm Number,
        and a status code value of 15

```

The attribute dot11PrivacyInvoked shall not take the value “true” if the attribute dot11PrivacyOptionImplemented is “false.” Setting the attribute dot11WEPKeyMappings to a value that includes more than dot11WEPKeyMappingLength entries is illegal and shall have an implementation-specific effect on the operation of the privacy service. Note that dot11WEPKeyMappings may contain between zero and dot11WEPKeyMappingLength entries, inclusive.

It is recommended that the values of the attributes in the aPrivacygrp not be changed during the authentication sequence as unintended operation may result.

9. MAC sublayer functional description

The MAC functional description is presented in this clause. The architecture of the MAC sublayer, including the distributed coordination function (DCF), the point coordination function (PCF), and their coexistence in an IEEE 802.11 LAN are introduced in 9.1. These functions are expanded on in 9.2 and 9.3, and a complete functional description of each is provided. Fragmentation and defragmentation are covered in 9.4 and 9.5. Multirate support is addressed in 9.6. The allowable frame exchange sequences are listed in 9.7. Finally, a number of additional restrictions to limit the cases in which MSDUs are reordered or discarded are described in 9.8.

9.1 MAC architecture

The MAC architecture can be described as shown in Figure 47 as providing the PCF through the services of the DCF.

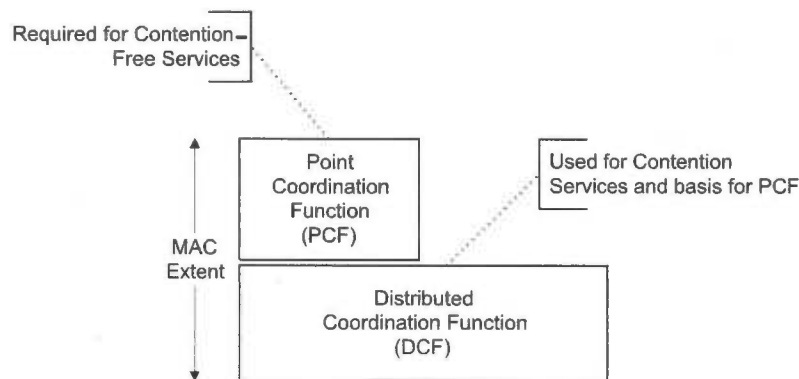


Figure 47—MAC architecture

9.1.1 Distributed coordination function (DCF)

The fundamental access method of the IEEE 802.11 MAC is a DCF known as *carrier sense multiple access with collision avoidance* (CSMA/CA). The DCF shall be implemented in all STAs, for use within both IBSS and infrastructure network configurations.

For a STA to transmit, it shall sense the medium to determine if another STA is transmitting. If the medium is not determined to be busy (see 9.2.1), the transmission may proceed. The CSMA/CA distributed algorithm mandates that a gap of a minimum specified duration exist between contiguous frame sequences. A transmitting STA shall ensure that the medium is idle for this required duration before attempting to transmit. If the medium is determined to be busy, the STA shall defer until the end of the current transmission. After deferral, or prior to attempting to transmit again immediately after a successful transmission, the STA shall select a random backoff interval and shall decrement the backoff interval counter while the medium is idle. A refinement of the method may be used under various circumstances to further minimize collisions—here the transmitting and receiving STA exchange short control frames [request to send (RTS) and clear to send (CTS) frames] after determining that the medium is idle and after any deferrals or backoffs, prior to data transmission. The details of CSMA/CA, deferrals, and backoffs are described in 9.2. RTS/CTS exchanges are also presented in 9.2.

9.1.2 Point coordination function (PCF)

The IEEE 802.11 MAC may also incorporate an optional access method called a PCF, which is only usable on infrastructure network configurations. This access method uses a point coordinator (PC), which shall

operate at the access point of the BSS, to determine which STA currently has the right to transmit. The operation is essentially that of polling, with the PC performing the role of the polling master. The operation of the PCF may require additional coordination, not specified in this standard, to permit efficient operation in cases where multiple point-coordinated BSSs are operating on the same channel, in overlapping physical space.

The PCF uses a virtual carrier-sense mechanism aided by an access priority mechanism. The PCF shall distribute information within Beacon management frames to gain control of the medium by setting the network allocation vector (NAV) in STAs. In addition, all frame transmissions under the PCF may use an interframe space (IFS) that is smaller than the IFS for frames transmitted via the DCF. The use of a smaller IFS implies that point-coordinated traffic shall have priority access to the medium over STAs in overlapping BSSs operating under the DCF access method.

The access priority provided by a PCF may be utilized to create a *contention-free* (CF) access method. The PC controls the frame transmissions of the STAs so as to eliminate contention for a limited period of time.

9.1.3 Coexistence of DCF and PCF

The DCF and the PCF shall coexist in a manner that permits both to operate concurrently within the same BSS. When a PC is operating in a BSS, the two access methods alternate, with a contention-free period (CFP) followed by a contention period (CP). This is described in greater detail in 9.3.

9.1.4 Fragmentation/defragmentation overview

The process of partitioning a MAC service data unit (MSDU) or a MAC management protocol data unit (MMPDU) into smaller MAC level frames, MAC protocol data units (MPDUs), is called fragmentation. Fragmentation creates MPDUs smaller than the original MSDU or MMPDU length to increase reliability, by increasing the probability of successful transmission of the MSDU or MMPDU in cases where channel characteristics limit reception reliability for longer frames. Fragmentation is accomplished at each immediate transmitter. The process of recombining MPDUs into a single MSDU or MMPDU is defined as defragmentation. Defragmentation is accomplished at each immediate recipient.

Only MPDUs with a unicast receiver address shall be fragmented. Broadcast/multicast frames shall not be fragmented even if their length exceeds aFragmentationThreshold.

When a directed MSDU is received from the LLC or a directed MMPDU is received from the MAC sublayer management entity (MLME) with a length greater than aFragmentationThreshold, the MSDU or MMPDU shall be fragmented. The MSDU or MMPDU is divided into MPDUs. Each fragment is a frame no larger than aFragmentationThreshold. It is possible that any fragment may be a frame smaller than aFragmentationThreshold. An illustration of fragmentation is shown in Figure 48.

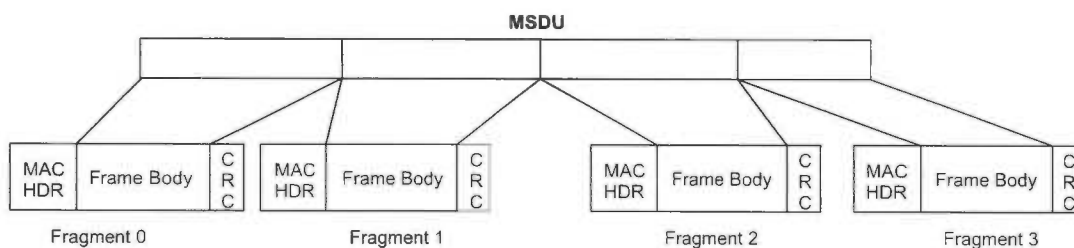


Figure 48—Fragmentation

The MPDUs resulting from the fragmentation of an MSDU or MMPDU are sent as independent transmissions, each of which is separately acknowledged. This permits transmission retries to occur per fragment,

rather than per MSDU or MMPDU. Unless interrupted due to medium occupancy limitations for a given PHY, the fragments of a single MSDU or MMPDU are sent as a burst during the CP, using a single invocation of the DCF medium access procedure. The fragments of a single MSDU or MMPDU are sent during a CFP as individual frames obeying the rules of the PC medium access procedure.

9.1.5 MAC data service

The MAC data service shall translate MAC service requests from LLC into input signals utilized by the MAC state machines. The MAC data service shall also translate output signals from the MAC state machines into service indications to LLC. The translations are given in the MAC data service state machine defined in Annex C.

9.2 DCF

The basic medium access protocol is a DCF that allows for automatic medium sharing between compatible PHYs through the use of CSMA/CA and a random backoff time following a busy medium condition. In addition, all directed traffic uses immediate positive acknowledgment (ACK frame) where retransmission is scheduled by the sender if no ACK is received.

The CSMA/CA protocol is designed to reduce the collision probability between multiple STAs accessing a medium, at the point where collisions would most likely occur. Just after the medium becomes idle following a busy medium (as indicated by the CS function) is when the highest probability of a collision exists. This is because multiple STAs could have been waiting for the medium to become available again. This is the situation that necessitates a random backoff procedure to resolve medium contention conflicts.

Carrier sense shall be performed both through physical and virtual mechanisms.

The virtual carrier-sense mechanism is achieved by distributing reservation information announcing the impending use of the medium. The exchange of RTS and CTS frames prior to the actual data frame is one means of distribution of this medium reservation information. The RTS and CTS frames contain a Duration/ID field that defines the period of time that the medium is to be reserved to transmit the actual data frame and the returning ACK frame. All STAs within the reception range of either the originating STA (which transmits the RTS) or the destination STA (which transmits the CTS) shall learn of the medium reservation. Thus a STA can be unable to receive from the originating STA, yet still know about the impending use of the medium to transmit a data frame.

Another means of distributing the medium reservation information is the Duration/ID field in directed frames. This field gives the time that the medium is reserved, either to the end of the immediately following ACK, or in the case of a fragment sequence, to the end of the ACK following the next fragment.

The RTS/CTS exchange also performs both a type of fast collision inference and a transmission path check. If the return CTS is not detected by the STA originating the RTS, the originating STA may repeat the process (after observing the other medium-use rules) more quickly than if the long data frame had been transmitted and a return ACK frame had not been detected.

Another advantage of the RTS/CTS mechanism occurs where multiple BSSs utilizing the same channel overlap. The medium reservation mechanism works across the BSA boundaries. The RTS/CTS mechanism may also improve operation in a typical situation where all STAs can receive from the AP, but cannot receive from all other STAs in the BSA.

The RTS/CTS mechanism cannot be used for MPDUs with broadcast and multicast immediate address because there are multiple destinations for the RTS, and thus potentially multiple concurrent senders of the CTS in response. The RTS/CTS mechanism need not be used for every data frame transmission. Because the

additional RTS and CTS frames add overhead inefficiency, the mechanism is not always justified, especially for short data frames.

The use of the RTS/CTS mechanism is under control of the `dot11RTSThreshold` attribute. This attribute may be set on a per-STA basis. This mechanism allows STAs to be configured to use RTS/CTS either always, never, or only on frames longer than a specified length.

A STA configured not to initiate the RTS/CTS mechanism shall still update its virtual carrier-sense mechanism with the duration information contained in a received RTS or CTS frame, and shall always respond to an RTS addressed to it with a CTS.

The medium access protocol allows for STAs to support different sets of data rates. All STAs shall receive all the data rates in `aBasicRateSet` and transmit at one or more of the `aBasicRateSet` data rates. To support the proper operation of the RTS/CTS and the virtual carrier-sense mechanism, all STAs shall be able to detect the RTS and CTS frames. For this reason the RTS and CTS frames shall be transmitted at one of the `aBasicRateSet` rates. (See 9.6 for a description of multirate operation.)

Data frames sent under the DCF shall use the frame type `Data` and subtype `Data` or `Null Function`. STAs receiving `Data` type frames shall only consider the frame body as the basis of a possible indication to LLC.

9.2.1 Carrier-sense mechanism

Physical and virtual carrier-sense functions are used to determine the state of the medium. When either function indicates a busy medium, the medium shall be considered busy; otherwise, it shall be considered idle.

A physical carrier-sense mechanism shall be provided by the PHY. See Clause 12 for how this information is conveyed to the MAC. The details of physical carrier sense are provided in the individual PHY specifications.

A virtual carrier-sense mechanism shall be provided by the MAC. This mechanism is referred to as the network allocation vector (NAV). The NAV maintains a prediction of future traffic on the medium based on duration information that is announced in RTS/CTS frames prior to the actual exchange of data. The duration information is also available in the MAC headers of all frames sent during the CP other than PS-Poll Control frames. The mechanism for setting the NAV using RTS/CTS in the DCF is described in 9.2.5.4, and use of the NAV in PCF is described in 9.3.2.2.

The carrier-sense mechanism combines the NAV state and the STA's transmitter status with physical carrier sense to determine the busy/idle state of the medium. The NAV may be thought of as a counter, which counts down to zero at a uniform rate. When the counter is zero, the virtual carrier-sense indication is that the medium is idle; when nonzero, the indication is busy. The medium shall be determined to be busy whenever the STA is transmitting.

9.2.2 MAC-Level acknowledgments

The reception of some frames, as described in 9.7, 9.2.8, and 9.3.3.4, requires the receiving STA to respond with an acknowledgment, generally an ACK frame, if the FCS of the received frame is correct. This technique is known as positive acknowledgment.

Lack of reception of an expected ACK frame indicates to the source STA that an error has occurred. Note, however, that the destination STA may have received the frame correctly, and that the error may have occurred in the reception of the ACK frame. To the initiator of the frame exchange, this condition is indistinguishable from an error occurring in the initial frame.

9.2.3 Interframe space (IFS)

The time interval between frames is called the IFS. A STA shall determine that the medium is idle through the use of the carrier-sense function for the interval specified. Four different IFSs are defined to provide priority levels for access to the wireless media; they are listed in order, from the shortest to the longest. Figure 49 shows some of these relationships.

- a) SIFS short interframe space
- b) PIFS PCF interframe space
- c) DIFS DCF interframe space
- d) EIFS extended interframe space

The different IFSs shall be independent of the STA bit rate. The IFS timings shall be defined as time gaps on the medium, and shall be fixed for each PHY (even in multirate-capable PHYs). The IFS values are determined from attributes specified by the PHY.

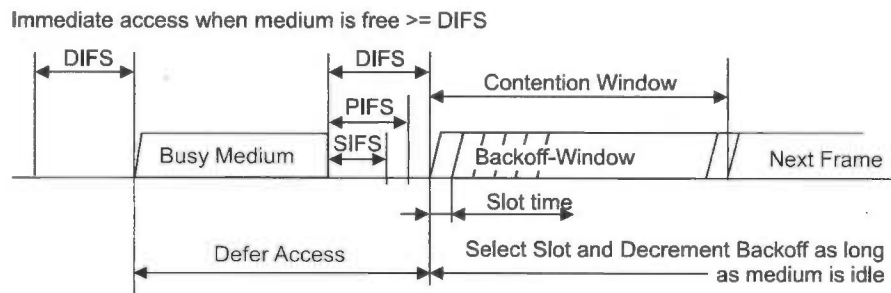


Figure 49—Some IFS relationships

9.2.3.1 Short IFS (SIFS)

The SIFS shall be used for an ACK frame, a CTS frame, the second or subsequent MPDU of a fragment burst, and by a STA responding to any polling by the PCF. It may also be used by a PC for any types of frames during the CFP (see 9.3). The SIFS is the time from the end of the last symbol of the previous frame to the beginning of the first symbol of the preamble of the subsequent frame as seen at the air interface. The valid cases where the SIFS may or shall be used are listed in the frame exchange sequences in 9.7.

The SIFS timing shall be achieved when the transmission of the subsequent frame is started at the TxSIFS Slot boundary as specified in 9.2.10. An IEEE 802.11 implementation shall not allow the space between frames that are defined to be separated by a SIFS time, as measured on the medium, to vary from the nominal SIFS value by more than $\pm 10\%$ of aSlotTime for the PHY in use.

SIFS is the shortest of the interframe spaces. SIFS shall be used when STAs have seized the medium and need to keep it for the duration of the frame exchange sequence to be performed. Using the smallest gap between transmissions within the frame exchange sequence prevents other STAs, which are required to wait for the medium to be idle for a longer gap, from attempting to use the medium, thus giving priority to completion of the frame exchange sequence in progress.

9.2.3.2 PCF IFS (PIFS)

The PIFS shall be used only by STAs operating under the PCF to gain priority access to the medium at the start of the CFP. A STA using the PCF shall be allowed to transmit contention-free traffic after its carrier-sense mechanism (see 9.2.1) determines that the medium is idle at the TxPIFS slot boundary as defined in 9.2.10. Subclause 9.3 describes the use of the PIFS by STAs operating under the PCF.

9.2.3.3 DCF IFS (DIFS)

The DIFS shall be used by STAs operating under the DCF to transmit data frames (MPDUs) and management frames (MMPDUs). A STA using the DCF shall be allowed to transmit if its carrier-sense mechanism (see 9.2.1) determines that the medium is idle at the TxDIFS slot boundary as defined in 9.2.10 after a correctly received frame, and its backoff time has expired. A STA using the DCF shall not transmit within an EIFS after it determines that the medium is idle following reception of a frame for which the PHYRX-END.indication primitive contained an error or a frame for which the MAC FCS value was not correct. A STA may transmit after subsequent reception of an error-free frame, resynchronizing the STA. This allows the STA to transmit using the DIFS following that frame.

9.2.3.4 Extended IFS (EIFS)

The EIFS shall be used by the DCF whenever the PHY has indicated to the MAC that a frame transmission was begun that did not result in the correct reception of a complete MAC frame with a correct FCS value. The duration of an EIFS is defined in 9.2.10. The EIFS interval shall begin following indication by the PHY that the medium is idle after detection of the erroneous frame, without regard to the virtual carrier-sense mechanism. The EIFS is defined to provide enough time for another STA to acknowledge what was, to this STA, an incorrectly received frame before this STA commences transmission. Reception of an error-free frame during the EIFS resynchronizes the STA to the actual busy/idle state of the medium, so the EIFS is terminated and normal medium access (using DIFS and, if necessary, backoff) continues following reception of that frame.

9.2.4 Random backoff time

A STA desiring to initiate transfer of data MPDUs and/or management MMPDUs shall invoke the carrier-sense mechanism (see 9.2.1) to determine the busy/idle state of the medium. If the medium is busy, the STA shall defer until the medium is determined to be idle without interruption for a period of time equal to DIFS when the last frame detected on the medium was received correctly, or after the medium is determined to be idle without interruption for a period of time equal to EIFS when the last frame detected on the medium was not received correctly. After this DIFS or EIFS medium idle time, the STA shall then generate a random backoff period for an additional deferral time before transmitting, unless the backoff timer already contains a nonzero value, in which case the selection of a random number is not needed and not performed. This process minimizes collisions during contention between multiple STAs that have been deferring to the same event.

$$\text{Backoff Time} = \text{Random}() \times \text{aSlotTime}$$

where

$\text{Random}()$ = Pseudorandom integer drawn from a uniform distribution over the interval $[0, CW]$, where CW is an integer within the range of values of the PHY characteristics aCW_{\min} and aCW_{\max} , $aCW_{\min} \leq CW \leq aCW_{\max}$. It is important that designers recognize the need for statistical independence among the random number streams among STAs.

$aSlotTime$ = The value of the correspondingly named PHY characteristic.

The contention window (CW) parameter shall take an initial value of aCW_{\min} . Every STA shall maintain a STA short retry count (SSRC) as well as a STA long retry count (SLRC), both of which shall take an initial value of zero. The SSRC shall be incremented whenever any short retry count associated with any MSDU is incremented. The SLRC shall be incremented whenever any long retry count associated with any MSDU is incremented. The CW shall take the next value in the series every time an unsuccessful attempt to transmit an MPDU causes either STA retry counter to increment, until the CW reaches the value of aCW_{\max} . A retry is defined as the entire sequence of frames sent, separated by SIFS intervals, in an attempt to deliver an MPDU, as described in 9.7. Once it reaches aCW_{\max} , the CW shall remain at the value of aCW_{\max} until it is reset. This improves the stability of the access protocol under high-load conditions. See Figure 50.

The CW shall be reset to aCW_{min} after every successful attempt to transmit an MSDU or MMPDU, when SLRC reaches $aLongRetryLimit$, or when SSRC reaches $dot11ShortRetryLimit$. The SSRC shall be reset to 0 whenever a CTS frame is received in response to an RTS frame, whenever an ACK frame is received in response to an MPDU or MMPDU transmission, or whenever a frame with a group address in the Address1 field is transmitted. The SLRC shall be reset to 0 whenever an ACK frame is received in response to transmission of an MPDU or MMPDU of length greater than $dot11RTSThreshold$, or whenever a frame with a group address in the Address1 field is transmitted.

The set of CW values shall be sequentially ascending integer powers of 2, minus 1, beginning with a PHY-specific aCW_{min} value, and continuing up to and including a PHY-specific aCW_{max} value.

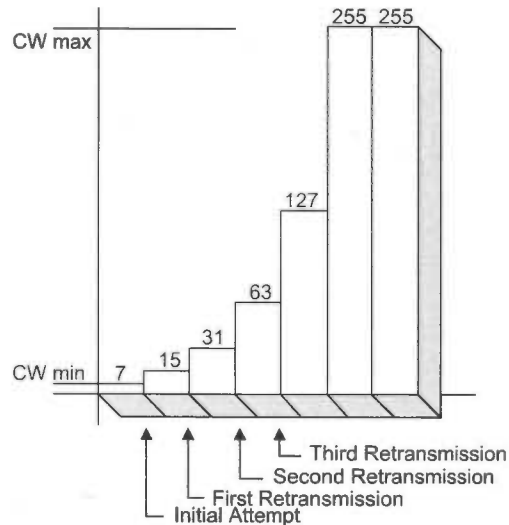


Figure 50—An example of exponential increase of CW

9.2.5 DCF access procedure

The CSMA/CA access method is the foundation of the DCF. The operational rules vary slightly between the DCF and the PCF.

9.2.5.1 Basic access

Basic access refers to the core mechanism a STA uses to determine whether it may transmit.

In general, a STA may transmit a pending MPDU when it is operating under the DCF access method, either in the absence of a PC, or in the CP of the PCF access method, when the STA determines that the medium is idle for greater than or equal to a DIFS period, or an EIFS period if the immediately preceding medium-busy event was caused by detection of a frame that was not received at this STA with a correct MAC FCS value. If, under these conditions, the medium is determined by the carrier-sense mechanism to be busy when a STA desires to initiate the initial frame of one of the frame exchanges described in 9.7, exclusive of the CF period, the random backoff algorithm described in 9.2.5.2 shall be followed. There are conditions, specified in 9.2.5.2 and 9.2.5.5, where the random backoff algorithm shall be followed even for the first attempt to initiate a frame exchange sequence.

In a STA having an FH PHY, control of the channel is lost at the dwell time boundary and the STA shall have to contend for the channel after that dwell boundary. It is required that STAs having an FH PHY complete transmission of the entire MPDU and associated acknowledgment (if required) before the dwell time bound-

ary. If, when transmitting or retransmitting an MPDU, there is not enough time remaining in the dwell to allow transmission of the MPDU plus the acknowledgment (if required), the STA shall defer the transmission by selecting a random backoff time, using the present CW (without advancing to the next value in the series). The short retry counter and long retry counter for the MSDU are not affected.

The basic access mechanism is illustrated in Figure 51.

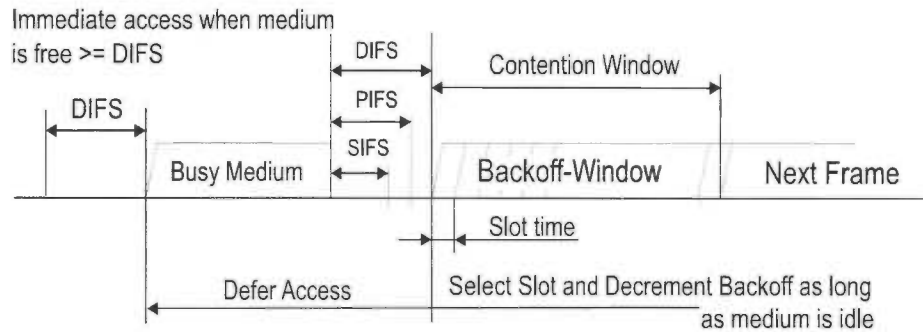


Figure 51—Basic access method

9.2.5.2 Backoff procedure

The backoff procedure shall be invoked for a STA to transfer a frame when finding the medium busy as indicated by either the physical or virtual carrier-sense mechanism (see Figure 52). The backoff procedure shall also be invoked when a transmitting STA infers a failed transmission as defined in 9.2.5.7 or 9.2.8.

To begin the backoff procedure, the STA shall set its Backoff Timer to a random backoff time using the equation in 9.2.4. All backoff slots occur following a DIFS period during which the medium is determined to be idle for the duration of the DIFS period, or following an EIFS period during which the medium is determined to be idle for the duration of the EIFS period following detection of a frame that was not received correctly.

A STA performing the backoff procedure shall use the carrier-sense mechanism (9.2.1) to determine whether there is activity during each backoff slot. If no medium activity is indicated for the duration of a particular backoff slot, then the backoff procedure shall decrement its backoff time by $aSlotTime$.

If the medium is determined to be busy at any time during a backoff slot, then the backoff procedure is suspended; that is, the backoff timer shall not decrement for that slot. The medium shall be determined to be idle for the duration of a DIFS period or EIFS, as appropriate (see 9.2.3), before the backoff procedure is allowed to resume. Transmission shall commence whenever the Backoff Timer reaches zero.

A backoff procedure shall be performed immediately after the end of every transmission with the More Fragments bit set to 0 of an MPDU of type Data, Management, or Control with subtype PS-Poll, even if no additional transmissions are currently queued. In the case of successful acknowledged transmissions, this backoff procedure shall begin at the end of the received ACK frame. In the case of unsuccessful transmissions requiring acknowledgment, this backoff procedure shall begin at the end of the ACK timeout interval. If the transmission is successful, the CW value reverts to $aCWmin$ before the random backoff interval is chosen, and the STA short retry count and/or STA long retry count are updated as described in 9.2.4. This assures that transmitted frames from a STA are always separated by at least one backoff interval.

The effect of this procedure is that when multiple STAs are deferring and go into random backoff, then the STA selecting the smallest backoff time using the random function will win the contention.

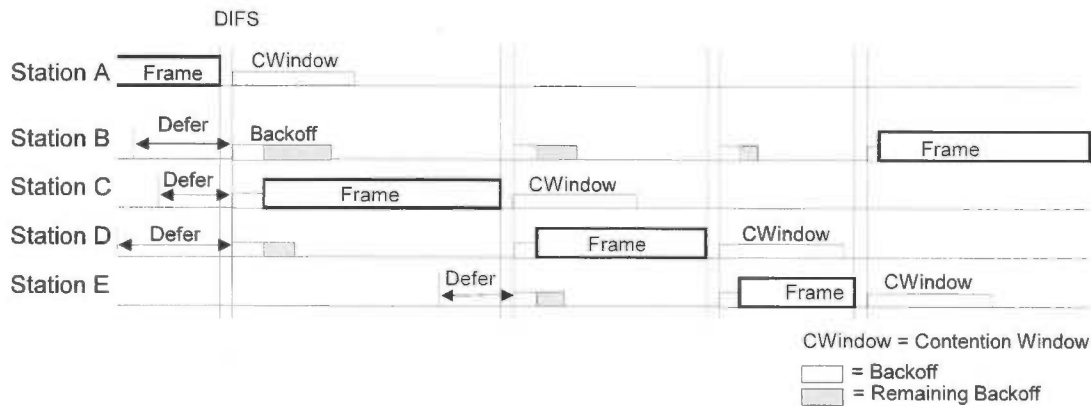


Figure 52—Backoff procedure

In an IBSS, the backoff time for a pending non-beacon or non-ATIM transmission shall not decrement in the period from the target beacon transmission time (TBTT) until the expiration of the ATIM window, and the backoff time for a pending ATIM management frame shall decrement only within the ATIM window. (See Clause 11.) Within an IBSS, a separate backoff interval shall be generated to precede the transmission of a beacon, as described in 11.1.2.2.

9.2.5.3 Recovery procedures and retransmit limits

Error recovery is always the responsibility of the STA that initiates a frame exchange sequence, as defined in 9.7. Many circumstances may cause an error to occur that requires recovery. For example, the CTS frame may not be returned after an RTS frame is transmitted. This may happen due to a collision with another transmission, due to interference in the channel during the RTS or CTS frame, or because the STA receiving the RTS frame has an active virtual carrier-sense condition (indicating a busy medium time period).

Error recovery shall be attempted by retrying transmissions for frame exchange sequences that the initiating STA infers have failed. Retries shall continue, for each failing frame exchange sequence, until the transmission is successful, or until the relevant retry limit is reached, whichever occurs first. STAs shall maintain a short retry count and a long retry count for each MSDU or MMPDU awaiting transmission. These counts are incremented and reset independently of each other.

After an RTS frame is transmitted, the STA shall perform the CTS procedure, as defined in 9.2.5.7. If the RTS transmission fails, the short retry count for the MSDU or MMPDU and the STA short retry count are incremented. This process shall continue until the number of attempts to transmit that MSDU or MMPDU reaches dot11ShortRetryLimit.

After transmitting a frame that requires acknowledgment, the STA shall perform the ACK procedure, as defined in 9.2.8. The short retry count for an MSDU or MMPDU and the STA short retry count shall be incremented every time transmission of a MAC frame of length less than or equal to dot11RTSThreshold fails for that MSDU or MMPDU. This short retry count and the STA short retry count shall be reset when a MAC frame of length less than or equal to dot11RTSThreshold succeeds for that MSDU or MMPDU. The long retry count for an MSDU or MMPDU and the STA long retry count shall be incremented every time transmission of a MAC frame of length greater than dot11RTSThreshold fails for that MSDU or MMPDU. This long retry count and the STA long retry count shall be reset when a MAC frame of length greater than dot11RTSThreshold succeeds for that MSDU or MMPDU. All retransmission attempts for an MSDU or MMPDU that has failed the ACK procedure one or more times shall be made with the Retry field set to 1 in the Data or Management type frame.

Retries for failed transmission attempts shall continue until the short retry count for the MSDU or MMPDU is equal to `dot11ShortRetryLimit` or until the long retry count for the MSDU or MMPDU is equal to `aLongRetryLimit`. When either of these limits is reached, retry attempts shall cease, and the MSDU or MMPDU shall be discarded.

A STA in power-save mode, in an ESS, initiates a frame exchange sequence by transmitting a PS-Poll frame to request data from an AP. In the event that neither an ACK frame nor a data frame is received from the AP in response to a PS-Poll frame, then the STA shall retry the sequence, by transmitting another PS-Poll frame, at its convenience. If the AP sends a data frame in response to a PS-Poll frame, but fails to receive the ACK frame acknowledging this data frame, the next PS-Poll frame from the same STA may cause a retransmission of the last MSDU. This duplicate MSDU shall be filtered at the receiving STA using the normal duplicate frame filtering mechanism. If the AP responds to a PS-Poll by transmitting an ACK frame, then responsibility for the data frame delivery error recovery shifts to the AP because the data is transferred in a subsequent frame exchange sequence, which is initiated by the AP. The AP shall attempt to deliver one MSDU to the STA that transmitted the PS-Poll, using any frame exchange sequence valid for a directed MSDU. If the power save STA that transmitted the PS-Poll returns to Doze state after transmitting the ACK frame in response to successful receipt of this MSDU, but the AP fails to receive this ACK frame, the AP will retry transmission of this MSDU until the relevant retry limit is reached. See Clause 11 for details on filtering of extra PS-Poll frames.

9.2.5.4 Setting and resetting the NAV

STAs receiving a valid frame shall update their NAV with the information received in the Duration/ID field, but only when the new NAV value is greater than the current NAV value and only when the frame is not addressed to the receiving STA. Various additional conditions may set or reset the NAV, as described in 9.3.2.2. When the NAV is reset, a PHY-CCARESET.request shall be issued.

Figure 53 indicates the NAV for STAs that may receive the RTS frame, while other STAs may only receive the CTS frame, resulting in the lower NAV bar as shown (with the exception of the STA to which the RTS was addressed).

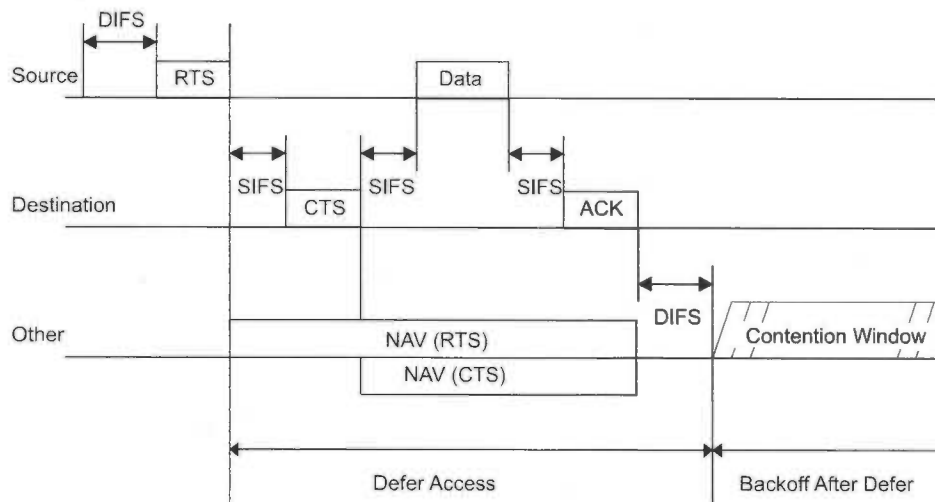


Figure 53—RTS/CTS/data/ACK and NAV setting

A STA that used information from an RTS frame as the most recent basis to update its NAV setting is permitted to reset its NAV if no PHY-RXSTART.indication is detected from the PHY during a period with a duration of $(2 \times aSIFSTime) + (CTS_Time) + (2 \times aSlotTime)$ starting at the PHY-RXEND.indication corresponding to the detection of the RTS frame. The “CTS_Time” shall be calculated using the length of the CTS frame and the data rate at which the RTS frame used for the most recent NAV update was received.

9.2.5.5 Control of the channel

The SIFS is used to provide an efficient MSDU delivery mechanism. Once the STA has contended for the channel, that STA shall continue to send fragments until either all fragments of a single MSDU or MMPDU have been sent, an acknowledgment is not received, or the STA is restricted from sending any additional fragments due to a dwell time boundary. Should the sending of the fragments be interrupted due to one of these reasons, when the next opportunity for transmission occurs the STA shall resume transmission. The algorithm by which the STA decides which of the outstanding MSDUs shall next be attempted after an unsuccessful transmission attempt is beyond the scope of this standard, but any such algorithm shall comply with the restrictions listed in 9.8.

Figure 54 illustrates the transmission of a multiple-fragment MSDU using the SIFS.

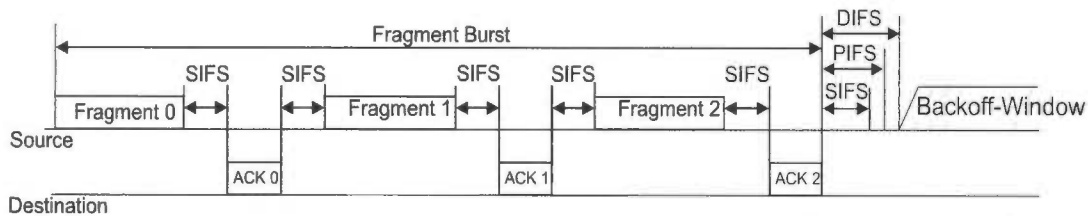


Figure 54—Transmission of a multiple-fragment MSDU using SIFS

When the source STA transmits a fragment, it shall release the channel, then immediately monitor the channel for an acknowledgment as described in 9.2.8.

When the destination STA has finished sending the acknowledgment, the SIFS following the acknowledgment shall be reserved for the source STA to continue (if necessary) with another fragment. The STA sending the acknowledgment shall not transmit on the channel immediately following the acknowledgment.

The process of sending multiple fragments after contending for the channel is defined as a fragment burst.

If the source STA receives an acknowledgment but there is not enough time to transmit the next fragment and receive an acknowledgment due to an impending dwell boundary, the source STA shall contend for the channel at the beginning of the next dwell time.

If the source STA does not receive an acknowledgment frame, it shall attempt to retransmit the failed MPDU or another eligible MPDU, as defined in 9.8, after performing the backoff procedure and the contention process.

After a STA contends for the channel to retransmit a fragment of an MSDU, it shall start with the last fragment that was not acknowledged. The destination STA shall receive the fragments in order (since the source sends them in order, and they are individually acknowledged). It is possible, however, that the destination STA may receive duplicate fragments. It shall be the responsibility of the receiving STA to detect and discard duplicate fragments.

A STA shall transmit after the SIFS only under the following conditions during a fragment burst:

- The STA has just received a fragment that requires acknowledgment.

- The source STA has received an acknowledgment for a previous fragment, has more fragment(s) for the same MSDU to transmit, and there is enough time before the next dwell boundary to send the next fragment and receive its acknowledgment.

The following rules shall also apply:

- When a STA has transmitted a frame other than an initial or intermediate fragment, that STA shall not transmit on the channel following the acknowledgment for that frame, without performing the backoff procedure.
- When an MSDU has been successfully delivered or all retransmission attempts have been exhausted, and the STA has a subsequent MSDU to transmit, then that STA shall perform a backoff procedure.
- Only unacknowledged fragments shall be retransmitted.

9.2.5.6 RTS/CTS usage with fragmentation

The following is a description of using RTS/CTS for a fragmented MSDU or MMPDU. The RTS/CTS frames define the duration of the following frame and acknowledgment. The Duration/ID field in the data and acknowledgment (ACK) frames specifies the total duration of the next fragment and acknowledgment. This is illustrated in Figure 55.

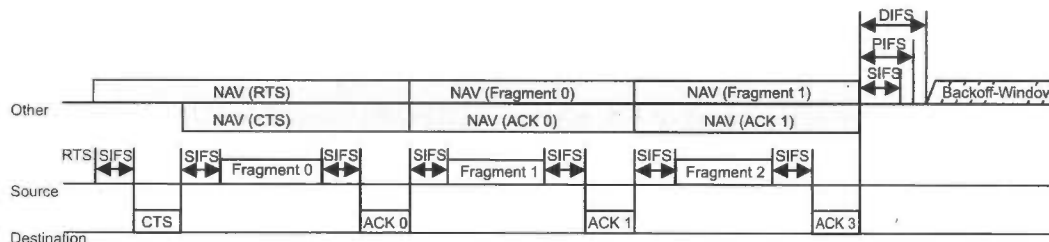


Figure 55—RTS/CTS with fragmented MSDU

Each frame contains information that defines the duration of the next transmission. The duration information from RTS frames shall be used to update the NAV to indicate busy until the end of ACK 0. The duration information from the CTS frame shall also be used to update the NAV to indicate busy until the end of ACK 0. Both Fragment 0 and ACK 0 shall contain duration information to update the NAV to indicate busy until the end of ACK 1. This shall be done by using the Duration/ID field in the Data and ACK frames. This shall continue until the last fragment, which shall have a duration of one ACK time plus one SIFS time, and its ACK, which shall have its Duration/ID field set to zero. Each fragment and ACK acts as a virtual RTS and CTS; therefore no further RTS/CTS frames need to be generated after the RTS/CTS that began the frame exchange sequence even though subsequent fragments may be larger than dot11RTSThreshold. At STAs using a frequency-hopping PHY, when there is insufficient time before the next dwell boundary to transmit the subsequent fragment, the STA initiating the frame exchange sequence may set the Duration/ID field in the last data or management frame to be transmitted before the dwell boundary to the duration of one ACK time plus one SIFS time.

In the case where an acknowledgment is sent but not received by the source STA, STAs that heard the fragment, or ACK, will mark the channel busy for the next frame exchange due to the NAV having been updated from these frames. This is the worst-case situation, and it is shown in Figure 56. If an acknowledgment is not sent by the destination STA, STAs that can only hear the destination STA will not update their NAV and may attempt to access the channel when their NAV updated from the previously received frame reaches zero. All STAs that hear the source will be free to access the channel after their NAV updated from the transmitted fragment has expired.

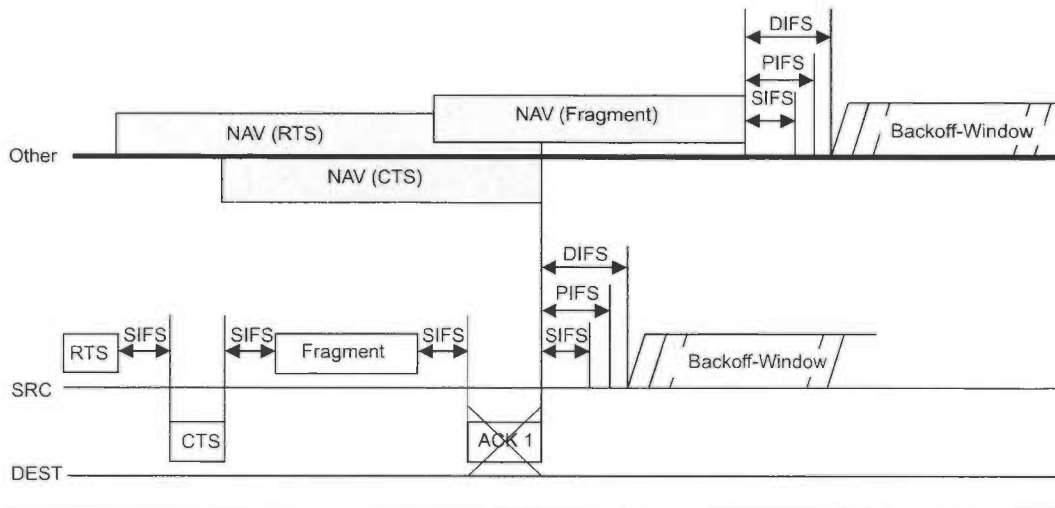


Figure 56—RTS/CTS with transmitter priority and missed acknowledgment

9.2.5.7 CTS procedure

A STA that is addressed by an RTS frame shall transmit a CTS frame after a SIFS period if the NAV at the STA receiving the RTS frame indicates that the medium is idle. If the NAV at the STA receiving the RTS frame indicates the medium is not idle, that STA shall not respond to the RTS frame. The RA field of the CTS frame shall be the value obtained from the TA field of the RTS frame to which this CTS frame is a response. The Duration/ID field in the CTS frame shall be the duration field from the received RTS frame, adjusted by subtraction of aSIFSTime and the number of microseconds required to transmit a CTS frame at the data rate used for the RTS frame to which this CTS frame is a response.

After transmitting an RTS frame, the STA shall wait for a CTSTimeout interval, starting at the PHY-TXEND.confirm. If a PHY-RXSTART.indication does not occur during the CTSTimeout interval, the STA shall conclude that the transmission of the RTS has failed, and this STA shall invoke its backoff procedure upon expiration of the CTSTimeout interval. If a PHY-RXSTART.indication does occur during the CTSTimeout interval, the STA shall wait for the corresponding PHY-RXEND.indication to determine whether the RTS transmission was successful. The recognition of a valid CTS frame sent by the recipient of the RTS frame, corresponding to this PHY-RXEND.indication, shall be interpreted as successful response, permitting the frame sequence to continue (see 9.7). The recognition of anything else, including any other valid frame, shall be interpreted as failure of the RTS transmission. In this instance, the STA shall invoke its backoff procedure at the PHY-RXEND.indication and may process the received frame.

9.2.6 Directed MPDU transfer procedure

A STA shall use an RTS/CTS exchange for directed frames only when the length of the MPDU is greater than the length threshold indicated by the dot11RTSThreshold attribute.

The dot11RTSThreshold attribute shall be a managed object within the MAC MIB, and its value may be set and retrieved by the MAC LME. The value 0 shall be used to indicate that all MPDUs shall be delivered with the use of RTS/CTS. Values of dot11RTSThreshold larger than the maximum MSDU length shall indicate that all MPDUs shall be delivered without RTS/CTS exchanges.

When an RTS/CTS exchange is used, the asynchronous data frame shall be transmitted after the end of the CTS frame and a SIFS period. No regard shall be given to the busy or idle status of the medium when transmitting this data frame.

When an RTS/CTS exchange is not used, the asynchronous data frame shall be transmitted following the success of the basic access procedure. With or without the use of the RTS/CTS exchange procedure, the STA that is the destination of an asynchronous data frame shall follow the ACK procedure.

9.2.7 Broadcast and multicast MPDU transfer procedure

In the absence of a PCF, when broadcast or multicast MPDUs are transferred from a STA with the ToDS bit clear, only the basic access procedure shall be used. Regardless of the length of the frame, no RTS/CTS exchange shall be used. In addition, no ACK shall be transmitted by any of the recipients of the frame. Any broadcast or multicast MPDUs transferred from a STA with a ToDS bit set shall, in addition to conforming to the basic access procedure of CSMA/CA, obey the rules for RTS/CTS exchange, because the MPDU is directed to the AP. The broadcast/multicast message shall be distributed into the BSS. The STA originating the message shall receive the message as a broadcast/multicast message. Therefore, all STAs shall filter out broadcast/multicast messages that contain their address as the source address. Broadcast and multicast MSDUs shall be propagated throughout the ESS.

There is no MAC-level recovery on broadcast or multicast frames, except for those frames sent with the ToDS bit set. As a result, the reliability of this traffic is reduced, relative to the reliability of directed traffic, due to the increased probability of lost frames from interference, collisions, or time-varying channel properties.

9.2.8 ACK procedure

An ACK frame shall be generated as shown in the frame exchange sequences listed in 9.7.

Upon successful reception of a frame of a type that requires acknowledgment with the ToDS bit set, an AP shall generate an ACK frame. An ACK frame shall be transmitted by the destination STA that is not an AP, whenever it successfully receives a unicast frame of a type that requires acknowledgment, but not if it receives a broadcast or multicast frame of such type. After a successful reception of a frame requiring acknowledgment, transmission of the ACK frame shall commence after a SIFS period, without regard to the busy/idle state of the medium.

The source STA shall wait ACKTimeout amount of time without receiving an ACK frame before concluding that the MPDU failed. (See Figure 57.)

After transmitting an MPDU that requires an ACK frame as a response (see 9.7), the STA shall wait for an ACK-Timeout interval, starting at the PHY-TXEND.confirm. If a PHY-RXSTART.indication does not occur during the ACKTimeout interval, the STA concludes that the transmission of the MPDU has failed, and this STA shall invoke its backoff procedure upon expiration of the ACKTimeout interval. If a PHY-RXSTART.indication does occur during the ACKTimeout interval, the STA shall wait for the corresponding PHY-RXEND.indication to determine whether the MPDU transmission was successful. The recognition of a valid ACK frame sent by the recipient of the MPDU requiring acknowledgment, corresponding to this PHY-RXEND.indication, shall be interpreted as successful acknowledgment, permitting the frame sequence to continue, or to end without retries, as appropriate for the particular frame sequence in progress. The recognition of anything else, including any other valid frame, shall be interpreted as failure of the MPDU transmission. In this instance, the STA shall invoke its backoff procedure at the PHY-RXEND.indication and may process the received frame. The sole exception is that recognition of a valid data frame sent by the recipient of a PS-Poll frame shall also be accepted as successful acknowledgment of the PS-Poll frame.

9.2.9 Duplicate detection and recovery

Since MAC-level acknowledgments and retransmissions are incorporated into the protocol, there is the possibility that a frame may be received more than once. Such duplicate frames shall be filtered out within the destination MAC.

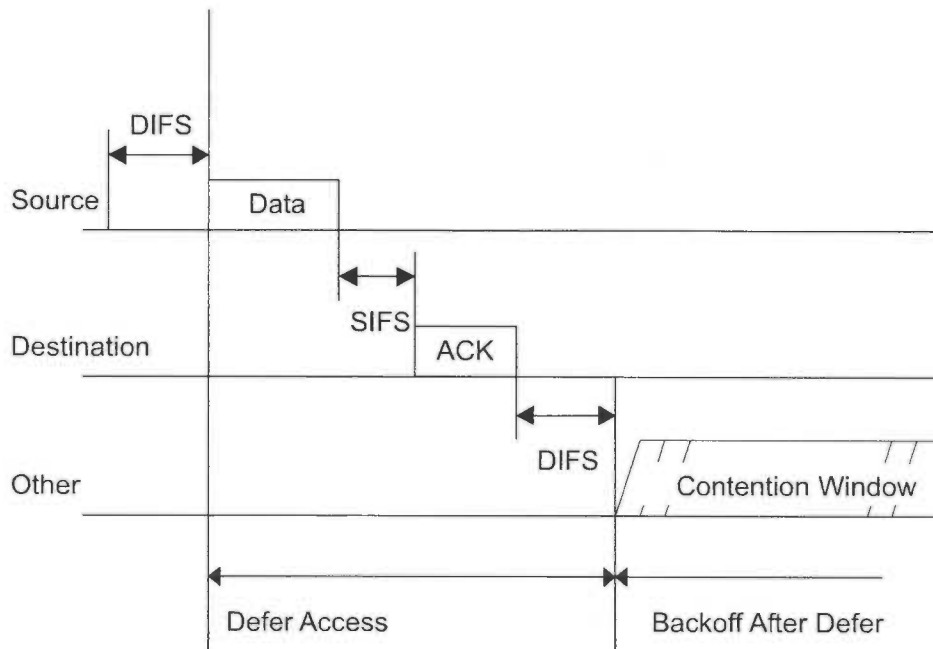


Figure 57—Directed data/ACK MPDU

Duplicate frame filtering is facilitated through the inclusion of a Sequence Control field (consisting of a sequence number and fragment number) within data and management frames. MPDUs that are part of the same MSDU shall have the same sequence number, and different MSDUs shall (with a high probability) have a different sequence number.

The sequence number is generated by the transmitting STA as an incrementing sequence of integers.

The receiving STA shall keep a cache of recently received <Address 2, sequence-number, fragment-number> tuples. A receiving STA is required to keep only the most recent cache entry per Address 2-sequence-number pair, storing only the most recently received fragment number for that pair. A receiving STA may omit tuples obtained from broadcast/multicast or ATIM frames from the cache.

A destination STA shall reject as a duplicate frame any frame that has the Retry bit set in the Frame Control field and that matches an <Address 2, sequence-number, and fragment-number> tuple of an entry in the cache.

There is a small possibility that a frame may be improperly rejected due to such a match; however, this occurrence would be rare and simply results in a lost frame (similar to an FCS error in other LAN protocols).

The destination STA shall perform the ACK procedure on all successfully received frames requiring acknowledgment, even if the frame is discarded due to duplicate filtering.

9.2.10 DCF timing relations

The relationships between the IFS specifications are defined as time gaps on the medium. The associated attributes are provided by the specific PHY. (See Figure 58.)

All timings that are referenced from the end of the transmission are referenced from the end of the last symbol of a frame on the medium. The beginning of transmission refers to the first symbol of the next frame on the medium.

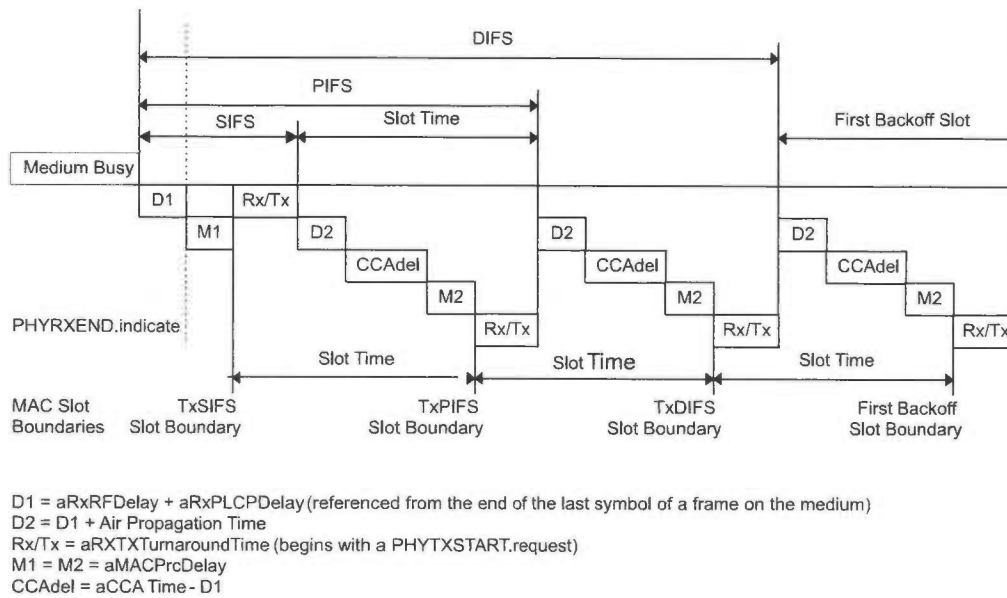


Figure 58—DCF timing relationships

aSIFSTime and aSlotTime are fixed per PHY.

aSIFSTime is: $aRxRFDelay + aRxPLCPDelay + aMACProcessingDelay + aRxTxTurnaroundTime$.

aSlotTime is: $aCCATime + aRxTxTurnaroundTime + aAirPropagationTime + aMACProcessingDelay$.

The PIFS and DIFS are derived by the following equations, as illustrated in Figure 58.

$$PIFS = aSIFSTime + aSlotTime$$

$$DIFS = aSIFSTime + 2 \times aSlotTime$$

The EIFS is derived from the SIFS and the DIFS and the length of time it takes to transmit an ACK Control frame at 1 Mbit/s by the following equation:

$$EIFS = aSIFSTime + (8 \times ACKSize) + aPreambleLength + aPLCPHeaderLngth + DIFS$$

where

ACKSize is the length, in bytes, of an ACK frame; and

$(8 \times ACKSize) + aPreambleLength + aPLCPHeaderLngth$ is expressed in microseconds required to transmit at the PHY's lowest mandatory rate.

Figure 58 illustrates the relation between the SIFS, PIFS, and DIFS as they are measured on the medium and the different MAC slot boundaries TxSIFS, TxPIFS, and TxDIFS. These slot boundaries define when the transmitter shall be turned on by the MAC to meet the different IFS timings on the medium, after subsequent detection of the CCA result of the previous slot time.

The following equations define the MAC Slot Boundaries, using attributes provided by the PHY, which are such that they compensate for implementation timing variations. The starting reference of these slot boundaries is again the end of the last symbol of the previous frame on the medium.

$$TxSIFS = SIFS - aRxTxTurnaroundTime$$

$$TxPIFS = TxSIFS + aSlotTime$$

$$TxDIFS = TxSIFS + 2 \times aSlotTime.$$

The tolerances are specified in the PLME SAP Interface Specification (10.4), and shall only apply to the SIFS specification, so that tolerances shall not accumulate.

9.3 PCF

The PCF provides contention-free frame transfer. The PC shall reside in the AP. It is an option for an AP to be able to become the PC. All STAs inherently obey the medium access rules of the PCF, because these rules are based on the DCF, and they set their NAV at the beginning of each CFP. The operating characteristics of the PCF are such that all STAs are able to operate properly in the presence of a BSS in which a PC is operating, and, if associated with a point-coordinated BSS, are able to receive all frames sent under PCF control. It is also an option for a STA to be able to respond to a contention-free poll (CF-Poll) received from a PC. A STA that is able to respond to CF-Polls is referred to as being CF-Pollable, and may request to be polled by an active PC. CF-Pollable STAs and the PC do not use RTS/CTS in the CFP. When polled by the PC, a CF-Pollable STA may transmit only one MPDU, which can be to any destination (not just to the PC), and may “piggyback” the acknowledgment of a frame received from the PC using particular data frame subtypes for this transmission. If the data frame is not in turn acknowledged, the CF-Pollable STA shall not retransmit the frame unless it is polled again by the PC, or it decides to retransmit during the CP. If the addressed recipient of a CF transmission is not CF-Pollable, that STA acknowledges the transmission using the DCF acknowledgment rules, and the PC retains control of the medium. A PC may use contention-free frame transfer solely for delivery of frames to STAs, and never to poll non-CF-Pollable STAs.

A PC may perform a backoff on retransmission of an unacknowledged frame during the CFP. A PC that is maintaining a polling list may retry the unacknowledged frame the next time the particular AID is at the top of the polling list.

A PC may retransmit an unacknowledged frame during the CFP after a PIFS time.

When more than one point-coordinated BSS is operating on the same PHY channel in overlapping space, the potential exists for collisions between PCF transfer activities by the independent PCs. The rules under which multiple, overlapping point-coordinated BSSs may coexist are presented in 9.3.3.2. As shown in Figure 47, the PCF is built on top of the CSMA/CA-based DCF, by utilizing the access priority provisions provided by this scheme. An active PC shall be located at an AP, which restricts PCF operation to infrastructure networks. PCF is activated at a PC-capable AP by setting the CFPMaxDuration parameter in the CF Parameter Set of the MLMESStart.request to a non-zero value.

Data frames sent during under the DCF shall use the data subtypes Data or Null Function. Data frames sent by, or in response to polling by, the PC during the CFP shall use the appropriate data subtypes based upon the following usage rules:

- Data+CF-Poll, Data+CF-Ack+CF-Poll, CF-Poll, and CF-Ack+CF-Poll shall only be sent by a PC.
- Data, Data+CF-Ack, Null Function, and CF-Ack may be sent by a PC or by any CF-Pollable STA.

STAs receiving Data type frames shall only consider the frame body as the basis of a possible indication to LLC, if the frame is of subtype Data, Data+CF-Ack, Data+CF-Poll, or Data+CF-Ack+CF-Poll. CF-Pollable STAs shall interpret all subtype bits of received Data type frames for CF purposes, but shall only inspect the frame body if the frame is of subtype Data, Data+CF-Ack, Data+CF-Poll, or Data+CF-Ack+CF-Poll.

9.3.1 CFP structure and timing

The PCF controls frame transfers during a CFP. The CFP shall alternate with a CP, when the DCF controls frame transfers, as shown in Figure 59. Each CFP shall begin with a Beacon frame that contains a DTIM element (hereafter referred to as a “DTIM”). The CFPs shall occur at a defined repetition rate, which shall be synchronized with the beacon interval as specified in the following paragraphs.

The PC generates CFPs at the *contention-free repetition rate* (CFPRate), which is defined as a number of DTIM intervals. The PC shall determine the CFPRate (depicted as a repetition interval in the illustrations in Figure 59 and Figure 60) to use from the CFPRate parameter in the CF Parameter Set. This value, in units of DTIM intervals, shall be communicated to other STAs in the BSS in the CFPeriod field of the CF Parameter Set element of Beacon frames. The CF Parameter Set element shall only be present in Beacon and Probe Response frames transmitted by STAs containing an active PC.

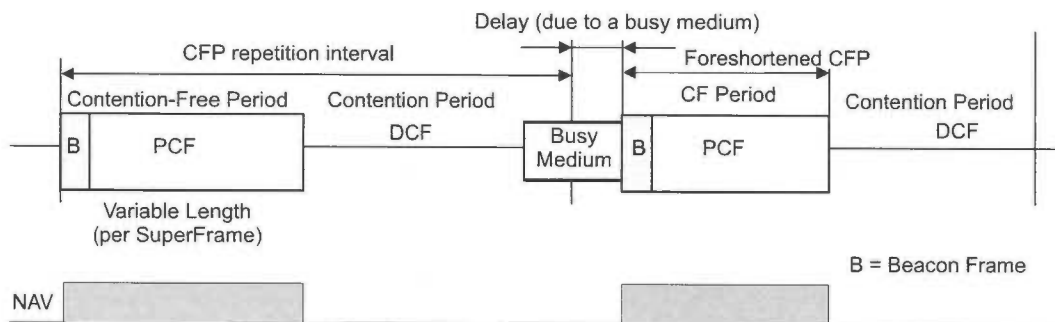


Figure 59—CFP/CP alternation

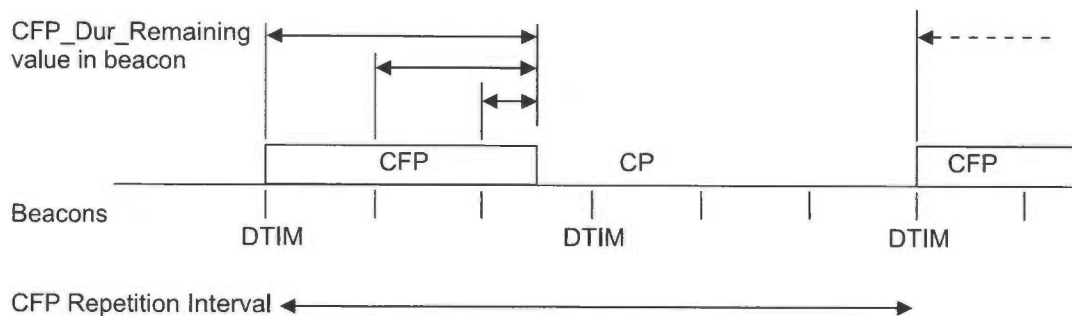


Figure 60—Beacons and CFPs

The length of the CFP is controlled by the PC, with maximum duration specified by the value of the CFP-MaxDuration Parameter in the CF Parameter Set at the PC. Neither the maximum duration nor the actual duration (signaled by transmission of a Control frame of subtype CF-End or CF-End+ACK by the PC) is

constrained to be a multiple of the beacon interval. If the CFP duration is greater than the beacon interval, the PC shall transmit beacons at the appropriate times during the CFP (subject to delay due to traffic at the nominal times, as with all beacons). The CF Parameter Set element in all beacons at the start of, or within, a CFP shall contain a nonzero value in the CFPDurRemaining field. This value, in units of TU, shall specify the maximum time from the transmission of this beacon to the end of this CFP. The value of the CFPDurRemaining field shall be zero in beacons sent during the CP. An example of these relationships is illustrated in Figure 60, which shows a case where the CFP is two DTIM intervals, the DTIM interval is three beacon intervals, and the aCFPMaxDuration value is approximately 2.5 beacon intervals.

The PC may terminate any CFP at or before the aCFPMaxDuration, based on available traffic and size of the polling list. Because the transmission of any beacon may be delayed due to a medium busy condition at the nominal beacon transmission time, a CFP may be foreshortened by the amount of the delay. In the case of a busy medium due to DCF traffic, the beacon shall be delayed for the time required to complete the current DCF frame exchange. In cases where the beacon transmission is delayed, the CFPDurRemaining value in the beacon at the beginning of the CFP shall specify a time that causes the CFP to end no later than TBTT plus the value of aCFPMaxDuration. This is illustrated in Figure 61.

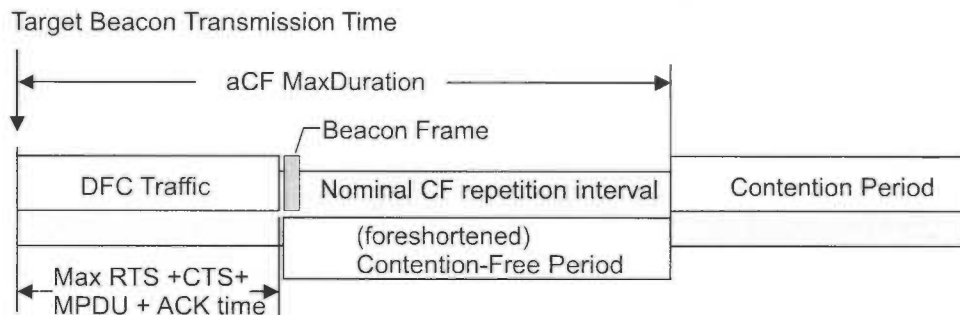


Figure 61—Example of delayed beacon and foreshortened CFP

9.3.2 PCF access procedure

The contention-free transfer protocol is based on a polling scheme controlled by a PC operating at the AP of the BSS. The PC gains control of the medium at the beginning of the CFP and attempts to maintain control for the entire CFP by waiting a shorter time between transmissions than the STAs using the DCF access procedure. All STAs in the BSS (other than the PC) set their NAVs to the CFPMaxDuration value at the nominal start time of each CFP. This prevents most contention by preventing non-pollled transmissions by STAs whether or not they are CF-Pollable. Acknowledgment of frames sent during the CFP may be accomplished using Data+CF-ACK, CF-ACK, Data+CF-ACK+CF-Poll (only on frames transmitted by the PC), or CF-ACK+CF-Poll (only on frames transmitted by the PC) frames in cases where a Data (or Null) frame immediately follows the frame being acknowledged, thereby avoiding the overhead of separate ACK Control frames. Non-CF-Pollable or unpollled CF-Pollable STAs acknowledge frames during the CFP using the DCF ACK procedure.

9.3.2.1 Fundamental access

At the nominal beginning of each CFP, the PC shall sense the medium. When the medium is determined to be idle for one PIFS period, the PC shall transmit a Beacon frame containing the CF Parameter Set element and a DTIM element.

After the initial beacon frame, the PC shall wait for at least one SIFS period, and then transmit one of the following: a data frame, a CF-Poll frame, a Data+CF-Poll frame, or a CF-End frame. If the CFP is null, i.e.,

there is no traffic buffered and no polls to send at the PC, a CF-End frame shall be transmitted immediately after the initial beacon.

STAs receiving directed, error-free frames from the PC are expected to respond after a SIFS period, in accordance with the transfer procedures defined in 9.3.3. If the recipient STA is not CF-Pollable, the response to receipt of an error-free data frame shall always be an ACK frame.

9.3.2.2 NAV operation during the CFP

The mechanism for handling the NAV during the CFP is designed to facilitate the operation of overlapping CFP coordinated infrastructure BSSs. The mechanism by which infrastructure BSSs coordinate their CFPs is beyond the scope of this standard.

Each STA, except the STA with the PC, shall preset its NAV to the CFPMaxDuration value (obtained from the CF Parameter Set element in beacons from this PC) at each target beacon transmission time (TBTT) (see Clause 11) at which a CFP is scheduled to start (based on the CFPPeriod field in the CF Parameter Set element of the Beacon frames from this PC). Each non-PC STA shall update its NAV using the CFPDurRemaining value in any error-free CF Parameter Set element of the Beacon frame that the STA receives. This includes CFPDurRemaining values in CF Parameter Set elements from Beacon frames received from other (overlapping) BSSs.

These actions prevent STAs from taking control of the medium during the CFP, which is especially important in cases where the CFP spans multiple medium-occupancy intervals, such as dwell periods of an FH PHY. This setting of the NAV also reduces the risk of hidden STAs determining the medium to be idle for a DIFS period during the CFP and possibly corrupting a transmission in progress.

A STA joining a BSS operating with a PC shall use the information in the CFPDurRemaining element of the CF parameter set of any received Beacon or Probe Response frames to update its NAV prior to initiating any transmissions.

The PC shall transmit a CF-End or CF-End+ACK frame at the end of each CFP. A STA that receives either of these frames, from any BSS, shall reset its NAV.

9.3.3 PCF transfer procedure

Frame transfers under the PCF typically consist of frames alternately sent from the AP/PC and sent to the AP/PC. During the CFP, the ordering of these transmissions, and the STA allowed to transmit frames to the PC at any given point in time, shall be controlled by the PC. Figure 62 depicts a frame transfer during a typical CFP. The rules under which this frame transfer takes place are detailed in the following subclauses.

In a STA having an FH PHY, control of the channel is lost at a dwell time boundary. It is required that the current MPDU transmission and the accompanying acknowledgment of the MPDU be transmitted before the dwell time boundary. After having been polled by the PC, if there is not enough time remaining in the dwell to allow transmission of the MPDU plus the acknowledgment, the STA shall defer the transmission of the MPDU and shall transmit a Null frame or CF-ACK frame. The short retry counter and long retry counter for the MSDU shall not be affected.

MaxMPDUTime is the time to transmit the maximum-sized MAC frame, expanded by WEP, plus the time to transmit the PHY preamble, header, trailer, and expansion bits, if any. In a STA having an FH PHY, the PC shall not transmit a CF-Poll to a STA if there is insufficient time remaining before the dwell boundary for the STA to respond with a Null frame or CF-ACK frame.

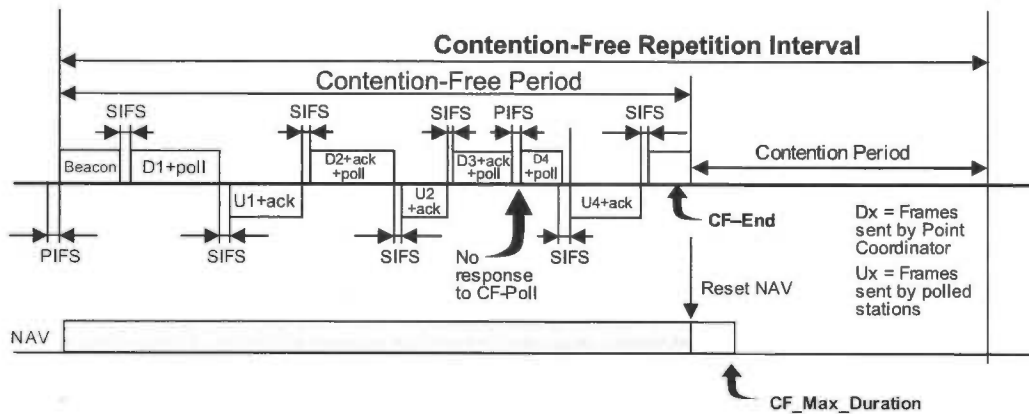


Figure 62—Example of PCF frame transfer

9.3.3.1 PCF transfers when the PCF STA is transmitter or recipient

The PC shall transmit frames between the Beacon that starts the CFP and the CF-End using the SIFS except in cases where a transmission by another STA is expected by the PC and a SIFS period elapses without the receipt of the expected transmission. In such cases the PC may send its next pending transmission as soon as one PIFS after the end of its last transmission. This permits the PC to retain control of the medium in the presence of an overlapping BSS. The PC may transmit any of the following frame types to CF-Pollable STAs:

- Data, used to send data from the PC when the addressed recipient is not being polled and there is no previous frame to acknowledge;
- Data+CF-ACK, used to send data from the PC when the addressed recipient is not being polled and the PC needs to acknowledge the receipt of a frame received from a CF-Pollable STA a SIFS period before starting this transmission;
- Data+CF-Poll, used to send data from the PC when the addressed recipient is the next STA to be permitted to transmit during this CFP and there is no previous frame to acknowledge;
- Data+CF-ACK+CF-Poll, used to send data from the PC when the addressed recipient is the next STA to be permitted to transmit during this CFP and the PC needs to acknowledge the receipt of a frame received from a CF-Pollable STA a SIFS period before starting this transmission;
- CF-Poll, used when the PC is not sending data to the addressed recipient, but the addressed recipient is the next STA to be permitted to transmit during this CFP and there is no previous frame to acknowledge;
- CF-ACK+CF-Poll, used when the PC is not sending data to the addressed recipient but the addressed recipient is the next STA to be permitted to transmit during this CFP and the PC needs to acknowledge the receipt of a frame from a CF-Pollable STA a SIFS period before starting this transmission;
- CF-ACK, used when the PC is not sending data to, or polling, the addressed recipient, but the PC needs to acknowledge receipt of a frame from a CF-Pollable STA a SIFS period before starting this transmission (useful when the next transmission by the PC is a management frame, such as a beacon); or
- Any management frame that is appropriate for the AP to send under the rules for that frame type.

The PC may transmit data or management frames to non-CF-Pollable, non-power-save STAs during the CFP. These STAs shall acknowledge receipt with ACK frames after a SIFS, as with the DCF. The PC may also transmit broadcast or multicast frames during the CFP. Because the Beacon frame that initiates the CFP

contains a DTIM element, if there are associated STAs using power-save mode, the broadcasts and multicasts buffered shall be sent immediately after any beacon containing a TIM element with a DTIM count field with a value of 0.

A CF-Pollable STA that receives a directed data frame of any subtype that includes CF-Poll may transmit one data frame a SIFS period after receiving the CF-Poll. CF-Pollable STAs shall ignore, but not reset, their NAV when performing transmissions in response to a CF-Poll.

Non-CF-Pollable STAs that receive a directed frame during the CFP shall transmit an ACK, but shall not reset their NAV.

For frames that require MAC-level acknowledgment, CF-Pollable STAs that received a CF-Poll (of any type) may perform this acknowledgment using the Data+CF-ACK subtype in the response to the CF-Poll. For example, the U1 frame in Figure 62 contains the acknowledgment to the preceding D1 frame. The D2 frame contains the acknowledgment to the preceding U1 frame. The PC may use the CF-ACK subtypes to acknowledge a received frame even if the data frame sent with the CF-ACK subtype is addressed to a different STA than the one being acknowledged. CF-Pollable STAs that are expecting an acknowledgment shall interpret the subtype of the frame (if any) sent by the PC a SIFS period after that STA's transmission to the PC. If a frame that requires MAC-level acknowledgment is received by a non-CF-Pollable STA, that STA shall not interpret the CF-Poll indication (if any), and shall acknowledge the frame by sending an ACK Control frame after a SIFS period.

The lengths of the frames may be variable, only bounded by the frame and/or fragment length limitations that apply for the BSS. If a CF-Pollable STA does not respond to a CF-Poll (of any type) within the SIFS period following a transmission from the PC, or a non-CF-Pollable STA does not return the ACK frame within a SIFS period following a transmission from the PC that requires acknowledgment, then the PC shall resume control and may transmit its next frame after a PIFS period from the end of the PC's last transmission.

A CF-Pollable STA shall always respond to a CF-Poll directed to its MAC address and received without error. If the STA has no frame to send when polled, the response shall be a Null frame. If the STA has no frame to send when polled, but an acknowledgment is required for the frame that conveyed the CF-Poll, the response shall be a CF-ACK (no data) frame. The null response is required to permit a "no-traffic" situation to be distinguished from a collision between overlapping PCs.

The CFP shall end when the CFP_{DurRemaining} time has elapsed since the Beacon frame originating the CFP or when the PC has no further frames to transmit nor STAs to poll. In either case, the end of the CFP shall be signaled by the transmission of a CF-End by the PC. If there is a received frame that requires acknowledgment at the time the CF-End is to be transmitted, the PC shall transmit a CF-End+ACK frame instead. All STAs of the BSS receiving a CF-End or CF-End+ACK shall reset their NAVs so they may attempt to transmit during the CP.

9.3.3.2 Operation with overlapping point-coordinated BSSs

Because the PCF operates without the CSMA/CA contention window randomization and backoff of the DCF, there is a risk of repeated collisions if multiple, overlapping, point-coordinated BSSs are operating on the same PHY channel, and their CFP Rates and beacon intervals are approximately equal. To minimize the risk of significant frame loss due to CF collisions, the PC shall use a DIFS plus a random backoff delay (with CW in the range of 1 to aCW_{min}) to start a CFP when the initial beacon is delayed because of deferral due to a busy medium. The PC may optionally use this backoff during the CFP prior to retransmitting an unacknowledged, directed data or management frame.

To further reduce the susceptibility to inter-PC collisions, the PC shall require that the medium be determined as being idle for a DIFS period plus a random (over a range of 1 to aCW_{min}) number of slot times once every aMediumOccupancyLimit TU during the CFP. This results in loss of control of the medium to

overlapping BSS or hidden STA traffic, because the STAs in this BSS are prevented from transmitting by their NAV setting to CFPMaxDuration or CFPDurRemaining. For operation of the PCF in conjunction with an FH PHY, aMediumOccupancyLimit shall be set equal to the dwell time. For operation in conjunction with other PHY types, aMediumOccupancyLimit may be set equal to CFPMaxDuration, unless extra protection against PCF collisions is desired. The aMediumOccupancyLimit is also useful for compliance in regulatory domains that impose limits on continuous transmission time by a single STA as part of a spectrum etiquette.

9.3.3.3 CFPMaxDuration limit

The value of CFPMaxDuration shall be limited to allow coexistence between contention and contention-free traffic.

The minimum value for CFPMaxDuration is two times MaxMPDUTime plus the time required to send the initial Beacon frame and the CF-End frame of the CFP. This may allow sufficient time for the AP to send one data frame to a STA, while polling that STA, and for the polled STA to respond with one data frame.

The maximum value for CFPMaxDuration is the duration of $(\text{BeaconPeriod} \times \text{DTIMPeriod} \times \text{CFPRate})$ minus $[\text{MaxMPDUTime} \text{ plus } (2 \times \text{aSIFSTime}) \text{ plus } (2 \times \text{aSlotTime}) \text{ plus } (8 \times \text{ACKSize})]$, expressed in microseconds, when operating with a contention window of aCWmin. MaxMPDUTime is the time to transmit the maximum-sized MAC frame, expanded by WEP, plus the time to transmit the PHY preamble, header, trailer, and expansion bits, if any. This allows sufficient time to send at least one data frame during the CP.

9.3.3.4 Contention-Free usage rules

A PC may send broadcast or multicast frames, and directed data or management frames to any active STA, as well as to CF-Pollable power save STAs. During the CFP, CF-Pollable STAs shall acknowledge after a SIFS period, the receipt of each Data+CF-Poll frame or Data+CF-ACK+CF-Poll frame using Data+CF-Ack or CF-Ack (no data) frames, the receipt of each CF_Poll (no data) using Data or Null (no data), and the receipt of all other data and management frames using ACK Control frames. Non-CF-Pollable STAs shall acknowledge receipt of data and management frames using ACK Control frames sent after a SIFS period. This non-CF-Pollable operation is the same as that already employed by such STAs for DCF operation.

When polled by the PCF (Data+CF-Poll, Data+CF-ACK+CF-Poll, CF-Poll, or CF-ACK+CF-Poll) a CF-Pollable STA may send one data frame to any destination. Such a frame directed to or through the PC STA shall be acknowledged by the PC, using the CF-ACK indication (Data+CF-ACK, Data+CF-ACK+CF-Poll, CF-ACK, CF-ACK+CF-Poll, or CF-End+ACK) sent after a SIFS. Such a frame directed to a non-CF-Pollable STA shall be acknowledged using an ACK Control frame sent after a SIFS period. A polled CF-Pollable STA with neither a data frame nor an acknowledgment to send shall respond by transmitting a Null frame after a SIFS period. A polled CF-Pollable STA with insufficient time before the end of the CFP or current medium occupancy limit, to send its queued MPDU and receive an acknowledgment, shall respond by transmitting a Null frame, or a CF-ACK frame if polled using Data+CF-Poll or Data+CF-ACK+CF-Poll, after a SIFS period. The CF-Pollable STA may set the More Data bit in its response to permit the PC to distinguish between an empty STA queue and a response due to insufficient time to transfer an MPDU.

The PC shall not issue frames with a subtype that includes CF-Polls if insufficient time remains in the current CFP to permit the polled STA to transmit a data frame containing a minimum length MPDU.

9.3.4 Contention-Free polling list

If the PC supports use of the CFP for inbound frame transfer as well as for frame delivery, the PC shall maintain a "polling list" for use in selecting STAs that are eligible to receive CF-Polls during CFPs. The polling list functional characteristics are defined below. If the PC supports the use of the CFP solely for frame delivery, the PC does not require a polling list, and shall never generate data frames with a subtype that includes

CF-Poll. The form of contention-free support provided by the PC is identified in the Capability Information field of Beacon, Association Response, Reassociation Response, and Probe Response management frames, which are sent from APs. Any such frames sent by STAs, as in noninfrastructure networks, shall always have these bits set to zero.

The polling list is used to force the polling of CF-Pollable STAs, whether or not the PC has pending traffic to transmit to those STAs. The polling list may be used to control the use of Data+CF-Poll and Data+CF-ACK+CF-Poll types for transmission of data frames being sent to CF-Pollable STAs by the PC. The polling list is a *logical* construct, which is not exposed outside of the PC. A minimum set of polling list maintenance techniques are required to ensure interoperability of arbitrary CF-Pollable STAs in BSSs controlled by arbitrary access points with active PCs. APs may also implement additional polling list maintenance techniques that are outside the scope of this standard.

9.3.4.1 Polling list processing

The PC shall send a CF-Poll to at least one STA during each CFP when there are entries in the polling list. During each CFP, the PC shall issue polls to a subset of the STAs on the polling list in order by ascending AID value.

While time remains in the CFP, all CF frames have been delivered, and all STAs on the polling list have been polled, the PC may generate one or more CF-Polls to *any* STAs on the polling list. While time remains in the CFP, all CF frames have been delivered, and all STAs on the polling list have been polled, the PC *may* send data or management frames to *any* STAs.

In order to gain maximum efficiency from the CFP, and the ability to piggyback acknowledgments on successor data frames in the opposite direction, the PC should generally use Data+CF-Poll and Data+CF-ACK+CF-Poll types for each data frame transmitted while sufficient time for the potential response to the CF-Poll remains in the CFP.

9.3.4.2 Polling list update procedure

A STA indicates its CF-Pollability using the CF-Pollable subfield of the Capability Information field of Association Request and Reassociation Request frames. If a STA desires to change the PC's record of CF-Pollability, that STA shall perform a reassociation. During association, a CF-Pollable STA may also request to be placed on the polling list for the duration of its association, or by setting the CF-Poll Request subfield in the Capability Information field. If a CF-Pollable STA desires never to be placed on the polling list, that STA shall perform Association with both the CF-Pollable subfield false and the CF-Poll Request subfield true. Never being polled is useful for CF-Pollable STAs that normally use power-save mode, permitting them to receive buffered traffic during the CFP (since they have to be awake to receive the DTIM that initiated the CFP), but not requiring them to stay awake to receive CF-Polls when they have no traffic to send. If a STA desires to be removed from the polling list, that STA shall perform a reassociation.

CF-Pollable STAs that are not on the polling list, but did not request never to be polled during their most recent association, may be dynamically placed on the polling list by the PC to handle bursts of frame transfer activity by that STA.

9.4 Fragmentation

The MAC may fragment and reassemble directed MSDUs or MMPDUs. The fragmentation and defragmentation mechanisms allow for fragment retransmission.

The length of a fragment MPDU shall be an equal number of octets for all fragments except the last, which may be smaller. The length of a fragment MPDU shall always be an even number of octets, except for the

last fragment of an MSDU or MMPDU, which may be either an even or an odd number of octets. The length of a fragment shall never be larger than `aFragmentationThreshold` unless WEP is invoked for the MPDU. If WEP is active for the MPDU, then the MPDU shall be expanded by IV and ICV (see 8.2.5); this may result in a fragment larger than `aFragmentationThreshold`.

When data is to be transmitted, the number of octets in the fragment (before WEP processing) shall be determined by `aFragmentationThreshold` and the number of octets in the MPDU that have yet to be assigned to a fragment at the instant the fragment is constructed for the first time. Once a fragment is transmitted for the first time, its frame body content and length shall be fixed until it is successfully delivered to the immediate receiving STA. A STA shall be capable of receiving fragments of arbitrary length.

If a fragment requires retransmission, its frame body content and length shall remain fixed for the lifetime of the MSDU or MMPDU at that STA. After a fragment is transmitted once, contents and length of that fragment are not allowed to fluctuate to accommodate the dwell time boundaries. Each fragment shall contain a Sequence Control field, which is comprised of a sequence number and fragment number. When a STA is transmitting an MSDU or MMPDU, the sequence number shall remain the same for all fragments of that MSDU or MMPDU. The fragments shall be sent in order of lowest fragment number to highest fragment number, where the fragment number value starts at zero, and increases by one for each successive fragment. The Frame Control field also contains a bit, the More Fragments bit, that is equal to zero to indicate the last (or only) fragment of the MSDU or MMPDU.

The source STA shall maintain a transmit MSDU timer for each MSDU being transmitted. The attribute `aMaxTransmitMSDULifetime` specifies the maximum amount of time allowed to transmit an MSDU. The timer starts on the attempt to transmit the first fragment of the MSDU. If the timer exceeds `aMaxTransmitMSDULifetime`, then all remaining fragments are discarded by the source STA and no attempt is made to complete transmission of the MSDU.

9.5 Defragmentation

Each fragment contains information to allow the complete MSDU or MMPDU to be reassembled from its constituent fragments. The header of each fragment contains the following information that is used by the destination STA to reassemble the MSDU or MMPDU:

- Frame type
- Address of the sender, obtained from the Address2 field
- Destination address
- *Sequence Control field*: This field allows the destination STA to check that all incoming fragments belong to the same MSDU or MMPDU, and the sequence in which the fragments should be reassembled. The sequence number within the Sequence Control field remains the same for all fragments of an MSDU or MMPDU, while the fragment number within the Sequence Control field increments for each fragment.
- *More Fragments indicator*: Indicates to the destination STA that this is not the last fragment of the MSDU or MMPDU. Only the last or sole fragment of the MSDU or MMPDU shall have this bit set to zero. All other fragments of the MSDU or MMPDU shall have this bit set to one.

The destination STA shall reconstruct the MSDU or MMPDU by combining the fragments in order of fragment number subfield of the Sequence Control field. If WEP has been applied to the fragment, it shall be decrypted before the fragment is used for defragmentation of the MSDU or MMPDU. If the fragment with the More Fragments bit set to zero has not yet been received, then the destination STA knows that the MSDU or MMPDU is not yet complete. As soon as the STA receives the fragment with the More Fragments bit set to zero, the STA knows that no more fragments may be received for the MSDU or MMPDU.

All STAs shall support the concurrent reception of fragments of at least three MSDUs or MMPDUs. Note that a STA receiving more than three fragmented MSDUs or MMPDUs concurrently may experience a significant increase in the number of frames discarded.

The destination STA shall maintain a Receive Timer for each MSDU or MMPDU being received, for a minimum of three MSDUs or MMPDUs. The STA may implement additional timers to be able to receive additional concurrent MSDUs or MMPDUs. The receiving STA shall discard all fragments that are part of an MSDU or MMPDU for which a timer is not maintained. There is also an attribute, `aMaxReceiveLifetime`, that specifies the maximum amount of time allowed to receive an MSDU. The receive MSDU or MMPDU timer starts on the reception of the first fragment of the MSDU or MMPDU. If the receive MSDU timer exceeds `aMaxReceiveLifetime`, then all received fragments of this MSDU or MMPDU are discarded by the destination STA. If additional fragments of a directed MSDU or MMPDU are received after its `aMaxReceiveLifetime` is exceeded, those fragments shall be acknowledged and discarded.

To properly reassemble MPDUs into an MSDU or MMPDU, a destination STA shall discard any duplicated fragments received. A STA shall discard duplicate fragments as described in 9.2.9. However, an acknowledgment shall be sent in response to a duplicate fragment of a directed MSDU.

9.6 Multirate support

Some PHYs have multiple data transfer rate capabilities that allow implementations to perform dynamic rate switching with the objective of improving performance. The algorithm for performing rate switching is beyond the scope of this standard, but in order to ensure coexistence and interoperability on multirate-capable PHYs, this standard defines a set of rules that shall be followed by all STAs.

All Control frames shall be transmitted at one of the rates in the `BSSBasicRateSet` (see 10.3.10.1), or at one of the rates in the PHY mandatory rate set so they will be understood by all STAs.

All frames with multicast and broadcast RA shall be transmitted at one of the rates included in the `BSSBasicRateSet`, regardless of their type.

Data and/or management MPDUs with a unicast immediate address shall be sent on any supported data rate selected by the rate switching mechanism (whose output is an internal MAC variable called `MACCurrentRate`, defined in units of 500 kbit/s, which is used for calculating the `Duration/ID` field of each frame). A STA shall not transmit at a rate that is known not to be supported by the destination STA, as reported in the supported rates element in the management frames. For frames of type `Data+CF-ACK`, `Data+CF-Poll+CF-ACK`, and `CF-Poll+CF-ACK`, the rate chosen to transmit the frame must be supported by both the addressed recipient STA and the STA to which the ACK is intended.

Under no circumstances shall a STA initiate transmission of a data or management frame at a data rate higher than the greatest rate in the `OperationalRateSet`, a parameter of the `MLME-JOIN.request` primitive.

In order to allow the transmitting STA to calculate the contents of the `Duration/ID` field, the responding STA shall transmit its Control Response frame (either CTS or ACK) at the same rate as the immediately previous frame in the frame exchange sequence (as defined in 9.7), if this rate belongs to the PHY mandatory rates, or else at the highest possible rate belonging to the PHY rates in the `BSSBasicRateSet`.

9.7 Frame exchange sequences

The allowable frame exchange sequences are summarized in Table 21 and Table 22. A legend applicable to both tables follows Table 22.

Table 21—Frame sequences

Sequence	Frames in sequence	Usage
Data(bc/mc)	1	Broadcast or multicast MSDU
Mgmt(bc)	1	Broadcast MMPDU
{RTS – CTS –} [Frag – ACK –] Last – ACK	2	Directed MSDU or MMPDU
PS-Poll – ACK	2	Deferred PS-POLL response
PS-Poll – [Frag – ACK –] Last – ACK	3	Immediate PS-POLL response
DTIM(CF) – [≤CF-Sequence> –] {CF-End}	2 or more	Start of CFP
[<CF-Sequence> –] {CF-End}	2 or more	Continuation of CFP after missing ACK or medium occupancy boundary

Table 22—CF frame sequences

CF frame sequence	Frames in sequence	Usage
Beacon(CF)	1	Beacon during CFP
Data(bc/mc)	1	Broadcast or multicast MSDU
Mgmt(bc)	1 or 2	Broadcast MMPDU
Mgmt(dir) – ACK	2 or 3	Directed MMPDU
Data(dir)+CF-Poll {+CF-Ack} – Data(dir)+CF-Ack – {CF-Ack(no data)}	2	Poll and ACK sent with MPDUs
Data(dir)+CF-Poll {+CF-Ack} – CF-Ack(no data)	2	Poll of STA with empty queue, insufficient time for queued MPDU, or too little time remaining before a dwell or medium occupancy boundary to send a queued frame
CF-Poll(no data){+CF-Ack} – Data(dir) – {CF-Ack(no data)}	2	Separate poll, ACK sent with MPDU
CF-Poll(no data){+CF-Ack} – Data(dir) – ACK	3	Polled STA sends to STA in BSS
CF-Poll(no data){+CF-Ack} – Null(no data)	2	Separate poll, STA queue empty, or insufficient time for queued MPDU or too little time remaining before a dwell or medium occupancy boundary to send a queued frame
Data(dir){+CF-Ack} – ACK	2	ACK if not CF-Pollable or not polled

LEGEND (For Table 21 and Table 22)

- 1—Items enclosed in brackets “[...]” may occur zero or more times in the sequence.
- 2—Items enclosed in braces “{...}” may occur zero or one time in the sequence.
- 3—An isolated hyphen “-” represents a SIFS interval separating the pair of frames.
- 4—“Data(bc/mc)” represents any frame of type Data with a broadcast or multicast address in the Address1 field.
- 5—“Mgmt(bc)” represents any Management type frame with a broadcast address in the DA field.
- 6—“RTS” represents a Control frame of subtype RTS.
- 7—“CTS” represents a Control frame of subtype CTS.

LEGEND (Continued)

- 8—"ACK" represents a Control frame of subtype ACK.
- 9—"Frag" represents an MPDU of type Data or an MMPDU of type Management with an individual address in the Address1 field that has the More Fragments field set to "1."
- 10—"Last" represents an MPDU of type Data or an MMPDU of type Management with an individual address in the Address1 field that has the More Fragments field set to "0."
- 11—"PS-Poll" represents a Control frame of subtype PS-Poll.
- 12—"DTIM(CF)" represents a management frame of subtype Beacon that contains a DTIM information element with a nonzero value in the CFPDurRemaining field of its Parameter Set element.
- 13—"CF-End" represents a Control frame of type CF-End, or (if the final frame of the immediately preceding <CF-Sequence> was a directed data or management frame requiring acknowledgment by the AP) of type CF-End+Ack.
- 14—"Beacon(CF)" represents a management frame of subtype Beacon with a nonzero value in the CFPDurRemaining field of its CF Parameter Set element.
- 15—"Data(dir)" represents any MPDU of type Data with an individual address in the Address1 field.
- 16—"Mgmt(dir)" represents any MMPDU of type Management with an individual address in the Address1 field.
- 17—"CF-Ack(no data)" represents a data frame of subtype CF-ACK (no data).
- 18—"CF-Poll(no data)" represents a data frame of subtype CF-Poll (no data).
- 19—"Null(no data)" represents a data frame of subtype Null Function (no data).
- 20—" {+CF-Ack} " indicates that the frame may or may not include a contention-free acknowledgment.
- 21—" +CF-Ack " indicates that the frame includes a contention-free acknowledgment.
- 22—" +CF-Poll " indicates that the frame includes a contention-free poll.
- 23—<CF-Sequence> represents a sequence of one or more frames sent during a CFP. A valid <CF-Sequence> shall consist of one of the frame sequences shown in Table 22. The collection of sequences of frame exchanges corresponding to [<CF-Sequence>] may occur in any order within the CFP.

Individual frames within each of these sequences are separated by a SIFS.

9.8 MSDU transmission restrictions

To avoid reordering MSDUs between pairs of LLC entities and/or unnecessarily discarding MSDUs, the following restrictions shall be observed by any STA that is able to concurrently process multiple outstanding MSDUs for transmission. Note that here the term "outstanding" refers to an MSDU or MMPDU that is eligible to be transmitted at a particular time. A STA may have any number (greater than or equal to one) of eligible MSDUs outstanding concurrently, subject to the restrictions below.

The STA shall ensure that no more than one MSDU or MMPDU from a particular SA to a particular individual RA is outstanding at a time. Note that a simpler, more restrictive invariant to maintain is that no more than one MSDU with a particular individual RA may be outstanding at a time.

In a STA where the optional StrictlyOrdered service class has been implemented, that STA shall ensure that there is no group-addressed (multidestination) MSDU of the StrictlyOrdered service class outstanding from the SA of any other outstanding MSDU (either directed or group-addressed). This is because a group-addressed MSDU is implicitly addressed to a collection of peer STAs that could include any individual RA.

It is recommended that the STA select a value of aMaxMSDUTransmitLifetime that is sufficiently large that the STA does not discard MSDUs due to excessive Transmit MSDU timeouts under normal operating conditions.

10. Layer management

10.1 Overview of management model

Both MAC and PHY layers conceptually include management entities, called MAC sublayer management and PHY layer management entities (MLME and PLME, respectively). These entities provide the layer management service interfaces through which layer management functions may be invoked.

In order to provide correct MAC operation, a station management entity (SME) shall be present within each STA. The SME is a layer-independent entity that may be viewed as residing in a separate management plane or as residing "off to the side." The exact functions of the SME are not specified in this standard, but in general this entity may be viewed as being responsible for such functions as the gathering of layer-dependent status from the various layer management entities, and similarly setting the value of layer-specific parameters. SME would typically perform such functions on behalf of general system management entities and would implement standard management protocols. Figure 11 depicts the relationship among management entities.

The various entities within this model interact in various ways. Certain of these interactions are defined explicitly within this standard, via a service access point (SAP) across which defined primitives are exchanged. Other interactions are not defined explicitly within this standard, such as the interfaces between MAC and MLME and between PLCP and PLME, represented as double arrows within Figure 63. The specific manner in which these MAC and PHY management entities are integrated into the overall MAC and PHY layers is not specified within this standard.

The management SAPs within this model are the following:

- SME-MLME SAP
- SME-PLME SAP
- MLME-PLME SAP

The latter two SAPs support identical primitives, and in fact may be viewed as a single SAP (called the PLME SAP) that may be used either directly by MLME or by SME. In this fashion, the model reflects what is anticipated to be a common implementation approach in which PLME functions are controlled by the MLME (on behalf of SME). In particular, PHY implementations are not required to have separate interfaces defined other than their interfaces with the MAC and MLME.

10.2 Generic management primitives

The management information specific to each layer is represented as a management information base (MIB) for that layer. The MAC and PHY layer management entities are viewed as "containing" the MIB for that layer. The generic model of MIB-related management primitives exchanged across the management SAPs is to allow the SAP user-entity to either GET the value of a MIB attribute, or to SET the value of a MIB attribute. The invocation of a SET.request primitive may require that the layer entity perform certain defined actions.

Figure 63 depicts these generic primitives.

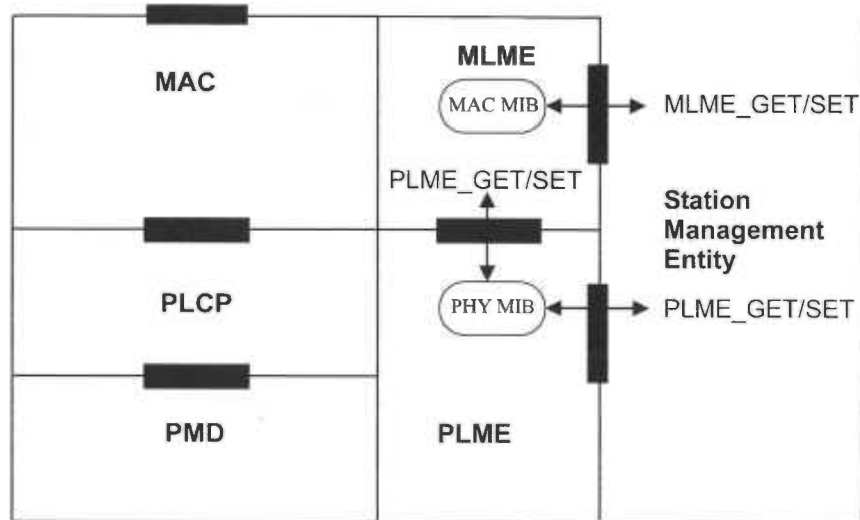


Figure 63—GET and SET operations

The GET and SET primitives are represented as REQUESTs with associated CONFIRM primitives. These primitives are prefixed by MLME or PLME depending upon whether the MAC or PHY layer management SAP is involved. In the following, XX denotes MLME or PLME:

XX-GET.request (MIBattribute)

Requests the value of the given MIBattribute.

XX-GET.confirm (status, MIBattribute, MIBattributevalue)

Returns the appropriate MIB attribute value if status = "success," otherwise returns an error indication in the Status field. Possible error status values include "invalid MIB attribute" and "attempt to get write-only MIB attribute."

XX-SET.request (MIBattribute, MIBattributevalue)

Requests that the indicated MIB attribute be set to the given value. If this MIBattribute implies a specific action, then this requests that the action be performed.

XX-SET.confirm (status, MIBattribute)

If status = "success," this confirms that the indicated MIB attribute was set to the requested value, otherwise it returns an error condition in status field. If this MIBattribute implies a specific action, then this confirms that the action was performed. Possible error status values include "invalid MIB attribute" and "attempt to set read-only MIB attribute."

Additionally, there are certain requests (with associated confirms) that may be invoked across a given SAP that do not involve the setting or getting of a specific MIB attribute. One of these is supported by each SAP, as follows:

- XX-RESET.request: where XX is MLME or PLME as appropriate
- XX-RESET.confirm

This service is used to initialize the management entities, the MIBs, and the datapath entities. It may include a list of attributes for items to be initialized to non-default values. The corresponding .confirm indicates success or failure of the request.

Other SAP-specific primitives are identified in 10.3.

10.3 MLME SAP interface

The services provided by the MLME to the SME are specified in this subclause. These services are described in an abstract way and do not imply any particular implementation or exposed interface. MLME SAP primitives are of the general form ACTION.request followed by ACTION.confirm. The SME uses the services provided by the MLME through the MLME SAP.

10.3.1 Power management

This mechanism supports the process of establishment and maintenance of the power management mode of a STA.

10.3.1.1 MLME-POWERMGT.request

10.3.1.1.1 Function

This primitive requests a change in the power management mode.

10.3.1.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-POWERMGT.request    (
                           PowerManagementMode,
                           WakeUp,
                           ReceiveDTIMs
                           )
```

Name	Type	Valid range	Description
PowerManagementMode	Enumeration	ACTIVE, POWER_SAVE	An enumerated type that describes the desired power management mode of the STA.
WakeUp	Boolean	True, false	When true, the MAC is forced immediately into the Awake state. This parameter has no effect if the current power management mode is ACTIVE.
ReceiveDTIMs	Boolean	True, false	When true, this parameter causes the STA to awaken to receive all DTIM frames. When false, the STA is not required to awaken for every DTIM frame.

10.3.1.1.3 When generated

This primitive is generated by the SME to implement the power-saving strategy of an implementation.

10.3.1.1.4 Effect of receipt

This request sets the STA's power management parameters. The MLME subsequently issues a MLME-POWERMGT.confirm that reflects the results of the power management change request.

10.3.1.2 MLME-POWERMGT.confirm

10.3.1.2.1 Function

This primitive confirms the change in power management mode.

10.3.1.2.2 Semantics of the service primitive

The primitive parameters are as follows:

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

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

10.3.1.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-POWERMGT.request to establish a new power management mode. It is not generated until the change has completed.

10.3.1.2.4 Effect of receipt

The SME is notified of the change of power management mode.

10.3.2 Scan

This mechanism supports the process of determining the characteristics of the available BSSs.

10.3.2.1 MLME-SCAN.request

10.3.2.1.1 Function

This primitive requests a survey of potential BSSs that the STA may later elect to try to join.

10.3.2.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SCAN.request        (
                           BSSType,
                           BSSID,
                           SSID,
                           ScanType,
                           ProbeDelay,
                           ChannelList,
                           MinChannelTime,
                           MaxChannelTime
                           )
```

Name	Type	Valid range	Description
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT, ANY_BSS	Determines whether Infrastructure BSS, Independent BSS, or both, are included in the scan
BSSID	MACAddress	Any valid individual or broadcast MAC address	Identifies a specific or broadcast BSSID
SSID	Octet string	0–32 octets	Specifies the desired SSID or the broadcast SSID
ScanType	Enumeration	ACTIVE, PASSIVE	Indicates either active or passive scanning
ProbeDelay	Integer	N/A	Delay (in μ s) to be used prior to transmitting a Probe frame during active scanning
ChannelList	Ordered set of integers	Each channel will be selected from the valid channel range for the appropriate PHY and carrier set.	Specifies a list of channels that are examined when scanning for a BSS
MinChannelTime	Integer	\geq ProbeDelay	The minimum time (in TU) to spend on each channel when scanning
MaxChannelTime	Integer	\geq MinChannelTime	The maximum time (in TU) to spend on each channel when scanning

10.3.2.1.3 When generated

This primitive is generated by the SME for a STA to determine if there are other BSSs that it may join.

10.3.2.1.4 Effect of receipt

This request initiates the scan process when the current frame exchange sequence is completed.

10.3.2.2 MLME-SCAN.confirm

10.3.2.2.1 Function

This primitive returns the descriptions of the set of BSSs detected by the scan process.

10.3.2.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```

MLME-SCAN.confirm      (
                        BSSDescriptionSet,
                        ResultCode
                        )
    
```

Name	Type	Valid range	Description
BSSDescriptionSet	Set of BSSDescriptions	N/A	The BSSDescriptionSet is returned to indicate the results of the scan request. It is a set containing zero or more instances of a BSSDescription.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS	Indicates the result of the MLME-SCAN.confirm

Each BSSDescription consists of the following elements:

Name	Type	Valid range	Description
BSSID	MACAddress	N/A	The BSSID of the found BSS
SSID	Octet string	1–32 octets	The SSID of the found BSS
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT	The type of the found BSS
Beacon Period	Integer	N/A	The Beacon period of the found BSS (in TU)
DTIM Period	Integer	As defined in frame format	The DTIM period of the BSS (in beacon periods)
Timestamp	Integer	N/A	The timestamp of the received frame (probe response/beacon) from the found BSS
Local Time	Integer	N/A	The value of the STA's TSF timer at the start of reception of the first octet of the timestamp field of the received frame (probe response or beacon) from the found BSS
PHY parameter set	As defined in frame format	As defined in frame format	The parameter set relevant to the PHY
CF parameter set	As defined in frame format	As defined in frame format	The parameter set for the CF periods, if found BSS supports CF mode
IBSS parameter set	As defined in frame format	As defined in frame format	The parameter set for the IBSS, if found BSS is an IBSS
CapabilityInformation	As defined in frame format	As defined in frame format	The advertised capabilities of the BSS
BSSBasicRateSet	Set of integers	2–127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kb/s) that must be supported by all STAs that desire to join this BSS. The STAs must be able to receive at each of the data rates listed in the set.

10.3.2.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-SCAN.request to ascertain the operating environment of the STA.

10.3.2.2.4 Effect of receipt

The SME is notified of the results of the scan procedure.

10.3.3 Synchronization

This mechanism supports the process of selection of a peer in the authentication process.

10.3.3.1 MLME-JOIN.request

10.3.3.1.1 Function

This primitive requests synchronization with a BSS.

10.3.3.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-JOIN.request      (
                        BSSDescription,
                        JoinFailureTimeout,
                        ProbeDelay,
                        OperationalRateSet
                        )
```

Name	Type	Valid range	Description
BSSDescription	BSSDescription	N/A	The BSSDescription of the BSS to join. The BSSDescription is a member of the set of descriptions that was returned as a result of a MLME-SCAN.request.
JoinFailureTimeout	Integer	≥1	The time limit, in units of beacon intervals, after which the join procedure will be terminated
ProbeDelay	Integer	N/A	Delay (in μs) to be used prior to transmitting a Probe frame during active scanning
OperationalRateSet	Set of integers	2–127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kbit/s) that the STA may use for communication within the BSS. The STA must be able to receive at each of the data rates listed in the set. The OperationalRateSet is a superset of the BSSBasicRateSet advertised by the BSS.

10.3.3.1.3 When generated

This primitive is generated by the SME for a STA to establish synchronization with a BSS.

10.3.3.1.4 Effect of receipt

This primitive initiates a synchronization procedure once the current frame exchange sequence is complete. The MLME synchronizes its timing with the specified BSS based on the elements provided in the BSSDescription parameter. The MLME subsequently issues a MLME-JOIN.confirm that reflects the results.

10.3.3.2 MLME-JOIN.confirm

10.3.3.2.1 Function

This primitive confirms synchronization with a BSS.

10.3.3.2.2 Semantics of the service primitive

The primitive parameters are as follows:

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

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

10.3.3.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-JOIN.request to establish synchronization with a BSS.

10.3.3.2.4 Effect of receipt

The SME is notified of the results of the synchronization procedure.

10.3.4 Authenticate

This mechanism supports the process of establishing an authentication relationship with a peer MAC entity.

10.3.4.1 MLME-AUTHENTICATE.request

10.3.4.1.1 Function

This primitive requests authentication with a specified peer MAC entity.

10.3.4.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-AUTHENTICATE.request (
    PeerSTAAddress,
    AuthenticationType,
    AuthenticateFailureTimeout
)
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the authentication process
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY	Specifies the type of authentication algorithm to use during the authentication process
AuthenticationFailure-Timeout	Integer	≥1	Specifies a time limit (in TU) after which the authentication procedure will be terminated

10.3.4.1.3 When generated

This primitive is generated by the SME for a STA to establish authentication with a specified peer MAC entity in order to permit Class 2 frames to be exchanged between the two STAs. During the authentication procedure, the SME may generate additional MLME-AUTHENTICATE.request primitives.

10.3.4.1.4 Effect of receipt

This primitive initiates an authentication procedure. The MLME subsequently issues a MLME-AUTHENTICATE.confirm that reflects the results.

10.3.4.2 MLME-AUTHENTICATE.confirm

10.3.4.2.1 Function

This primitive reports the results of an authentication attempt with a specified peer MAC entity.

10.3.4.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```

MLME-AUTHENTICATE.confirm (
    PeerSTAAddress,
    AuthenticationType,
    ResultCode
)
    
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication process was attempted. This value must match the peerSTAAddress parameter specified in the corresponding MLME-AUTHENTICATE.request.
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY	Specifies the type of authentication algorithm that was used during the authentication process. This value must match the authenticationType parameter specified in the corresponding MLME-AUTHENTICATE.request.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TIMEOUT, TOO_MANY_SIMULTANEOUS_REQUESTS, REFUSED	Indicates the result of the MLME-AUTHENTICATE.request.

10.3.4.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-AUTHENTICATE.request to authenticate with a specified peer MAC entity.

10.3.4.2.4 Effect of receipt

The SME is notified of the results of the authentication procedure.

10.3.4.3 MLME-AUTHENTICATE.indication

10.3.4.3.1 Function

This primitive reports the establishment of an authentication relationship with a specific peer MAC entity.

10.3.4.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```

MLME-AUTHENTICATE.indication (
    PeerSTAAddress,
    AuthenticationType
)
    
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication relationship was established
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY	Specifies the type of authentication algorithm that was used during the authentication process

10.3.4.3.3 When generated

This primitive is generated by the MLME as a result of the establishment of an authentication relationship with a specific peer MAC entity that resulted from an authentication procedure that was initiated by that specific peer MAC entity.

10.3.4.3.4 Effect of receipt

The SME is notified of the establishment of the authentication relationship.

10.3.5 De-authenticate

This mechanism supports the process of invalidating an authentication relationship with a peer MAC entity.

10.3.5.1 MLME-DEAUTHENTICATE.request

10.3.5.1.1 Function

This primitive requests that the authentication relationship with a specified peer MAC entity be invalidated.

10.3.5.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.request (
    PeerSTAAddress,
    ReasonCode
)
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the deauthentication process
ReasonCode	As defined in frame format	As defined in frame format	Specifies the reason for initiating the deauthentication procedure

10.3.5.1.3 When generated

This primitive is generated by the SME for a STA to invalidate authentication with a specified peer MAC entity in order to prevent the exchange of Class 2 frames between the two STAs. During the deauthentication procedure, the SME may generate additional MLME-DEAUTHENTICATE.request primitives.

10.3.5.1.4 Effect of receipt

This primitive initiates a deauthentication procedure. The MLME subsequently issues a MLME-DEAUTHENTICATE.confirm that reflects the results.

10.3.5.2 MLME-DEAUTHENTICATE.confirm

10.3.5.2.1 Function

This primitive reports the results of a deauthentication attempt with a specified peer MAC entity.

10.3.5.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.confirm (
    PeerSTAAddress,
    ResultCode
)
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the deauthentication process was attempted
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TOO_MANY_SIMUL- TANEOUS_REQUESTS	Indicates the result of the MLME-DEAUTHENTICATE.request

10.3.5.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-DEAUTHENTICATE.request to invalidate the authentication relationship with a specified peer MAC entity.

10.3.5.2.4 Effect of receipt

The SME is notified of the results of the deauthentication procedure.

10.3.5.3 MLME-DEAUTHENTICATE.indication

10.3.5.3.1 Function

This primitive reports the invalidation of an authentication relationship with a specific peer MAC entity.

10.3.5.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.indication (
    PeerSTAAddress,
    ReasonCode
)
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication relationship was invalidated
ReasonCode	As defined in frame format.	As defined in frame format	Specifies the reason the deauthentication procedure was initiated

10.3.5.3.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an authentication relationship with a specific peer MAC entity.

10.3.5.3.4 Effect of receipt

The SME is notified of the invalidation of the specific authentication relationship.

10.3.6 Associate

The following primitives describe how a STA becomes associated with an access point (AP).

10.3.6.1 MLME-ASSOCIATE.request

10.3.6.1.1 Function

This primitive requests association with a specified peer MAC entity that is acting as an AP.

10.3.6.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```

MLME-ASSOCIATE.request    (
                            PeerSTAAddress,
                            AssociateFailureTimeout,
                            CapabilityInformation,
                            ListenInterval
                            )
    
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the association process
AssociateFailureTimeout	Integer	≥1	Specifies a time limit (in TU) after which the associate procedure will be terminated
CapabilityInformation	As defined in frame format	As defined in frame format	Specifies the operational capability definitions to be used by the MAC entity
ListenInterval	Integer	≥0	Specifies the number of beacon intervals that may pass before the STA awakens and listens for the next beacon

10.3.6.1.3 When generated

This primitive is generated by the SME when a STA wishes to establish association with an AP.

10.3.6.1.4 Effect of receipt

This primitive initiates an association procedure. The MLME subsequently issues an MLME-ASSOCIATE.confirm that reflects the results.

10.3.6.2 MLME-ASSOCIATE.confirm

10.3.6.2.1 Function

This primitive reports the results of an association attempt with a specified peer MAC entity that is acting as an AP.

10.3.6.2.2 Semantics of the service primitive

The primitive parameters are as follows:

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

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TIMEOUT, REFUSED	Indicates the result of the MLME-ASSOCIATE.request

10.3.6.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-ASSOCIATE.request to associate with a specified peer MAC entity that is acting as an AP.

10.3.6.2.4 Effect of receipt

The SME is notified of the results of the association procedure.

10.3.6.3 MLME-ASSOCIATE.indication

10.3.6.3.1 Function

This primitive reports the establishment of an association with a specific peer MAC entity.

10.3.6.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ASSOCIATE.indication (
                           PeerSTAAddress
                           )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the association was established

10.3.6.3.3 When generated

This primitive is generated by the MLME as a result of the establishment of an association with a specific peer MAC entity that resulted from an association procedure that was initiated by that specific peer MAC entity.

10.3.6.3.4 Effect of receipt

The SME is notified of the establishment of the association.

10.3.7 Reassociate

The following primitives describe how a STA becomes associated with another AP.

10.3.7.1 MLME-REASSOCIATE.request

10.3.7.1.1 Function

This primitive requests a change in association to a specified new peer MAC entity that is acting as an AP.

10.3.7.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```

MLME-REASSOCIATE.request    (
                               NewAPAddress,
                               ReassociateFailureTimeout,
                               CapabilityInformation,
                               ListenInterval
                               )
    
```

Name	Type	Valid range	Description
NewAPAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the reassociation process
ReassociateFailureTimeout	Integer	≥1	Specifies a time limit (in TU) after which the reassociate procedure will be terminated
CapabilityInformation	As defined in frame format	As defined in frame format	Specifies the operational capability definitions to be used by the MAC entity
ListenInterval	Integer	≥0	Specifies the number of beacon intervals that may pass before the STA awakens and listens for the next beacon.

10.3.7.1.3 When generated

This primitive is generated by the SME for a STA to change association to a specified new peer MAC entity that is acting as an AP.

10.3.7.1.4 Effect of receipt

This primitive initiates a reassociation procedure. The MLME subsequently issues a MLME-REASSOCIATE.confirm that reflects the results.

10.3.7.2 MLME-REASSOCIATE.confirm

10.3.7.2.1 Function

This primitive reports the results of a reassociation attempt with a specified peer MAC entity that is acting as an AP.

10.3.7.2.2 Semantics of the service primitive

The primitive parameters are as follows:

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

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, TIMEOUT, REFUSED	Indicates the result of the MLME-REASSOCI- ATE.request

10.3.7.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-REASSOCIATE.request to reassociate with a specified peer MAC entity that is acting as an AP.

10.3.7.2.4 Effect of receipt

The SME is notified of the results of the reassociation procedure.

10.3.7.3 MLME-REASSOCIATE.indication

10.3.7.3.1 Function

This primitive reports the establishment of a reassociation with a specified peer MAC entity.

10.3.7.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REASSOCIATE.indication (
    PeerSTAAddress
)
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the reassociation was established

10.3.7.3.3 When generated

This primitive is generated by the MLME as a result of the establishment of a reassociation with a specific peer MAC entity that resulted from a reassociation procedure that was initiated by that specific peer MAC entity.

10.3.7.3.4 Effect of receipt

The SME is notified of the establishment of the reassociation.

10.3.8 Disassociate

10.3.8.1 MLME-DISASSOCIATE.request

10.3.8.1.1 Function

This primitive requests disassociation with a specified peer MAC entity that is acting as an AP.

10.3.8.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.request    (
                               PeerSTAAddress,
                               ReasonCode
                               )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the disassociation process
ReasonCode	As defined in frame format	As defined in frame format	Specifies the reason for initiating the disassociation procedure

10.3.8.1.3 When generated

This primitive is generated by the SME for a STA to establish disassociation with an AP.

10.3.8.1.4 Effect of receipt

This primitive initiates a disassociation procedure. The MLME subsequently issues an MLME-DISASSOCIATE.confirm that reflects the results.

10.3.8.2 MLME-DISASSOCIATE.confirm

10.3.8.2.1 Function

This primitive reports the results of a disassociation procedure with a specific peer MAC entity that is acting as an AP.

10.3.8.2.2 Semantics of the service primitive

The primitive parameters are as follows:

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

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, TIMEOUT, REFUSED	Indicates the result of the MLME-DISASSOCIATE.request

10.3.8.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-DISASSOCIATE.request to disassociate with a specified peer MAC entity that is acting as an AP.

10.3.8.2.4 Effect of receipt

The SME is notified of the results of the disassociation procedure.

10.3.8.3 MLME-DISASSOCIATE.indication

10.3.8.3.1 Function

This primitive reports disassociation with a specific peer MAC entity.

10.3.8.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.indication (
    PeerSTAAddress,
    ReasonCode
)
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the association relationship was invalidated
ReasonCode	As defined in frame format	As defined in frame format	Specifies the reason the disassociation procedure was initiated

10.3.8.3.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an association relationship with a specific peer MAC entity.

10.3.8.3.4 Effect of receipt

The SME is notified of the invalidation of the specific association relationship.

10.3.9 Reset

This mechanism supports the process of resetting the MAC.

10.3.9.1 MLME-RESET.request

10.3.9.1.1 Function

This primitive requests that the MAC entity be reset.

10.3.9.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESET.request      (
                          STAAddress,
                          SetDefaultMIB
                          )
```

Name	Type	Valid range	Description
STAAddress	MACAddress	Any valid MAC address	Specifies the MAC address that is to be used by the MAC entity that is being reset. This value may be used to provide a locally administered STA address.
SetDefaultMIB	Boolean	True, false	If true, all MIB attributes are set to their default values. The default values are implementation dependent. If false, the MAC is reset, but all MIB attributes retain the values that were in place prior to the generation of the MLME-RESET.request primitive.

10.3.9.1.3 When generated

This primitive is generated by the SME to reset the MAC to initial conditions. The MLME-RESET.request primitive must be used prior to use of the MLME-START.request primitive.

10.3.9.1.4 Effect of receipt

This primitive sets the MAC to initial conditions, clearing all internal variables to the default values. MIB attributes may be reset to their implementation-dependent default values by setting the SetDefaultMIB flag to true. The MLME subsequently issues a MLME-RESET.confirm that reflects the results.

10.3.9.2 MLME-RESET.confirm

10.3.9.2.1 Function

This primitive reports the results of a reset procedure.

10.3.9.2.2 Semantics of the service primitive

The primitive parameters are as follows:

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

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS	Indicates the result of the MLME-RESET.request

10.3.9.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-RESET.request to reset the MAC entity.

10.3.9.2.4 Effect of receipt

The SME is notified of the results of the reset procedure.

10.3.10 Start

This mechanism supports the process of creating a new BSS.

10.3.10.1 MLME-START.request

10.3.10.1.1 Function

This primitive requests that the MAC entity start a new BSS.

10.3.10.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```

MLME-START.request      (
                          SSID,
                          BSSType,
                          BeaconPeriod,
                          DTIMPeriod,
                          CF parameter set,
                          PHY parameter set,
                          IBSS parameter set,
                          ProbeDelay,
                          CapabilityInformation,
                          BSSBasicRateSet,
                          OperationalRateSet
                          )
    
```

Name	Type	Valid range	Description
SSID	Octet string	1–32 octets	The SSID of the BSS
BSSType	Enumeration	INFRA-STRUCTURE, INDEPENDENT	The type of the BSS
Beacon Period	Integer	≥1	The Beacon period of the BSS (in TU)
DTIM Period	Integer	As defined in frame format	The DTIM Period of the BSS (in beacon periods)
CF parameter set	As defined in frame format	As defined in frame format	The parameter set for CF periods, if the BSS supports CF mode. aCFPPeriod is modified as a side effect of the issuance of an MLME-START.request primitive.
PHY parameter set	As defined in frame format	As defined in frame format	The parameter set relevant to the PHY
IBSS parameter set	As defined in frame format	As defined in frame format	The parameter set for the IBSS, if BSS is an IBSS

Name	Type	Valid range	Description
ProbeDelay	Integer	N/A	Delay (in μ s) to be used prior to transmitting a Probe frame during active scanning
CapabilityInformation	As defined in frame format	As defined in frame format	The capabilities to be advertised for the BSS
BSSBasicRateSet	Set of integers	2–127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kbit/s) that must be supported by all STAs to join this BSS. The STA that is creating the BSS must be able to receive and transmit at each of the data rates listed in the set.
OperationalRateSet	Set of integers	2–127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kbit/s) that the STA may use for communication within the BSS. The STA must be able to receive at each of the data rates listed in the set. The OperationalRateSet is a superset of the BSSBasicRateSet advertised by the BSS.

10.3.10.1.3 When generated

This primitive is generated by the SME to start either an infrastructure BSS (with the MAC entity acting as an AP), or to start an independent BSS (with the MAC entity acting as the first STA in the IBSS).

The MLME-START.request primitive must be generated after an MLME-RESET.request primitive has been used to reset the MAC entity and before an MLME-JOIN.request primitive has been used to successfully join an existing infrastructure BSS or independent BSS.

The MLME-START.request primitive must not be used after successful use of the MLME-START.request primitive or successful use of the MLME-JOIN.request without generating an intervening MLME-RESET.request primitive.

10.3.10.1.4 Effect of receipt

This primitive initiates the BSS initialization procedure once the current frame exchange sequence is complete. The MLME subsequently issues an MLME-START.confirm that reflects the results of the creation procedure.

10.3.10.2 MLME-START.confirm

10.3.10.2.1 Function

This primitive reports the results of a BSS creation procedure.

10.3.10.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```

MLME-START.confirm      (
                          ResultCode
                          )
    
```

Name	Type	Valid range	resetDescription
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, BSS_ALREADY_STARTED_ OR_JOINED	Indicates the result of the MLME-START.request

10.3.10.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-START.request to create a new BSS.

10.3.10.2.4 Effect of receipt

The SME is notified of the results of the BSS creation procedure.

10.4 PLME SAP interface

The PHY management service interface consists of the generic PLMEGET and PLMESET primitives on PHY MIB attributes, as described previously, together with the PLME-RESET and PLME-CHARACTERISTICS primitives and the following specific primitives.

10.4.1 PLME-RESET.request**10.4.1.1 Function**

This primitive shall be a request by the LME to reset the PHY. The PHY shall be always reset to the receive state to avoid accidental data transmission.

10.4.1.2 Semantics of the service primitive

The semantics of the primitive are as follows:

PLME-RESET.request ()

There are no parameters associated with this primitive.

10.4.1.3 When generated

This primitive shall be generated at any time to reset the PHY.

10.4.1.4 Effect of receipt

Receipt of this primitive by the PHY sublayer shall cause the PHY entity to reset both the transmit and the receive state machines and place the PHY into the receive state.

10.4.2 PLME-CHARACTERISTICS.request**10.4.2.1 Function**

This primitive is a request by the LME to provide the PHY operational characteristics.

10.4.2.2 Semantics of the service primitive

The semantics of the primitive are as follows:

PLME-CHARACTERISTICS.request ()

There are no parameters associated with this primitive.

10.4.2.3 When generated

This primitive is generated by the LME, at initialization time, to request the PHY entity to provide its operational characteristics.

10.4.2.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity will be to generate a PLME-CHARACTERISTICS.confirm primitive that conveys its operational characteristics.

10.4.3 PLME-CHARACTERISTICS.confirm**10.4.3.1 Function**

This primitive provides the PHY operational parameters.

10.4.3.2 Semantics of the service primitive

The primitive provides the following parameters:

```

PLME-CHARACTERISTICS.confirm(
    aSlotTime,
    aSIFSTime,
    aCCATime,
    aRxTxTurnaroundTime,
    aTxPLCPDelay,
    aRxPLCPDelay,
    aRxTxSwitchTime,
    aTxRampOnTime,
    aTxRampOffTime,
    aTxRFDelay,
    aRxRFDelay,
    aAirPropagationTime,
    aMACProcessingDelay,
    aPreambleLength,
    aPLCPHeaderLength,
    aMPDUDurationFactor,
    aMPDUMaxLength,
    aCWmin,
    aCWmax
)

```


Name	Type	Description
aSlotTime	integer	The Slot Time (in μs) that the MAC will use for defining the PIFS and DIFS periods. See 9.2.10.
aSIFSTime	integer	The nominal time (in μs) that the MAC and PHY will require to receive the last symbol of a frame at the air interface, process the frame, and respond with the first symbol on the air interface of the earliest possible response frame. See 9.2.10.
aCCATime	integer	The minimum time (in μs) the CCA mechanism has available to assess the medium within every time slot to determine whether the medium is busy or idle.
aRxTxTurn-aroundTime	integer	The maximum time (in μs) that the PHY requires to change from receiving to transmitting the start of the first symbol. The following equation is used to derive the RxTxTurn-aroundTime: $\text{aTxPLCPDelay} + \text{aRxTxSwitchTime} + \text{aTxRampOnTime} + \text{aTxRFDelay}.$
aTxPLCPDelay	integer	The nominal time (in μs) that the PLCP uses to deliver a symbol from the MAC interface to the transmit data path of the PMD.
aRxPLCPDelay	integer	The nominal time (in μs) that the PLCP uses to deliver a bit from the PMD receive path to the MAC.
aRxTxSwitch-Time	integer	The nominal time (in μs) that the PMD takes to switch from Receive to Transmit.
aTxRampOnTime	integer	The maximum time (in μs) that the PMD takes to turn the Transmitter on.
aTxRampOffTime	integer	The nominal time (in μs) that the PMD takes to turn the Transmit Power Amplifier off.
aTxRFDelay	integer	The nominal time (in μs) between the issuance of a PMD-DATA.request to the PMD and the start of the corresponding symbol at the air interface. The start of a symbol is defined to be 1/2 symbol period prior to the center of the symbol for FH, or 1/2 chip period prior to the center of the first chip of the symbol for DS, or 1/2 slot time prior to the center of the corresponding slot for IR.
aRxRFDelay	integer	The nominal time (in μs) between the end of a symbol at the air interface to the issuance of a PMD-DATA.indicate to the PLCP. The end of a symbol is defined to be 1/2 symbol period after the center of the symbol for FH, or 1/2 chip period after the center of the last chip of the symbol for DS, or 1/2 slot time after the center of the corresponding slot for IR.
aAirPropagation-Time	integer	The anticipated time (in μs) it takes a transmitted signal to go from the transmitting station to the receiving station.
aMACProcessingDelay	integer	The nominal time (in μs) that the MAC uses to process a frame and prepare a response to the frame.
aPreambleLength	integer	The current PHY's Preamble Length (in μs). If the actual value of the length of the modulated preamble is not an integral number of microseconds, the value shall be rounded up to the next higher value.
aPLCPHeaderLength	integer	The current PHY's PLCP Header Length (in μs). If the actual value of the length of the modulated header is not an integral number of microseconds, the value shall be rounded up to the next higher value.
aMPDUDuration-Factor	integer	The overhead added by the PHY to the MPDU as it is transmitted through the wireless medium expressed as a scaling factor applied to the number of bits in the MPDU. The value of aMPDUDurationFactor is generated by the following equation: $\text{Truncate}[\frac{(\text{PPDUbits}/\text{PSDUbits}) - 1}{10^9}]$ The total time to transmit a PPDU over the air is generated by the following equation rounded up to the next integer μs : $\text{aPreambleLength} + \text{aPLCPHeaderLength} + \frac{((\text{aMPDUDurationFactor} \times 8 \times \text{PSDUoctets}) / 10^9) + (8 \times \text{PSDUoctets})}{\text{data rate}}$ where data rate is in Mbit/s. The total time (in μs) to the beginning of any octet in a PPDU from the first symbol of the preamble can be calculated using the duration factor in the following equation: $\text{Truncate}[\frac{\text{aPreambleLength} + \text{aPLCPHeaderLength} + ((\text{aMPDUDurationFactor} \times 8 \times N) / 10^9) + (8 \times N)}{\text{data rate}}] + 1,$ where data rate is in Mbit/s and where N counts the number of octets in the PPDU prior to the desired octet, but does not count the number of octets in the preamble PLCP Header.
aMPDUMax-Length	integer	The maximum number of octets in an MPDU that can be conveyed by a PLCPDU.
aCWmin	integer	The minimum size of the contention window, in units of aSlotTime.
aCWmax	integer	The maximum size of the contention window, in units of aSlotTime.

10.4.3.3 When generated

This primitive will be issued by the PHY entity in response to a PLME-CHARACTERISTICS.request.

10.4.3.4 Effect of receipt

The receipt of this primitive provides the operational characteristics of the PHY entity.

10.4.4 PLME-DSSSTESTMODE.request

10.4.4.1 Function

This primitive requests that the DSSS PHY entity enter a test mode operation. The parameters associated with this primitive are considered as recommendations and are optional in any particular implementation.

10.4.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```

PLME-DSSSTESTMODE.request (
    TEST_ENABLE,
    TEST_MODE,
    SCRAMBLE_STATE,
    SPREADING_STATE,
    DATA_TYPE,
    DATA_RATE;
)
    
```

Name	Type	Valid range	Description
TEST_ENABLE	Boolean	True, false	If true, enables the PHY test mode according to the remaining parameters
TEST_MODE	integer	1, 2, 3	TEST_MODE selects one of three operational states: 01 = transparent receive 02 = continuous transmit 03 = 50% duty cycle
SCRAMBLE_STATE	Boolean	True, false	If true, sets the operational state of the scrambler to ON
SPREADING_STATE	Boolean	True, false	If true, selects the operational state of the chipping
DATA_TYPE	integer	1, 2, 3	Selects one of three data patterns to be used for the transmit portions of the tests
DATA_RATE	integer	2, 4	Selects between 1 and 2 Mbit/s operation 02 = 1 Mbit/s 04 = 2 Mbit/s

10.4.4.3 When generated

This primitive shall be generated at any time to enter the DSSS PHY test mode.

10.4.4.4 Effect of receipt

Receipt of this primitive by the PHY sublayer shall cause the DSSS PHY entity to enter the test mode of operation.

10.4.5 PLME-DSSSTESTOUTPUT.request

10.4.5.1 Function

This optional primitive shall be a request by the LME to enable selected test signals from the PHY. The parameters associated with this primitive are considered as recommendations and are optional in any particular implementation.

10.4.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```

PLME-DSSSTESTOUTPUT.request    (
                                TEST_OUTPUT,
                                )
    
```

Name	Type	Valid range	Description
TEST_OUTPUT	Boolean	True, false	If true, enables the selected test signals for testing DS PHY

TEST_OUTPUT enables and disables selected signals for debugging and testing the PHY. Some signals that may be available for output are PHY-TXSTART.request, PHY-RXSTART.indicate(RXVECTOR), PHY-CCA.indicate, the chipping clock, the data clock, the symbol clock, TX data, and RX data.

10.4.5.3 When generated

This primitive shall be generated at any time to enable the test outputs when in the DSSS PHY test mode.

10.4.5.4 Effect of receipt

Receipt of this primitive by the DSSS PHY sublayer shall cause the DSSS PHY entity to enable the test outputs using the modes set by the most recent PLME-DSSSTESTMODE.request primitive.

11. MAC sublayer management entity

11.1 Synchronization

All STAs within a single BSS shall be synchronized to a common clock using the mechanisms defined herein.

11.1.1 Basic approach

A timing synchronization function (TSF) keeps the timers for all STAs in the same BSS synchronized. All STAs shall maintain a local TSF timer.

11.1.1.1 TSF for infrastructure networks

In an infrastructure network, the AP shall be the timing master and shall perform the TSF. The AP shall initialize its TSF timer independently of any simultaneously started APs in an effort to minimize the synchronization of the TSF timers of multiple APs. The AP shall periodically transmit special frames called *beacons* that contain a copy of its TSF timer to synchronize the other STAs in a BSS. A receiving STA shall always accept the timing information in beacons sent from the AP servicing its BSS. If a STA's TSF timer is different from the timestamp in the received beacon, the receiving STA shall set its local timer to the received timestamp value.

Beacons shall be generated for transmission by the AP once every BeaconPeriod time units.

11.1.1.2 TSF for an independent BSS (IBSS)

The TSF in an IBSS shall be implemented via a distributed algorithm that shall be performed by all of the members of the BSS. Each STA in the BSS shall transmit beacons according to the algorithm described in this clause. Each STA in an IBSS shall adopt the timing received from any beacon or probe response that has a TSF value later than its own TSF timer.

11.1.2 Maintaining synchronization

Each STA shall maintain a TSF timer with modulus 2^{64} counting in increments of microseconds. STAs expect to receive beacons at a nominal rate. The interval between beacons is defined by the aBeaconPeriod parameter of the STA. A STA sending a beacon shall set the value of the beacon's timestamp so that it equals the value of the STA's TSF timer at the time that the first bit of the timestamp is transmitted to the PHY plus the transmitting STA's delays through its local PHY from the MAC-PHY interface to its interface with the wireless medium (antenna, LED emission surface, etc.). The algorithms in this clause define a mechanism that maintains the synchronization of the TSF timers in a BSS to within 4 μ s plus the maximum propagation delay of the PHY for PHYs of 1 Mbit/s, or greater.

11.1.2.1 Beacon generation in infrastructure networks

The AP shall define the timing for the entire BSS by transmitting beacons according to the aBeaconPeriod attribute within the AP. This defines a series of TBTTs exactly aBeaconPeriod time units apart. Time zero is defined to be a TBTT with the beacon being a DTIM and transmitted at the beginning of a CFP. At each TBTT, the AP shall schedule a beacon as the next frame for transmission. If the medium is determined by the carrier-sense mechanism (see 9.2.1) to be unavailable, the AP shall delay the actual transmission of a beacon according to the basic medium access rules specified in Clause 9. The beacon period is included in Beacon and Probe Response frames, and STAs shall adopt that beacon period when joining the BSS.

NOTE—Though the transmission of a beacon may be delayed because of CSMA deferrals, subsequent beacons shall be scheduled at the nominal beacon interval. This is shown in Figure 64.

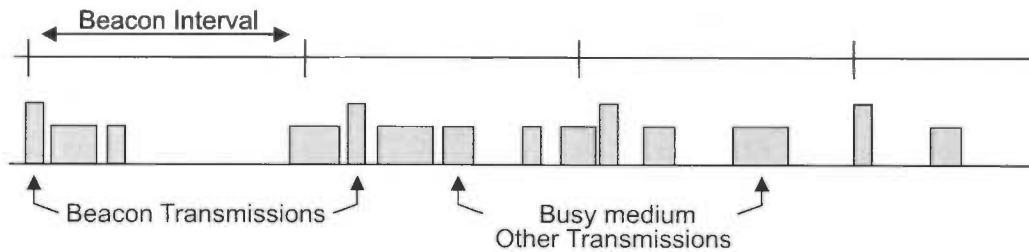


Figure 64—Beacon transmission on a busy network

11.1.2.2 Beacon generation in an IBSS

Beacon generation in an IBSS is distributed. The beacon period is included in Beacon and Probe Response frames, and STAs shall adopt that beacon period when joining the IBSS. All members of the IBSS participate in beacon generation. Each STA shall maintain its own TSF timer that is used for aBeaconPeriod timing. The beacon interval within an IBSS is established by the STA that instantiates the IBSS. This defines a series of TBTTs exactly aBeaconPeriod time units apart. Time zero is defined to be a TBTT. At each TBTT the STA shall

- a) Suspend the decrementing of the backoff timer for any pending non-beacon or non-ad hoc traffic indication (ATIM) transmission,
- b) Calculate a random delay uniformly distributed in the range between zero and twice $aCW_{min} \times aSlotTime$,
- c) Wait for the period of the random delay, decrementing the random delay timer using the same algorithm as for backoff,
- d) Cancel the remaining random delay and the pending beacon transmission, if a beacon arrives before the random delay timer has expired, and the ATIM backoff timer shall resume decrementing.
- e) Send a beacon if the random delay has expired and no beacon has arrived during the delay period.

(See Figure 65.)

The beacon transmission shall always occur during the Awake Period of STAs that are operating in a low-power mode. This is described in more detail in 11.2.

11.1.2.3 Beacon reception

STAs shall use information from the CF Parameter Set element of all received Beacon frames to update their NAV as specified in 9.3.2.2.

STAs in an infrastructure network shall only use other information in received Beacon frames, if the BSSID field is equal to the MAC address currently in use by the STA contained in the AP of the BSS.

STAs in an IBSS shall use other information in any received Beacon frame for which the IBSS subfield of the Capability field is set to 1 and the content of the SSID element is equal to the SSID of the IBSS. Use of this information is specified in 11.1.4.

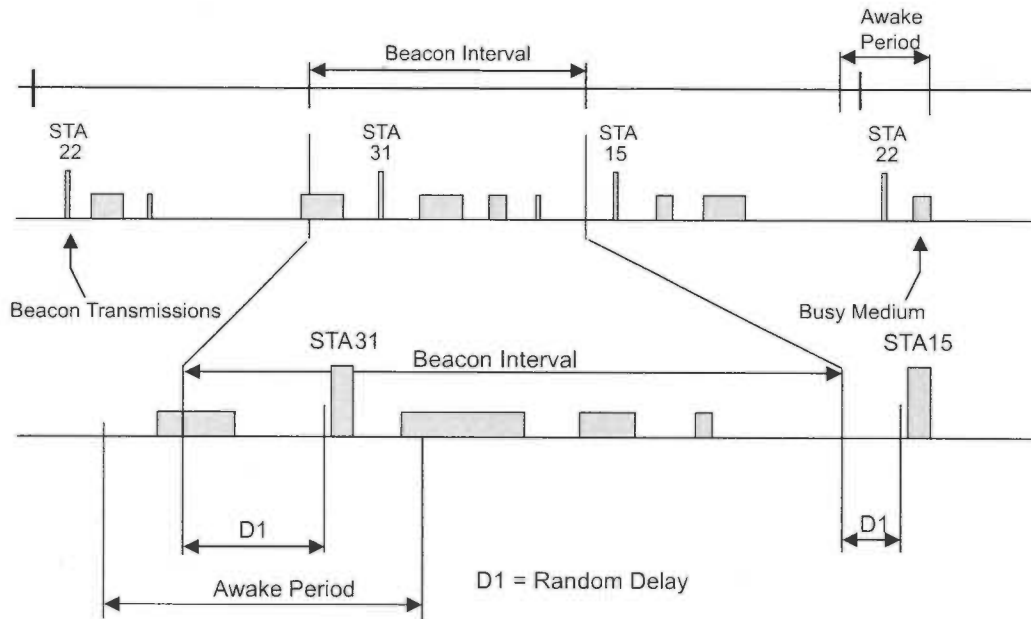


Figure 65—Beacon transmission in an IBSS

11.1.2.4 TSF timer accuracy

Upon receiving a Beacon frame with a valid FCS and BSSID or SSID, as described in 11.1.2.3, a STA shall update its TSF timer according to the following algorithm: The received timestamp value shall be adjusted by adding an amount equal to the receiving STA's delay through its local PHY components plus the time since the first bit of the timestamp was received at the MAC/PHY interface. In the case of an infrastructure BSS, the STA's TSF timer shall then be set to the adjusted value of the timestamp. In the case of an IBSS, the STA's TSF timer shall be set to the adjusted value of the received timestamp, if the adjusted value of the timestamp is later than the value of the STA's TSF timer. The accuracy of the TSF timer shall be $\pm 0.01\%$.

11.1.3 Acquiring synchronization, scanning

A STA shall operate in either a Passive Scanning mode or an Active Scanning mode depending on the current value of the ScanMode parameter of the MLME-SCAN.request primitive.

Upon receipt of the MLME-SCAN.request primitive, a STA shall perform scanning. The SSID parameter indicates the SSID for which to scan. To become a member of a particular ESS using passive scanning, a STA shall scan for Beacon frames containing that ESS's SSID, returning all Beacon frames matching the desired SSID in the BSSDescriptionSet parameter of the corresponding MLME-SCAN.confirm primitive with the appropriate bits in the Capabilities Information field indicating whether the beacon came from an Infrastructure BSS or IBSS. To actively scan, the STA shall transmit Probe frames containing the desired SSID. Upon completion of scanning, an MLME-SCAN.confirm is issued by the MLME indicating all of the BSS information received.

Upon receipt of an MLME-JOIN.request, the STA will join a BSS by adopting the BSSID, TSF timer value, PHY parameters, and the beacon period specified in the request.

Upon receipt of an MLME-SCAN.request with the broadcast SSID, the STA shall passively scan for any Beacon frames, or actively transmit Probe frames containing the broadcast SSID, as appropriate depending

upon the value of ScanMode. Upon completion of scanning, an MLME-SCAN.confirm is issued by the MLME indicating all of the BSS information received.

If a STA's scanning does not result in finding a BSS with the desired SSID and of the desired type, or does not result in finding any BSS, the STA may start an IBSS upon receipt of the MLME-START.request.

A STA may start its own BSS without first scanning for a BSS to join.

When a STA starts a BSS, that STA shall determine the BSSID of the BSS. If the BSSType indicates an infrastructure BSS, then the STA shall start an infrastructure BSS and the BSSID shall be equal to the STA's dot11StationID. The value of the BSSID shall remain unchanged, even if the value of dot11StationID is changed after the completion of the MLME-Start.request. If the BSSType indicates an IBSS, the STA shall start an IBSS, and the BSSID shall be an individual locally administered IEEE MAC address as defined in 5.2 of IEEE Std 802-1990. The remaining 46 bits of that MAC address shall be a number selected in a manner that minimizes the probability of STAs generating the same number, even when those STAs are subjected to the same initial conditions. The value SSID parameter shall be used as the SSID of the new BSS. It is important that designers recognize the need for statistical independence among the random number streams among STAs.

11.1.3.1 Passive scanning

If a ScanType is passive, the STA shall listen to each channel scanned for no longer than a maximum duration defined by the ChannelTime parameter.

11.1.3.2 Active scanning

Active scanning involves the generation of Probe frames and the subsequent processing of received Probe Response frames. The details of the active scanning procedures are as specified in the following subclauses.

11.1.3.2.1 Sending a probe response

STAs, subject to criteria below, receiving Probe Request frames shall respond with a probe response only if the SSID in the probe request is the broadcast SSID or matches the specific SSID of the STA. Probe Response frames shall be sent as directed frames to the address of the STA that generated the probe request. The probe response shall be sent using normal frame transmission rules. An AP shall respond to all probe requests meeting the above criteria. In an IBSS, the STA that generated the last beacon shall be the STA that responds to a probe request.

In each BSS there shall be at least one STA that is awake at any given time to respond to probe requests. A STA that sent a beacon shall remain in the Awake state and shall respond to probe requests until a Beacon frame with the current BSS ID is received. If the STA is an AP, it shall always remain in the Awake state and always respond to probe requests. There may be more than one STA in an IBSS that responds to any given probe request, particularly in cases where more than one STA transmitted a Beacon frame following the most recent TBTT, either due to not receiving successfully a previous beacon or due to collisions between beacon transmissions.

11.1.3.2.2 Active scanning procedure

Upon receipt of the MLME-SCAN.request with ScanType indicating an active scan, a STA shall use the following procedure:

For each channel to be scanned,

- a) Wait until the ProbeDelay time has expired or a PHYRxStart.indication has been received;
- b) Perform the Basic Access procedure as defined in 9.2.5.1;

- c) Send a probe with the broadcast destination, SSID, and broadcast BSSID;
- d) Clear and start a ProbeTimer;
- e) If PHYCCA.indication (busy) has not been detected before the ProbeTimer reaches MinChannelTime, then clear NAV and scan the next channel, else when ProbeTimer reaches MaxChannelTime, process all received probe responses;
- f) Clear NAV and scan the next channel.

See Figure 66.

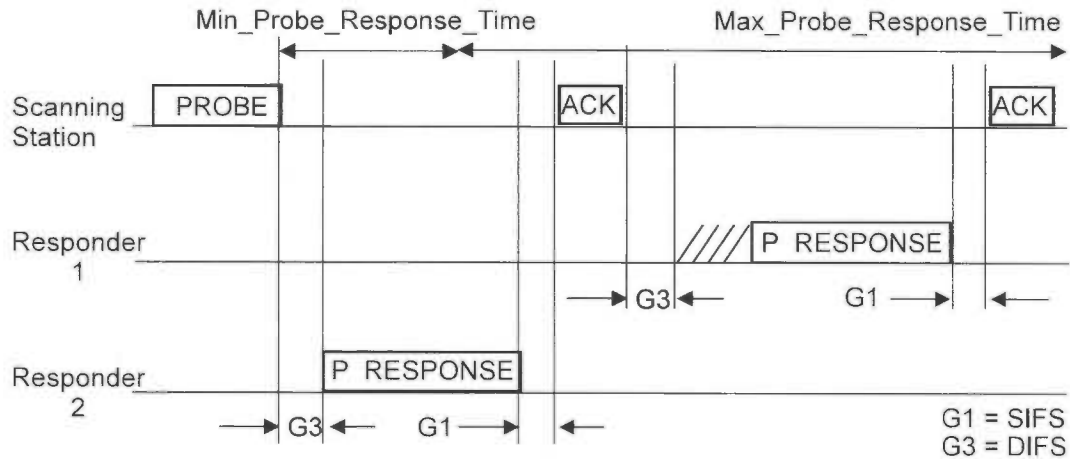


Figure 66—Probe response

When all channels in the ChannelList have been scanned, the MLME shall issue an MLME-Scan.confirm with the BSSDescriptionSet containing all of the information gathered during the scan.

11.1.3.3 Initializing a BSS

Upon receipt of an MLME-Start.request, a STA shall determine the BSS's BSSID (as described in 11.1.3), select channel synchronization information, select a beacon period, initialize and start its TSF timer, and begin transmitting beacons.

11.1.3.4 Synchronizing with a BSS

Upon receipt of an MLME-Join.request, a STA shall adopt the BSSID, channel synchronization information, and TSF timer value of the parameters in the request. Upon receipt of a Beacon frame from the BSS, the MLME shall issue an MLME-Join.confirm indicating the operation was successful. If the JoinFailureTimeout expires prior to the receipt of a Beacon frame from the BSS, the MLME shall issue an MLME-Join.confirm indicating the operation was unsuccessful.

11.1.4 Adjusting STA timers

In the infrastructure network, STAs shall always adopt the timer in a beacon or probe response coming from the AP in their BSS.

In an IBSS, a STA shall always adopt the information in the contents of a Beacon or Probe Response frame when that frame contains a matching SSID and the value of the time stamp is later than the STA's TSF timer. In response to an MLME-Join.request, a STA shall initialize its TSF timer to 0 and shall not transmit a

beacon or probe response until it hears a beacon or probe response from a member of the IBSS with a matching SSID.

All Beacon and Probe Response frames carry a Timestamp field. A STA receiving such a frame from another STA in an IBSS with the same SSID shall compare the Timestamp field with its own TSF time. If the Timestamp field of the received frame is later than its own TSF time, the STA shall adopt all parameters contained in the Beacon frame.

11.1.5 Timing synchronization for frequency-hopping (FH) PHYs

NOTE—This subclause pertains only to STAs using an FH PHY.

The TSF described here provides a mechanism for STAs in an FH system to synchronize their transitions from one channel to another (their “hops”). Every STA shall maintain a table of all of the hopping sequences that are used in the system. All of the STAs in a BSS shall use the same hopping sequence. Each beacon and probe response includes the channel synchronization information necessary to determine the hop pattern and timing for the BSS.

STAs shall use their TSF timer to time the `aCurrentDwellTime`. The `aCurrentDwellTime` is the length of time that STAs shall stay on each frequency in their hopping sequence. Once STAs are synchronized, they have the same TSF timer value.

STAs in the BSS shall issue an appropriate PLME service primitive for the PHY in use to tune to the next frequency in the hopping sequence whenever

$$\text{TSF timer MOD } a\text{CurrentDwellTime} = 0$$

11.2 Power management

11.2.1 Power management in an infrastructure network

STAs changing Power Management mode shall inform the AP of this fact using the Power Management bits within the Frame Control field of transmitted frames. The AP shall not arbitrarily transmit MSDUs to STAs operating in a power-save (PS) mode, but shall buffer MSDUs and only transmit them at designated times.

The STAs that currently have buffered MSDUs within the AP are identified in a *traffic indication map* (TIM), which shall be included as an element within all beacons generated by the AP. A STA shall determine that an MSDU is buffered for it by receiving and interpreting a TIM.

STAs operating in PS modes shall periodically listen for beacons, as determined by the STA's ListenInterval and ReceivedTIMs parameters of the MLME-Power-Mgt.request primitive.

In a BSS operating under the DCF, or during the contention period of a BSS using the PCF, upon determining that an MSDU is currently buffered in the AP, a STA operating in the *PS mode* shall transmit a short PS-Poll frame to the AP, which shall respond with the corresponding buffered MSDU immediately, or acknowledge the PS-Poll and respond with the corresponding MSDU at a later time. If the TIM indicating the buffered MSDU is sent during a contention-free period (CFP), a CF-Pollable STA operating in the PS mode does not send a PS-Poll frame, but remains active until the buffered MSDU is received (or the CFP ends). If any STA in its BSS is in PS mode, the AP shall buffer all broadcast and multicast MSDUs and deliver them to all STAs immediately following the next Beacon frame containing a *delivery TIM* (DTIM) transmission.

A STA shall remain in its current Power Management mode until it informs the AP of a Power Management mode change via a successful frame exchange. Power Management mode shall not change during any single frame exchange sequence, as described in 9.7.

11.2.1.1 STA Power Management modes

A STA may be in one of two different power states:

- *Awake*: STA is fully powered.
- *Doze*: STA is not able to transmit or receive and consumes very low power.

The manner in which a STA transitions between these two power states shall be determined by the STA's Power Management mode. These modes are summarized in Table 23.

The Power Management mode of a STA is selected by the PowerManagementMode parameter of the MLME-POWERMGT.request. Once the STA updates its Power Management mode, the MLME shall issue an MLME-POWERMGT.confirm indicating the success of the operation.

Table 23—Power Management modes

Active mode or AM	STA may receive frames at any time. In Active mode, a STA shall be in the Awake state. A STA on the polling list of a PCF shall be in Active mode for the duration of the CFP.
Power Save or PS	STA listens to selected beacons (based upon the ListenInterval parameter of the MLME-Associate.request primitive) and sends PS-Poll frames to the AP if the TIM element in the most recent beacon indicates a directed MSDU buffered for that STA. The AP shall transmit buffered directed MSDUs to a PS STA only in response to a PS-Poll from that STA, or during the CFP in the case of a CF-Pollable PS STA. In PS mode, a STA shall be in the Doze state and shall enter the Awake state to receive selected beacons, to receive broadcast and multicast transmissions following certain received beacons, to transmit, and to await responses to transmitted PS-Poll frames or (for CF-Pollable STAs) to receive contention-free transmissions of buffered MSDUs.

To change Power Management modes, a STA shall inform the AP through a successful frame exchange initiated by the STA. The Power Management bit in the Frame Control field of the frame sent by the STA in this exchange indicates the Power Management mode that the STA shall adopt upon successful completion of the entire frame exchange.

A STA that is changing from Doze to Awake in order to transmit shall perform clear channel assessment (CCA) until a frame sequence is detected by which it can correctly set its NAV, or until a period of time equal to the ProbeDelay has transpired.

11.2.1.2 AP TIM transmissions

The TIM shall identify the STAs for which traffic is pending and buffered in the AP. This information is coded in a *partial virtual bitmap*, as described in 7.3.2.6. In addition, the TIM contains an indication whether broadcast/multicast traffic is pending. Every STA is assigned an Association ID code (AID) by the AP as part of the association process. AID 0 (zero) is reserved to indicate the presence of buffered broadcast/multicast MSDUs. The AP shall identify those STAs for which it is prepared to deliver buffered MSDUs by setting bits in the TIM's partial virtual bitmap that correspond to the appropriate SIDs.

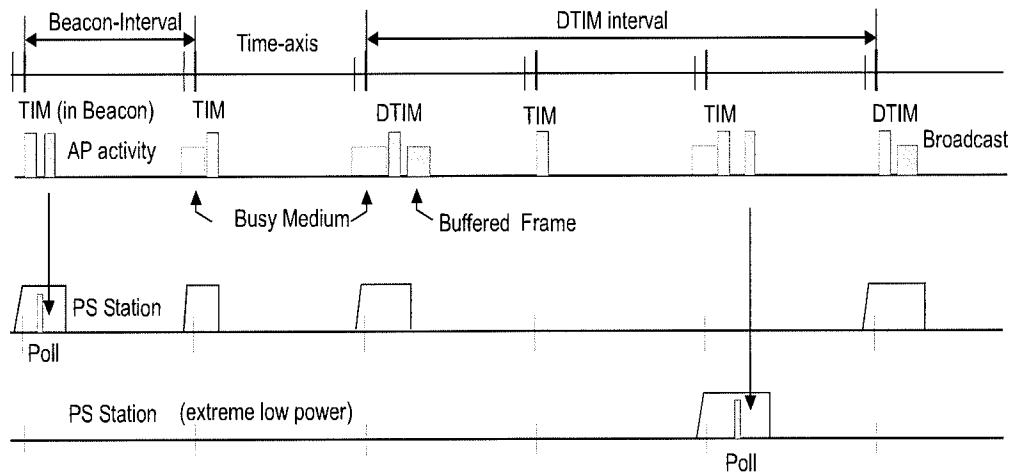
11.2.1.3 TIM types

Two different TIM types are distinguished: TIM and DTIM. After a DTIM, the AP shall send out the buffered broadcast/multicast MSDUs using normal frame transmission rules, before transmitting any unicast frames.

The AP shall transmit a TIM with every beacon. Every DTIMPeriod, a TIM of type “DTIM” is transmitted within a beacon, rather than an ordinary TIM.

Figure 67 illustrates the AP and STA activity under the assumption that a DTIM is transmitted once every three TIMs. The top line in Figure 67 represents the time axis, with the beacon interval shown together with a DTIM Interval of three beacon intervals. The second line depicts AP activity. The AP schedules beacons for transmission every beacon interval, but the beacons may be delayed if there is traffic at the TBTT. This is indicated as “busy medium” on the second line. For the purposes of this figure, the important fact about beacons is that they contain TIMs, some of which may be DTIMs. Note that the second STA with ReceiveDTIMs set to false does not power up its receiver for all DTIMs.

The third and fourth lines in Figure 67 depict the activity of two STAs operating with different power management requirements. Both STAs power-on their receivers whenever they need to listen for a TIM. This is indicated as a ramp-up of the receiver power prior to the TBTT. The first STA, for example, powers up its receiver and receives a TIM in the first beacon; that TIM indicates the presence of a buffered MSDU for the receiving STA. The receiving STA then generates a PS-Poll frame, which elicits the transmission of the buffered data MSDU from the AP. Broadcast and multicast MSDUs are sent by the AP subsequent to the transmission of a beacon containing a DTIM. The DTIM is indicated by the DTIM count field of the TIM element having a value of 0.



Example: DTIM at every 3 TIM intervals

Figure 67—Infrastructure power management operation (no PCF operating)

11.2.1.4 AP operation during the contention period

APs shall maintain a Power Management status for each currently associated STA that indicates in which Power Management mode the STA is currently operating. An AP shall, depending on the Power Management mode of the STA, temporarily buffer the MSDU or management frame destined to the STA. No MSDUs or management frames received for STAs operating in the Active mode shall be buffered for power management reasons.

- a) MSDUs, or management frames destined for PS STAs, shall be temporarily buffered in the AP. The algorithm to manage this buffering is beyond the scope of this standard.
- b) MSDUs, or management frames destined for STAs in the Active mode, shall be directly transmitted.
- c) At every beacon interval, the AP shall assemble the partial virtual bitmap containing the buffer status per destination for STAs in the PS mode, and shall send this out in the TIM field of the beacon. The bit for AID 0 (zero) shall be set whenever broadcast or multicast traffic is buffered.
- d) All broadcast/multicast MSDUs, with the Order bit in the Frame Control field clear, shall be buffered if any associated STAs are in PS mode.
- e) Immediately after every DTIM, the AP shall transmit all buffered broadcast/multicast MSDUs. The More Data field of each broadcast/multicast frame shall be set to indicate the presence of further buffered broadcast/multicast MSDUs. If the AP is unable to transmit all of the buffered broadcast/multicast MSDUs before the TBTT following the DTIM, the AP shall indicate that it will continue to deliver the broadcast/multicast MSDUs by setting the bit for AID 0 (zero) of the TIM element of every Beacon frame, until all buffered broadcast/multicast frames have been transmitted.
- f) A single buffered MSDU or management frame for a STA in the PS mode shall be forwarded to the STA after a PS-Poll has been received from that STA. The More Data field shall be set to indicate the presence of further buffered MSDUs or management frames for the polling STA. Further PS-Poll frames from the same STA shall be acknowledged and ignored until the MSDU or management frame has either been successfully delivered, or presumed failed due to maximum retries being exceeded. This prevents a retried PS-Poll from being treated as a new request to deliver a buffered frame.
- g) An AP shall have an aging function to delete pending traffic when it is buffered for an excessive time period.
- h) Whenever an AP is informed that a STA changes to the Active mode, then the AP shall send buffered MSDUs and management frames (if any exist) to that STA without waiting for a PS-Poll.

11.2.1.5 AP operation during the CFP

APs shall maintain a Power Management status for each currently associated CF-Pollable STA that indicates in which Power Management mode the STA is currently operating. An AP shall, for STAs in PS mode, temporarily buffer the MSDU destined to the STA.

- a) MSDUs destined for PS STAs shall be temporarily buffered in the AP. The algorithm to manage this buffering is beyond the scope of this standard.
- b) MSDUs destined to STAs in the Active mode shall be transmitted as defined in Clause 9.
- c) Prior to every CFP, and at each beacon interval within the CFP, the AP shall assemble the partial virtual bitmap containing the buffer status per destination for STAs in the PS mode, set the bits in the partial virtual bitmap for STAs the point coordinator (PC) is intending to poll during this CFP, and shall send this out in the TIM field of the DTIM. The bit for AID 0 (zero) shall be set whenever broadcast or multicast traffic is buffered.
- d) All broadcast and multicast MSDUs, with the Order bit in the Frame Control field clear, shall be buffered if any associated STAs are in the PS mode, whether or not those STAs are CF-Pollable.
- e) Immediately after every DTIM (Beacon frame with DTIM Count field of the TIM element equal to zero), the AP shall transmit all buffered broadcast and multicast frames. The More Data field shall be set to indicate the presence of further buffered broadcast/multicast MSDUs. If the AP is unable to transmit all of the buffered broadcast/multicast MSDUs before the TBTT following the DTIM, the AP shall indicate that it will continue to deliver the broadcast/multicast MSDUs by setting the bit for AID 0 (zero) of the TIM element of every Beacon frame, until all buffered broadcast/multicast frames have been transmitted.
- f) Buffered MSDUs or management frames for STAs in the PS mode shall be forwarded to the CF-Pollable STAs under control of the PC. Transmission of these buffered MSDUs or management frames shall begin immediately after transmission of buffered broadcast and multicast frames (if any), and shall occur in order by increasing AID of CF-Pollable STAs. A CF-Pollable STA for which the TIM element of the most recent beacon indicated buffered MSDUs or management frames shall

be in the Awake state at least until the receipt of a directed frame from the AP in which the Frame Control field does not indicate the existence of more buffered MSDUs or management frames. After acknowledging the last of the buffered MSDUs or management frames, the CF-Pollable STA operating in the PS mode may enter the Doze state until the next DTIM is expected.

- g) An AP shall have an aging function to delete pending traffic buffered for an excessive time period. The exact specification of the aging function is beyond the scope of this standard.
- h) Whenever an AP detects that a CF-Pollable STA has changed from the PS mode to the Active mode, then the AP shall queue any buffered frames addressed to that STA for transmission to that CF-Pollable STA as directed by the AP's PC function (PCF).

11.2.1.6 Receive operation for STAs in PS mode during the contention period

STAs in PS mode shall operate as follows to receive an MSDU or management frame from the AP when no PC is operating and during the contention period when a PC is operating.

- a) STAs shall wake up early enough to be able to receive the next scheduled beacon after ListenInterval from the last TBTT.
- b) When a STA detects that the bit corresponding to its AID is set in the TIM, the STA shall issue a PS-Poll to retrieve the buffered MSDU or management frame. If more than one bit is set in the TIM, the PS-Poll shall be transmitted after a random delay uniformly distributed between zero and aCWmin.
- c) The STA shall remain in the Awake state until it receives the response to its poll, or it receives another beacon whose TIM indicates that the AP does not have any MSDUs or management frames buffered for this STA. If the bit corresponding to the STA's AID is set in the subsequent TIM, the STA shall issue another PS-Poll to retrieve the buffered MSDU or management frame(s).
- d) If the More Data field in the received MSDU or management frame indicates that more traffic for that STA is buffered, the STA, at its convenience, shall Poll until no more MSDUs or management frames are buffered for that STA.
- e) When ReceiveDTIMs is true, the STA shall wake up early enough to be able to receive every DTIM. A STA receiving broadcast/multicast MSDUs shall remain awake until the More Data field of the broadcast/multicast MSDUs indicates there are no further buffered broadcast/multicast MSDUs, or until a TIM is received indicating there are no more buffered broadcast/multicast MSDUs.

11.2.1.7 Receive operation for STAs in PS mode during the CFP

STAs in PS mode that are associated as CF-Pollable shall operate as follows in a BSS with an active PC to receive MSDUs or management frames from the AP during the CFP:

- a) STAs shall enter the Awake state so as to receive the Beacon frame (which contains a DTIM) at the start of each CFP.
- b) To receive broadcast/multicast MSDUs, the STA shall wake up early enough to be able to receive every DTIM that may be sent during the CFP. A STA receiving broadcast/multicast MSDUs shall remain awake until the More Data field of the broadcast/multicast MSDUs indicates there are no further buffered broadcast/multicast MSDUs, or until a TIM is received indicating there are no more broadcast/multicast MSDUs buffered.
- c) When a STA detects that the bit corresponding to its AID is set in the DTIM at the start of the CFP (or in a subsequent TIM during the CFP), the STA shall remain in the Awake state for at least that portion of the CFP through the time that the STA receives a directed MSDU or management frame from the AP with the More Data field in the Frame Control field indicating that no further traffic is buffered.
- d) If the More Data field in the Frame Control field of the last MSDU or management frame received from the AP indicates that more traffic for the STA is buffered, then, when the CFP ends, the STA may remain in the Awake state and transmit PS-Poll frames during the contention period to request the delivery of additional buffered MSDU or management frames, or may enter the Doze state

during the contention period (except at TBTTs for DTIMs expected during the contention period), awaiting the start of the next CFP.

11.2.1.8 STAs operating in the Active mode

A STA operating in this mode shall have its receiver activated continuously; it does not need to interpret the traffic announcement part of the beacons.

11.2.1.9 AP aging function

The AP shall have an aging function to delete buffered traffic when it has been buffered for an excessive period of time. That function shall be based on the ListenInterval parameter of the MLME-Associate.request primitive of the STA for which the traffic is buffered. The AP aging function shall not cause the buffered traffic to be discarded after any period that is shorter than the ListenInterval of the STA for which the traffic is buffered. The exact specification of the aging function is beyond the scope of this standard.

11.2.2 Power management in an IBSS

This subclause specifies the power management mechanism for use within an IBSS.

11.2.2.1 Basic approach

The basic approach is similar to the infrastructure case in that the STAs are synchronized, and multicast MSDUs and those MSDUs that are to be transmitted to a power-conserving STA are first announced during a period when all STAs are awake. The announcement is done via an ad hoc traffic indication message (ATIM). A STA in the PS mode shall listen for these announcements to determine if it needs to remain in the awake state.

When an MSDU is to be transmitted to a destination STA that is in a PS mode, the transmitting STA first transmits an ATIM frame during the ATIM Window, in which all the STAs including those operating in a PS mode are awake. The ATIM Window is defined as a specific period of time, defined by aATIMWindow, following a TBTT, during which only Beacon or ATIM frames shall be transmitted. ATIM transmission times are randomized, after a Beacon frame is either transmitted or received by the STA, using the backoff procedure with the contention window equal to aCWminx. Directed ATIMs shall be acknowledged. If a STA transmitting a directed ATIM does not receive an acknowledgment, the STA shall execute the backoff procedure for retransmission of the ATIM. Multicast ATIMs shall not be acknowledged.

If a STA receives a directed ATIM frame during the ATIM Window, it shall acknowledge the directed ATIM and stay awake for the entire beacon interval waiting for the announced MSDU(s) to be received. If a STA does not receive an ATIM, it may enter the Doze state at the end of the ATIM Window. Transmissions of MSDUs announced by ATIMs are randomized after the ATIM Window, using the backoff procedure described in Clause 9.

It is possible that an ATIM may be received from more than one STA, and that a STA that receives an ATIM may receive more than a single MSDU from the transmitting STA. ATIM frames are only addressed to the destination STA of the MSDU.

An ATIM for a broadcast or multicast MSDU shall have a destination address identical to that of the MSDU.

After the ATIM interval, only those directed MSDUs that have been successfully announced with an acknowledged ATIM, and broadcast/multicast MSDUs that have been announced with an ATIM, shall be transmitted to STAs in the PS mode. Transmission of these frames shall be done using the normal DCF access procedure.

Figure 68 illustrates the basic power-save operation.

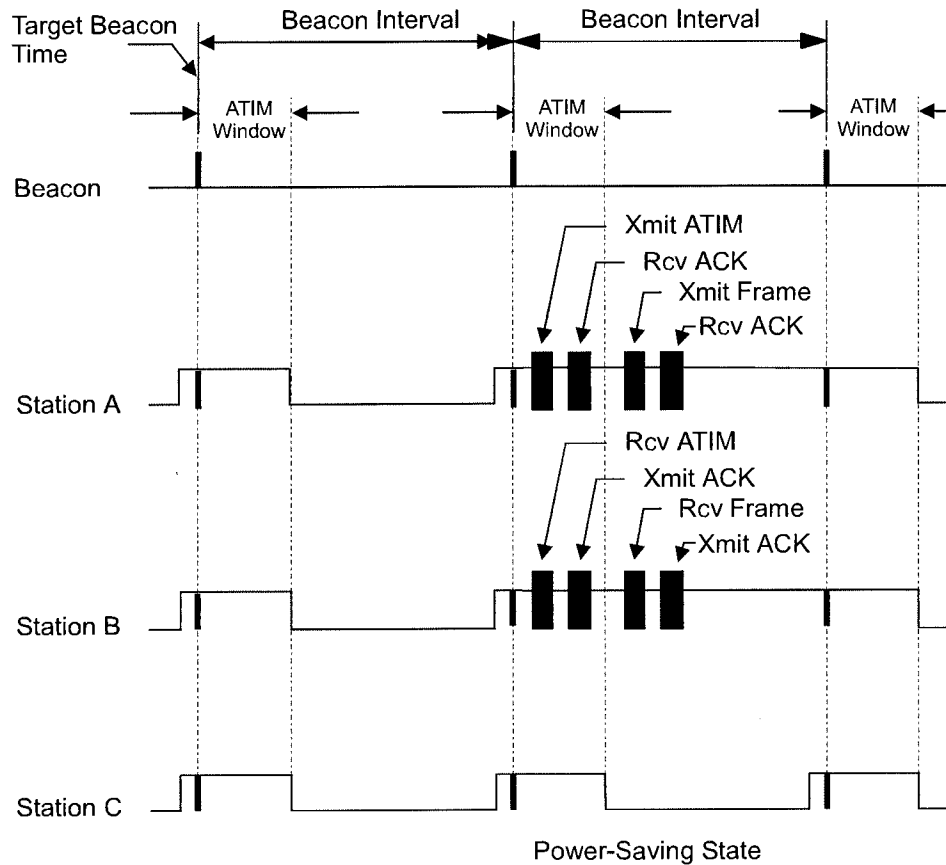


Figure 68—Power management in an IBSS—Basic operation

The estimated power-saving state of another STA may be based on the power management information transmitted by that STA and on additional information available locally, such as a history of failed transmission attempts. The use of RTS/CTS in an IBSS may reduce the number of transmissions to a STA that is in PS mode. If an RTS is sent and a CTS is not received, the transmitting STA may assume that the destination STA is in PS mode. The method of estimating the power management state of other STAs in the IBSS is outside the scope of this standard.

11.2.2.2 Initialization of power management within an IBSS

The following procedure shall be used to initialize power management within a new IBSS, or to learn about the power management being used within an existing IBSS.

- a) A STA joining an existing IBSS by the procedure in 11.1.3.3 shall update its ATIM Window with the value contained in the ATIM Window field of the IBSS Parameter Set element within the Beacon or Probe Response management frame received during the scan procedure.
- b) A STA creating a new IBSS by the procedure in 11.1.3.3 shall set the value of the ATIM Window field of the IBSS Parameter Set element within the Beacon management frames transmitted to the value of its ATIM Window.
- c) The start of the ATIM Window shall be the TBTT, defined in 11.1.2.2. The end of the ATIM Window shall be defined as

TSF timer MOD BeaconInterval = ATIMWindow.

- d) The ATIM Window period shall be static during the lifetime of the IBSS.
- e) An ATIM Window value of zero shall indicate that power management is not in use within the IBSS.

11.2.2.3 STA power state transitions

A STA may enter PS mode if and only if the value of the ATIM Window in use within the IBSS is greater than zero. A STA shall set the Power Management subfield in the Frame Control field of MSDUs that it transmits according to the procedure in 7.1.3.1.7.

A STA in PS mode shall transition between Awake and Doze states according to the following rules:

- a) If a STA is operating in PS mode, it shall enter the Awake state prior to each TBTT.
- b) If a STA receives a directed ATIM management frame containing its individual address, or a multicast ATIM management frame during the ATIM Window it shall remain in the Awake state until the end of the next ATIM Window.
- c) If a STA transmits a Beacon or an ATIM management frame, it shall remain in the Awake state until the end of the next ATIM Window regardless of whether an acknowledgment is received for the ATIM.
- d) If the STA has not transmitted an ATIM and does not receive either a directed ATIM management frame containing its individual address, or a multicast ATIM management frame during the ATIM Window, it may return to the Doze state following the end of the current ATIM Window.

11.2.2.4 ATIM and frame transmission

If power management is in use within an IBSS, all STAs shall buffer MSDUs for STAs that are known to be in PS mode. The algorithm used for the estimation of the power management state of STAs within the IBSS is outside the scope of this standard. MSDUs may be sent to STAs in Active mode at any valid time.

- a) Following the reception or transmission of the beacon, during the ATIM Window, the STA shall transmit a directed ATIM management frame to each STA for which it has one or more buffered unicast MSDUs. If the STA has one or more buffered multicast MSDUs, with the Strictly Ordered bit clear, it shall transmit an appropriately addressed multicast ATIM frame. A STA transmitting an ATIM management frame shall remain awake for the entire current beacon interval.
- b) All STAs shall use the backoff procedure defined in 9.2.5.2 for transmission of the first ATIM following the beacon. All remaining ATIMs shall be transmitted using the conventional DCF access procedure.
- c) ATIM management frames shall only be transmitted during the ATIM Window.
- d) A STA shall transmit no frame types other than RTS, CTS, and ACK Control frames and Beacon and ATIM management frames during the ATIM Window.
- e) Directed ATIM management frames shall be acknowledged. If no acknowledgment is received, the ATIM shall be retransmitted using the conventional DCF access procedure. Multicast ATIM management frames shall not be acknowledged.
- f) If a STA is unable to transmit an ATIM during the ATIM Window, for example due to contention with other STAs, the STA shall retain the buffered MSDU(s) and attempt to transmit the ATIM during the next ATIM Window.
- g) Immediately following the ATIM Window, a STA shall begin transmission of buffered broadcast/multicast frames for which an ATIM was previously transmitted. Following the transmission of any broadcast/multicast frames, any MSDUs and management frames addressed to STAs for which an acknowledgment for a previously transmitted ATIM frame was received shall be transmitted. All STAs shall use the backoff procedure defined in 9.2.5.2 for transmission of the first frame following the ATIM Window. All remaining frames shall be transmitted using the conventional DCF access procedure.

- h) A buffered MSDU may be transmitted using fragmentation. If an MSDU has been partially transmitted when the next beacon frame is sent, the STA shall retain the buffered MSDU and announce the remaining fragments by transmitting an ATIM during the next ATIM Window.
- i) If an STA is unable to transmit a buffered MSDU during the beacon interval in which it was announced, for example due to contention with other STAs, the STA shall retain the buffered MSDU and announce the MSDU again by transmitting an ATIM during the next ATIM Window.
- j) Following the transmission of all buffered MSDUs, a STA may transmit MSDUs without announcement to STAs that are known to be in the Awake state for the current beacon interval due to an appropriate ATIM management or Beacon frame having been transmitted or received.
- k) A STA may discard frames buffered for later transmission to power-saving STAs if the STA determines that the frame has been buffered for an excessive amount of time or if other conditions internal to the STA implementation make it desirable to discard buffered frames (for example, buffer starvation). In no case shall a frame be discarded that has been buffered for less than aBeaconPeriod. The algorithm to manage this buffering is beyond the scope of this standard.

11.3 Association and reassociation

This subclause defines how a STA associates and reassociates with an AP.

11.3.1 STA association procedures

Upon receipt of an MLME-ASSOCIATE.request, a STA shall associate with an AP via the following procedure:

- a) The STA shall transmit an association request to an AP with which that STA is authenticated.
- b) If an Association Response frame is received with a status value of “successful,” the STA is now associated with the AP and the MLME shall issue an MLME-ASSOCIATE.confirm indicating the successful completion of the operation.
- c) If an Association Response frame is received with a status value other than “successful” or the AssociateFailureTimeout expires, the STA is not associated with the AP and the MLME shall issue an MLME-ASSOCIATE.confirm indicating the failure of the operation.

11.3.2 AP association procedures

An AP shall operate as follows in order to support the association of STAs.

- a) Whenever an Association Request frame is received from a STA and the STA is authenticated, the AP shall transmit an association response with a status code as defined in 7.3.1.9. If the status value is “successful,” the Association ID assigned to the STA shall be included in the response. If the STA is not authenticated, the AP shall transmit a Deauthentication frame to the STA.
- b) When the association response with a status value of “successful” is acknowledged by the STA, the STA is considered to be associated with this AP.
- c) The AP shall inform the distribution system (DS) of the association and the MLME shall issue an MLME-ASSOCIATE.indication.

11.3.3 STA reassociation procedures

Upon receipt of an MLME-REASSOCIATE.request, a STA shall reassociate with an AP via the following procedure:

- a) The STA shall transmit a Reassociation Request frame to an AP.

- b) If a Reassociation Response frame is received with a status value of “successful,” the STA is now associated with the AP and the MLME shall issue an MLME-REASSOCIATE.confirm indicating the successful completion of the operation.
- c) If a Reassociation Response frame is received with a status value other than “successful” or the ReassociateFailureTimeout expires, the STA is not associated with the AP and the MLME shall issue an MLME-REASSOCIATE.confirm indicating the failure of the operation.

11.3.4 AP reassociation procedures

An AP shall operate as follows in order to support the reassociation of STAs.

- a) Whenever a Reassociation Request frame is received from a STA and the STA is authenticated, the AP shall transmit a reassociation response with a status value as defined in 7.3.1.9. If the status value is “successful,” the Association ID assigned to the STA shall be included in the response. If the STA is not authenticated, the AP shall transmit a Deauthentication frame to the STA.
- b) When the reassociation response with a status value of “successful” is acknowledged by the STA, the STA is considered to be associated with this AP.
- c) The AP shall inform the DS of the reassociation and the MLME shall issue an MLME-REASSOCIATE.indication.

11.4 Management information base (MIB) definitions

The MIB comprises the managed objects, attributes, actions, and notifications required to manage a station. The definition of these managed objects, attributes, actions, and notifications, as well as their structure, is presented in Annex D.

12. Physical layer (PHY) service specification

12.1 Scope

The PHY services provided to the IEEE 802.11 wireless LAN MAC are described in this clause. Different PHYs are defined as part of the IEEE 802.11 standard. Each PHY can consist of two protocol functions as follows:

- a) A physical layer convergence function, which adapts the capabilities of the physical medium dependent (PMD) system to the PHY service. This function is supported by the physical layer convergence procedure (PLCP), which defines a method of mapping the IEEE 802.11 MAC sublayer protocol data units (MPDUs) into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.
- b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a wireless medium (WM) between two or more STAs.

Each PMD sublayer may require the definition of a unique PLCP. If the PMD sublayer already provides the defined PHY services, the physical layer convergence function might be null.

12.2 PHY functions

The protocol reference model for the IEEE 802.11 architecture is shown in Figure 11. Most PHY definitions contain three functional entities: the PMD function, the physical layer convergence function, and the layer management function.

The PHY service is provided to the MAC entity at the STA through a service access point (SAP), called the PHY-SAP, as shown in Figure 11. A set of primitives might also be defined to describe the interface between the physical layer convergence protocol sublayer and the PMD sublayer, called the PMD-SAP.

12.3 Detailed PHY service specifications

12.3.1 Scope and field of application

The services provided by the PHY to the IEEE 802.11 MAC are specified in this subclause. These services are described in an abstract way and do not imply any particular implementation or exposed interface.

12.3.2 Overview of the service

The PHY function as shown in Figure 11 is separated into two sublayers: the PLCP sublayer and the PMD sublayer. The function of the PLCP sublayer is to provide a mechanism for transferring MPDUs between two or more STAs over the PMD sublayer.

12.3.3 Overview of interactions

The primitives associated with communication between the IEEE 802.11 MAC sublayer and the IEEE 802.11 PHY fall into two basic categories:

- a) Service primitives that support MAC peer-to-peer interactions;
- b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

12.3.4 Basic service and options

All of the service primitives described here are considered mandatory unless otherwise specified.

12.3.4.1 PHY-SAP peer-to-peer service primitives

Table 24 indicates the primitives for peer-to-peer interactions.

Table 24—PHY-SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm
PHY-DATA	X	X	X

12.3.4.2 PHY-SAP sublayer-to-sublayer service primitives

Table 25 indicates the primitives for sublayer-to-sublayer interactions.

Table 25—PHY-SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm
PHY-TXSTART	X		X
PHY-TXEND	X		X
PHY-CCARESET	X		X
PHY-CCA		X	
PHY-RXSTART		X	
PHY-RXEND		X	

12.3.4.3 PHY-SAP service primitives parameters

Table 26 shows the parameters used by one or more of the PHY-SAP service primitives.

Table 26—PHY-SAP service primitive parameters

Parameter	Associated primitive	Value
DATA	PHY-DATA.request PHY-DATA.indication	Octet value X'00'–X'FF'
TXVECTOR	PHY-TXSTART.request	A set of parameters
STATUS	PHY-CCA.indication	BUSY, IDLE
RXVECTOR	PHY-RXSTART.indication	A set of parameters
RXERROR	PHY-RXEND.indication	NoError, FormatViolation, Carrier-Lost, UnsupportedRate

12.3.4.4 Vector descriptions

Several service primitives include a parameter vector. This vector is a list of parameters that may vary depending on the PHY type. Table 27 lists the parameter values required by the MAC or PHY in each of the parameter vectors. Parameters in the vectors that are management rather than MAC may be specific to the PHY and are listed in the clause covering that PHY.

Table 27—Vector descriptions

Parameter	Associate vector	Value
DATARATE	TXVECTOR, RXVECTOR	PHY dependent. The name of the field used to specify the Tx data rate and report the Rx data rate may vary for different PHYs.
LENGTH	TXVECTOR, RXVECTOR	PHY dependent

12.3.5 PHY-SAP detailed service specification

The following subclause describes the services provided by each PHY sublayer primitive.

12.3.5.1 PHY-DATA.request

12.3.5.1.1 Function

This primitive defines the transfer of an octet of data from the MAC sublayer to the local PHY entity.

12.3.5.1.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-DATA.request (DATA)

The DATA parameter is an octet of value X'00' through X'FF'.

12.3.5.1.3 When generated

This primitive is generated by the MAC sublayer to transfer an octet of data to the PHY entity. This primitive can only be issued following a transmit initialization response (PHY-TXSTART.confirm) from the PHY layer.

12.3.5.1.4 Effect of receipt

The receipt of this primitive by the PHY entity causes the PLCP transmit state machine to transmit an octet of data. When the PHY entity receives the octet, it will issue a PHY-DATA.confirm to the MAC sublayer.

12.3.5.2 PHY-DATA.indication

12.3.5.2.1 Function

This primitive indicates the transfer of data from the PHY sublayer to the local MAC entity.

12.3.5.2.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-DATA.indication (DATA)

The DATA parameter is an octet of value X'00' through X'FF'.

12.3.5.2.3 When generated

The PHY-DATA.indication is generated by a receiving PHY entity to transfer the received octet of data to the local MAC entity. The time between receipt of the last bit of the provided octet from the wireless medium and the receipt of this primitive by the MAC entity will be the sum of aRXRFDelay + aRxPLCPDelay.

12.3.5.2.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

12.3.5.3 PHY-DATA.confirm

12.3.5.3.1 Function

This primitive is issued by the PHY sublayer to the local MAC entity to confirm the transfer of data from the MAC entity to the PHY sublayer.

12.3.5.3.2 Semantics of the service primitive

The semantics of the primitive are as follows:

PHY-DATA.confirm

This primitive has no parameters.

12.3.5.3.3 When generated

This primitive will be issued by the PHY sublayer to the MAC entity whenever the PLCP has completed the transfer of data from the MAC entity to the PHY sublayer. The PHY sublayer will issue this primitive in response to every PHY-DATA.request primitive issued by the MAC sublayer.

12.3.5.3.4 Effect of receipt

The receipt of this primitive by the MAC will cause the MAC to start the next MAC entity request.

12.3.5.4 PHY-TXSTART.request

12.3.5.4.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to start the transmission of an MPDU.

12.3.5.4.2 Semantics of the service primitive

The primitive provides the following parameters:

PHY-TXSTART.request (TXVECTOR)

The TXVECTOR represents a list of parameters that the MAC sublayer provides to the local PHY entity in order to transmit an MPDU. This vector contains both PLCP and PHY management parameters. The required PHY parameters are listed in 12.3.4.4.

12.3.5.4.3 When generated

This primitive will be issued by the MAC sublayer to the PHY entity whenever the MAC sublayer needs to begin the transmission of an MPDU.

12.3.5.4.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity will be to start the local transmit state machine.

12.3.5.5 PHY-TXSTART.confirm

12.3.5.5.1 Function

This primitive is issued by the PHY sublayer to the local MAC entity to confirm the start of a transmission. The PHY sublayer will issue this primitive in response to every PHY-TXSTART.request primitive issued by the MAC sublayer.

12.3.5.5.2 Semantics of the service primitive

The semantics of the primitive are as follows:

PHY-TXSTART.confirm

There are no parameters associated with this primitive.

12.3.5.5.3 When generated

This primitive will be issued by the PHY sublayer to the MAC entity whenever the PHY has received a PHY-TXSTART.request from the MAC entity and is ready to begin receiving data octets.

12.3.5.5.4 Effect of receipt

The receipt of this primitive by the MAC entity will cause the MAC to start the transfer of data octets.

12.3.5.6 PHY-TXEND.request

12.3.5.6.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity that the current transmission of the MPDU be completed.

12.3.5.6.2 Semantics of the service primitive

The semantics of the primitive are as follows:

PHY-TXEND.request

There are no parameters associated with this primitive.

12.3.5.6.3 When generated

This primitive will be generated whenever the MAC sublayer has received the last PHY-DATA.confirm from the local PHY entity for the MPDU currently being transferred.

12.3.5.6.4 Effect of receipt

The effect of receipt of this primitive by the local PHY entity will be to stop the transmit state machine.

12.3.5.7 PHY-TXEND.confirm

12.3.5.7.1 Function

This primitive is issued by the PHY sublayer to the local MAC entity to confirm the completion of a transmission. The PHY sublayer issues this primitive in response to every PHY-TXEND.request primitive issued by the MAC sublayer.

12.3.5.7.2 Semantics of the service primitive

The semantics of the primitive are as follows:

PHY-TXEND.confirm

There are no parameters associated with this primitive.

12.3.5.7.3 When generated

This primitive will be issued by the PHY sublayer to the MAC entity whenever the PHY has received a PHY-TXEND.request immediately after transmitting the end of the last bit of the last data octet indicating that the last data octet has been transferred.

12.3.5.7.4 Effect of receipt

The receipt of this primitive by the MAC entity provides the time reference for the contention backoff protocol.

12.3.5.8 PHY-CCARESET.request

12.3.5.8.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to reset the clear channel assessment (CCA) state machine.

12.3.5.8.2 Semantics of the service primitive

The semantics of the primitives are as follows:

PHY-CCARESET.request

There are no parameters associated with this primitive.

12.3.5.8.3 When generated

This primitive is generated by the MAC sublayer for the local PHY entity at the end of a NAV timer. This request can be used by some PHY implementations that may synchronize antenna diversity with slot timings.

12.3.5.8.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to reset the PLCP CS/CCA assessment timers to the state appropriate for the end of a received frame.

12.3.5.9 PHY-CCARESET.confirm**12.3.5.9.1 Function**

This primitive is issued by the PHY sublayer to the local MAC entity to confirm that the PHY has reset the CCA state machine.

12.3.5.9.2 Semantics of the service primitive

The semantics of the primitives are as follows:

PHY-CCARESET.request

There are no parameters associated with this primitive.

12.3.5.9.3 When generated

This primitive is issued by the PHY sublayer to the MAC entity whenever the PHY has received a PHY-CCARESET.request.

12.3.5.9.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

12.3.5.10 PHY-CCA.indication**12.3.5.10.1 Function**

This primitive is an indication by the PHY sublayer to the local MAC entity of the current state of the medium.

12.3.5.10.2 Semantics of the service primitive

The primitive provides the following parameter:

PHY-CCA.indication (STATE)

The STATE parameter can be one of two values: BUSY or IDLE. The parameter value is BUSY if the channel assessment by the PHY sublayer determines that the channel is not available. Otherwise, the value of the parameter is IDLE.

12.3.5.10.3 When generated

This primitive is generated every time the status of the channel changes from channel idle to channel busy or from channel busy to channel idle. This includes the period of time when the PHY sublayer is receiving data. The PHY sublayer maintains the channel busy indication until the period indicated by the length field in a valid PLCP Header has expired.

12.3.5.10.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

12.3.5.11 PHY-RXSTART.indication

12.3.5.11.1 Function

This primitive is an indication by the PHY sublayer to the local MAC entity that the PLCP has received a valid start frame delimiter (SFD) and PLCP Header.

12.3.5.11.2 Semantics of the service primitive

The primitive provides the following parameter:

PHY-RXSTART.indication (RXVECTOR)

The RXVECTOR represents a list of parameters that the PHY sublayer provides the local MAC entity upon receipt of a valid PLCP Header. This vector may contain both MAC and MAC management parameters. The required parameters are listed in 12.3.4.4.

12.3.5.11.3 When generated

This primitive is generated by the local PHY entity to the MAC sublayer whenever the PHY has successfully validated the PLCP Header error check CRC at the start of a new PLCP PDU.

12.3.5.11.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

12.3.5.12 PHY-RXEND.indication

12.3.5.12.1 Function

This primitive is an indication by the PHY sublayer to the local MAC entity that the MPDU currently being received is complete.

12.3.5.12.2 Semantics of the service primitive

The primitive provides the following parameter:

PHY-RXEND.indication (RXERROR)

The RXERROR parameter can convey one or more of the following values: NoError, FormatViolation, CarrierLost, or UnsupportedRate. A number of error conditions may occur after the PLCP's receive state machine has detected what appears to be a valid preamble and SFD. The following describes the parameter returned for each of those error conditions.

- *NoError*. This value is used to indicate that no error occurred during the receive process in the PLCP.
- *FormatViolation*. This value is used to indicate that the format of the received PLCPPDU was in error.
- *CarrierLost*. This value is used to indicate that during the reception of the incoming MPDU, the carrier was lost and no further processing of the MPDU can be accomplished.
- *UnsupportedRate*. This value is used to indicate that during the reception of the incoming PLCPPDU, a nonsupported data rate was detected.

12.3.5.12.3 When generated

This primitive is generated by the PHY sublayer for the local MAC entity to indicate that the receive state machine has completed a reception with or without errors.

12.3.5.12.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

13. PHY management

The MIB comprises the managed objects, attributes, actions, and notifications required to manage a station. The definition of these managed objects, attributes, actions, and notifications, as well as their structure, is presented in Annex D.

14. Frequency-Hopping spread spectrum (FHSS) PHY specification for the 2.4 GHz Industrial, Scientific, and Medical (ISM) band

14.1 Overview

14.1.1 Overview of FHSS PHY

The PHY services provided to the IEEE 802.11 wireless LAN MAC for the 2.4 GHz frequency-hopping spread spectrum (FHSS) system are described in this clause. The FHSS PHY consists of the following two protocol functions:

- a) A physical layer convergence function, which adapts the capabilities of the physical medium dependent (PMD) system to the PHY service. This function is supported by the physical layer convergence procedure (PLCP), which defines a method of mapping the IEEE 802.11 MAC sublayer protocol data units (MPDUs) into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.
- b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a wireless medium (WM) between two or more STAs.

14.1.2 FHSS PHY functions

The 2.4 GHz FHSS PHY architecture is shown in Figure 11. The FHSS PHY contains three functional entities: the PMD function, the physical layer convergence function, and the physical layer management function. Each of these functions is described in detail in the following subclauses.

The FHSS PHY service is provided to the MAC entity at the STA through a PHY service access point (SAP) called the PHY-SAP, as shown in Figure 11. A set of primitives might also be defined that describe the interface between the physical layer convergence protocol sublayer and the PMD sublayer, called the PMD-SAP.

14.1.2.1 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PHY convergence sublayer is defined. This function simplifies provision of a PHY service interface to the IEEE 802.11 MAC services.

14.1.2.2 Physical layer management entity (PLME)

The PLME performs management of the local PHY functions in conjunction with the MAC management entity.

14.1.2.3 PMD sublayer

The PMD sublayer provides a transmission interface used to send and receive data between two or more STAs.

14.1.3 Service specification method and notation

The models represented by state diagrams in the following subclauses are intended as the primary specifications of the functions provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology.

The service of a layer or sublayer is the set of capabilities that it offers to a user in the next higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition of service is independent of any particular implementation.

14.2 FHSS PHY-specific service parameter lists

14.2.1 Overview

The architecture of the IEEE 802.11 MAC is intended to be PHY independent. Some PHY implementations require medium management state machines running in the MAC sublayer in order to meet certain PMD requirements. These PHY-dependent MAC state machines reside in a sublayer defined as the MAC sublayer management entity (MLME). The MLME in certain PMD implementations may need to interact with the physical layer management entity (PLME) as part of the normal PHY-SAP primitives. These interactions are defined by the PLME parameter list currently defined in the PHY Service Primitives as TXVECTOR and RXVECTOR. The list of these parameters and the values they may represent are defined in the specific PHY specifications for each PMD. This subclause addresses the TXVECTOR and RXVECTOR for the FHSS PHY.

All of the values included in the TXVECTOR or RXVECTOR described in this subclause are considered mandatory unless otherwise specified. The 1 Mbit/s and 2 Mbit/s data rates are the only rates currently supported. Other indicated data rates are for possible future use.

14.2.2 TXVECTOR parameters

The parameters in Table 28 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request service primitive.

Table 28—TXVECTOR parameters

Parameter	Associate primitive	Value
LENGTH	PHY-TXSTART.request (TXVECTOR)	1–4095
DATARATE	PHY-TXSTART.request (TXVECTOR)	1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5

14.2.2.1 TXVECTOR LENGTH

The LENGTH parameter has the value of 1 to 4095. This parameter is used to indicate the number of octets in the MPDU that the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start a transmission.

14.2.2.2 TXVECTOR DATARATE

The DATARATE parameter describes the bit rate at which the PLCP should transmit the PSDU. Its value can be any of the rates as defined in Table 28, and supported by the conformant FH PHY.

14.2.3 RXVECTOR parameters

The parameters in Table 29 are defined as part of the RXVECTOR parameter list in the PHY-RXSTART.indicate service primitive.

Table 29—RXVECTOR parameters

Parameter	Associate primitive	Value
LENGTH	PHY-RXSTART.indicate (RXVECTOR)	1–4095
RSSI	PHY-RXSTART.indicate (RXVECTOR)	0–RSSI Max
DATARATE	PHY-RXSTART.request (RXVECTOR)	1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5

14.2.3.1 TRXVECTOR LENGTH

The LENGTH parameter has the value of 1 to 4095. This parameter is used to indicate the value contained in the LENGTH field that the PLCP has received in the PLCP Header. The MAC and PLCP will use this value to determine the number of octet transfers that will occur between the two sublayers during the transfer of the received PSDU.

14.2.3.2 RXVECTOR RSSI

The receive signal strength indicator (RSSI) is an optional parameter that has a value of 0 through RSSI Max. This parameter is a measure by the PHY sublayer of the energy observed at the antenna used to receive the current PPDU. RSSI shall be measured between the beginning of the start frame delimiter (SFD) and the end of the PLCP header error check (HEC). RSSI is intended to be used in a relative manner. Absolute accuracy of the RSSI reading is not specified.

14.3 FHSS PLCP sublayer

14.3.1 Overview

This subclause provides a convergence procedure to map MPDUs into a frame format designed for FHSS radio transceivers. The procedures for transmission, carrier sense, and reception are defined for single and multiple antenna diversity radios.

14.3.1.1 State diagram notation

The operation of the procedures can be described by state diagrams. Each diagram represents the domain and consists of a group of connected, mutually exclusive states. Only one state is active at any given time. Each state is represented by a rectangle as shown in Figure 69. These are divided into two parts by a horizontal line. In the upper part the state is identified by a name. The lower part contains the name of any signal that is generated. Actions described by short phrases are enclosed in brackets.

Each permissible transition between the states is represented graphically by an arrow from the initial to the terminal state. A transition that is global in nature (for example, an exit condition from all states to the IDLE or RESET state) is indicated by an open arrow. Labels on transitions are qualifiers that must be fulfilled before the transition will be taken. The label UCT designates an unconditional transition. Qualifiers described by short phrases are enclosed in parentheses.

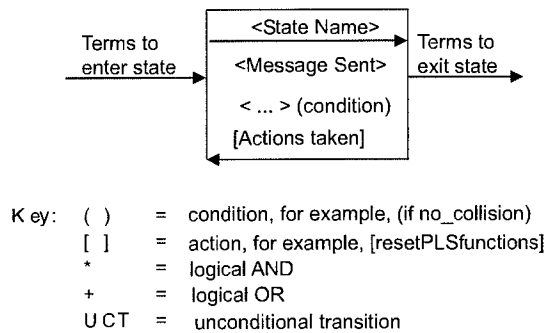


Figure 69—State diagram notation example

State transitions and sending and receiving of messages occur instantaneously. When a state is entered and the condition to leave that state is not immediately fulfilled, the state executes continuously, sending the messages and executing the actions contained in the state in a continuous manner.

Some devices described in this standard are allowed to have two or more ports. State diagrams capable of describing the operation of devices with an unspecified number of ports require qualifier notation that allows testing for conditions at multiple ports. The notation used is a term that includes a description in parentheses of which ports must meet the term for the qualifier to be satisfied (e.g., ANY and ALL). It is also necessary to provide for term-assignment statements that assign a name to a port that satisfies a qualifier. The following convention is used to describe a term-assignment statement that is associated with a transition:

- a) The character “:” (colon) is a delimiter used to denote that a term assignment statement follows.
- b) The character “<” (left arrow) denotes assignment of the value following the arrow to the term preceding the arrow.

The state diagrams contain the authoritative statement of the procedures they depict; when apparent conflicts between descriptive text and state diagrams arise, the state diagrams are to take precedence. This does not, however, override any explicit description in the text that has no parallel in the state diagrams.

The models presented by state diagrams are intended as the primary specifications to be provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology. It is the functional behavior of any unit that must match the standard, not its internal structure. The internal details of the model are useful only to the extent that they specify the external behavior clearly and precisely.

14.3.2 PLCP frame format

The PLCP protocol data unit (PPDU) frame format provides for the asynchronous transfer of MAC sublayer MPDUs from any transmitting STA to all receiving STAs within the wireless LAN’s BSS. The PPDU illustrated in Figure 70 consists of three parts: a PLCP Preamble, a PLCP Header, and a PSDU. The PLCP Preamble provides a period of time for several receiver functions. These functions include antenna diversity, clock and data recovery, and field delineation of the PLCP Header and the PSDU. The PLCP Header is used to specify the length of the whitened PSDU field and support any PLCP management information. The PPDU contains the PLCP Preamble, the PLCP Header, and the PSDU modified by the PPDU data whitener.

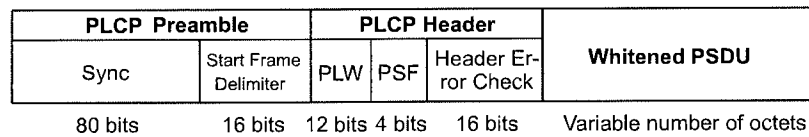


Figure 70—PLCP frame format

14.3.2.1 PLCP Preamble

The PLCP Preamble contains two separate subfields; the Preamble Synchronization (SYNC) field and the Start Frame Delimiter (SFD), to allow the PHY circuitry to reach steady-state demodulation and synchronization of bit clock and frame start.

14.3.2.1.1 Preamble SYNC field

The Preamble SYNC field is an 80-bit field containing an alternating zero-one pattern, transmitted starting with zero and ending with one, to be used by the PHY sublayer to detect a potentially receivable signal, select an antenna if diversity is utilized, and reach steady-state frequency offset correction and synchronization with the received packet timing.

14.3.2.1.2 Start Frame Delimiter (SFD)

The SFD consists of the 16-bit binary pattern 0000 1100 1011 1101 (transmitted leftmost bit first). The first bit of the SFD follows the last bit of the sync pattern. The SFD defines the frame timing.

14.3.2.2 PLCP Header field

The PLCP Header field contains three separate subfields: a 12-bit PSDU Length Word (PLW), a 4-bit PLCP Signaling field (PSF), and a 16-bit PLCP HEC field.

14.3.2.2.1 PSDU length word

The PSDU length word (PLW) is passed from the MAC as a parameter within the PHY-TXSTART.request primitive. The PLW specifies the number of octets contained in the PSDU. Its valid values are X'001'–X'FFF', representing counts of one to 4095 octets. The PLW is transmitted lsb first and msb last. The PLW is used by the receiving STA, in combination with the 32/33 coding algorithm specified in this clause, to determine the last bit in the packet.

14.3.2.2.2 PLCP Signaling field (PSF)

The 4-bit PSF is defined in Table 30. The PSF is transmitted bit 0 first and bit 3 last.

Table 30—PSF bit descriptions

Bit	Parameter name	Parameter values	Description																																				
0	Reserved	Default = 0	Reserved																																				
1:3	PLCP_BITRATE	<table style="border: none;"> <tr> <td><u>b1</u></td> <td><u>b2</u></td> <td><u>b3</u></td> <td>= Data Rate</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>= 1.0 Mbit/s,</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>= 1.5 Mbit/s,</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>= 2.0 Mbit/s,</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>= 2.5 Mbit/s,</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>= 3.0 Mbit/s,</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>= 3.5 Mbit/s,</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>= 4.0 Mbit/s,</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>= 4.5 Mbit/s</td> </tr> </table>	<u>b1</u>	<u>b2</u>	<u>b3</u>	= Data Rate	0	0	0	= 1.0 Mbit/s,	0	0	1	= 1.5 Mbit/s,	0	1	0	= 2.0 Mbit/s,	0	1	1	= 2.5 Mbit/s,	1	0	0	= 3.0 Mbit/s,	1	0	1	= 3.5 Mbit/s,	1	1	0	= 4.0 Mbit/s,	1	1	1	= 4.5 Mbit/s	This field indicates the data rate of the whitened PSDU from 1 Mbit/s to 4.5 Mbit/s in 0.5 Mbit/s increments.
<u>b1</u>	<u>b2</u>	<u>b3</u>	= Data Rate																																				
0	0	0	= 1.0 Mbit/s,																																				
0	0	1	= 1.5 Mbit/s,																																				
0	1	0	= 2.0 Mbit/s,																																				
0	1	1	= 2.5 Mbit/s,																																				
1	0	0	= 3.0 Mbit/s,																																				
1	0	1	= 3.5 Mbit/s,																																				
1	1	0	= 4.0 Mbit/s,																																				
1	1	1	= 4.5 Mbit/s																																				

14.3.2.2.3 Header Error Check (HEC) field

The HEC field is a 16-bit CCITT CRC-16 error detection field. The HEC uses the CCITT CRC-16 generator polynomial $G(x)$ as follows:

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

The HEC shall be the one's complement (modulo 2) of the following:

- a) The remainder of $x^k \times (x^{15} + x^{14} + \dots + x^2 + x^1 + 1)$ divided (modulo 2) by $G(x)$, where k is the number of bits in the PSF and PLW fields of the PLCP Header;
- b) The remainder after multiplication by x^{16} and then division (modulo 2) by $G(x)$ of the content (treated as a polynomial) of the PSF and PLW fields.

The HEC shall be transmitted with the coefficient of the highest term first.

As a typical implementation, at the transmitter, the initial remainder of the division is preset to all ones and is then modified by division of the PSF and PLW fields by the generator polynomial, $G(x)$. The one's complement of this remainder is inserted in the HEC field with the msb transmitted first.

At the receiver, the initial remainder of the division is again preset to all ones. The division of the received PSF, PLW, and HEC fields by the generator polynomial, $G(x)$, results, in the absence of transmission errors, in a unique nonzero value, which is the following polynomial $R(x)$:

$$R(x) = x^{12} + x^{11} + x^{10} + x^8 + x^3 + x^2 + x^1 + 1$$

14.3.2.3 PLCP data whitener

The PLCP data whitener uses a length-127 frame-synchronous scrambler followed by a 32/33 bias-suppression encoding to randomize the data and to minimize the data dc bias and maximum run lengths. Data octets are placed in the transmit serial bit stream lsb first and msb last. The frame synchronous scrambler uses the generator polynomial $S(x)$ as follows:

$$S(x) = x^7 + x^4 + 1$$

and is illustrated in Figure 71. The 127-bit sequence generated repeatedly by the scrambler is (leftmost bit used first) 00001110 11110010 11001001 00000010 00100110 00101110 10110110 00001100 11010100 11100111 10110100 00101010 11111010 01010001 10111000 11111111. The same scrambler is used to scramble transmit data and to descramble receive data. The data whitening starts with the first bit of the PSDU, which follows the last bit of the PLCP Header. The specific bias suppression encoding and decoding method used is defined in Figure 75 and Figure 80. The format of the packet after data whitening is as shown in Figure 72.

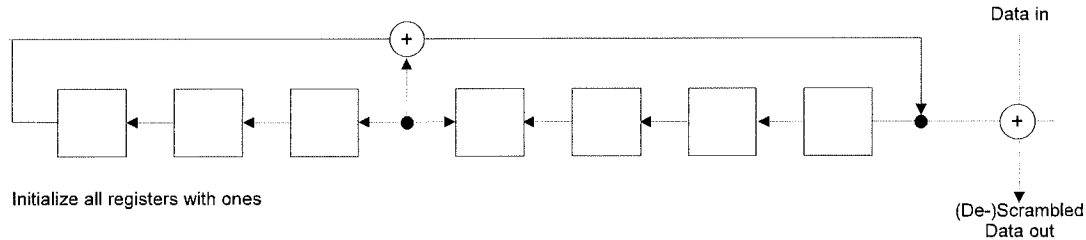


Figure 71—Frame synchronous scrambler/descrambler

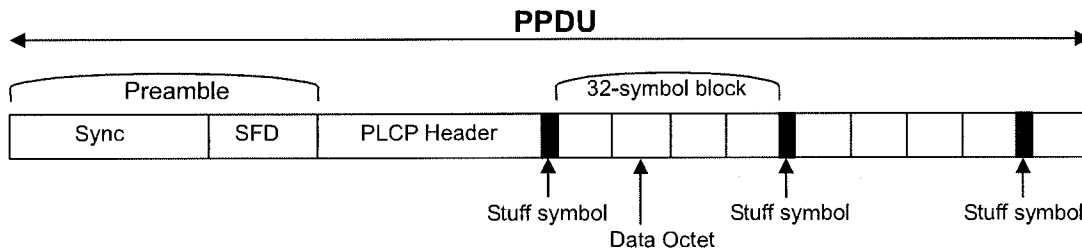


Figure 72—PLCP data whitener format

14.3.3 PLCP state machines

The PLCP consists of three state machines, as illustrated in the overview diagram of Figure 73: the transmit (TX), carrier sense/clear channel assessment (CS/CCA), and receive (RX) state machines. The three PLCP state machines are defined in the subclauses below; Figure 73 is not a state diagram itself. Execution of the PLCP state machines normally is initiated by the FH PLME state machine and begins at the CS/CCA state machine. The PLCP returns to the FH PLME state machine upon interrupt to service a PLME service request, such as PLME-SET, PLME-RESET, etc.

14.3.3.1 PLCP transmit procedure

The PLCP transmit procedure is invoked by the CS/CCA procedure immediately upon receiving a PHY-TXSTART.request(TXVECTOR) from the MAC sublayer. The CSMA/CA protocol is performed by the MAC with the PHY PLCP in the CS/CCA procedure prior to executing the transmit procedure.

14.3.3.1.1 Transmit state machine

The PLCP transmit state machine illustrated in Figure 74 includes functions that must be performed prior to, during, and after PPDU data transmission. Upon entering the transmit procedure in response to a PHY-TXSTART.request (TXVECTOR) from the MAC, the PLCP shall switch the PHY PMD circuitry from receive to transmit state; ramp on the transmit power amplifier in the manner prescribed in 14.6; and transmit the preamble sync pattern and SFD. The PLCP shall generate the PLCP Header as defined in 14.3.2.2 in

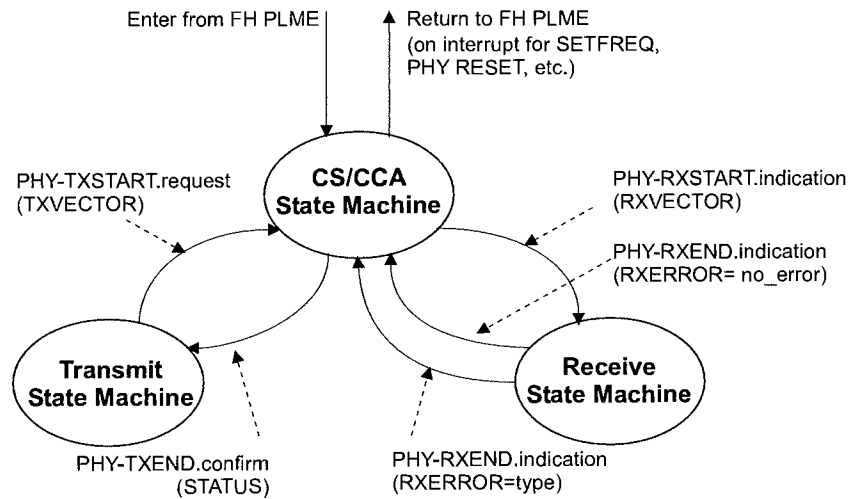


Figure 73—PLCP top-level state diagram

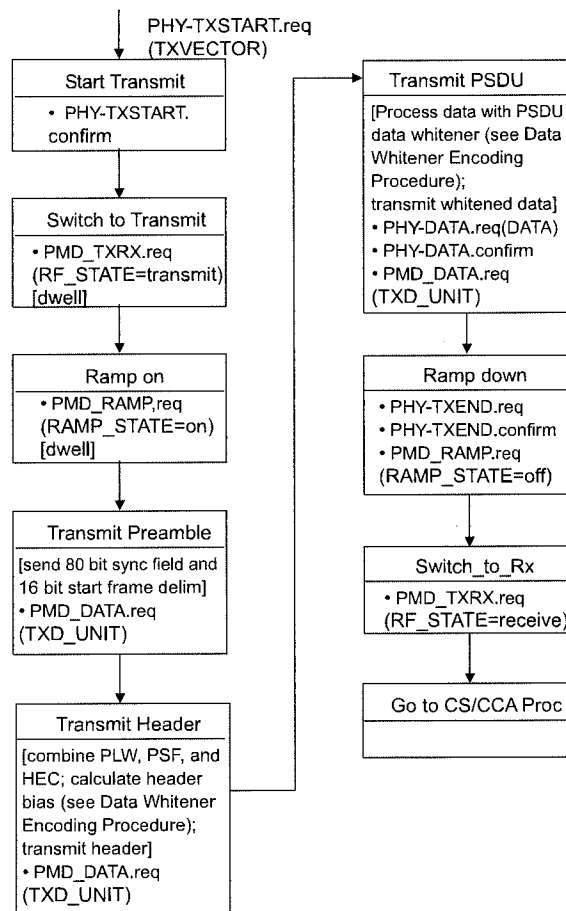


Figure 74—Transmit state machine

sufficient time to send the bits at their designated bit slot time. The PLCP shall add the PLCP Header to the start of the PSDU data.

Prior to transmitting the first PSDU data bit, the PLCP shall send a PHY-TXSTART.confirm message to the MAC indicating that the PLCP is ready to receive an MPDU data octet. The MAC will pass an MPDU data octet to the PHY with a PHY-DATA.request(DATA), which the PHY will respond to with a PHY-DATA.confirm. This sequence of PHY-DATA.request(DATA) and PHY-DATA.confirm shall be executed until the last data octet is passed to the PLCP. During transmission of the PSDU data, each bit of the PSDU shall be processed by the data whitener algorithm defined in Figure 75 and described in 14.3.2.3. Each PSDU data octet is processed and transmitted lsb first and msb last.

```

Data whitener encoding algorithm:

/*      If msb of stuff symbol = 1 then the next block is inverted; 0 = not inverted */
/*      Accumulate PLCP Header; begin stuffing on first bit of the PSDU */

/***** Calculate number of 32-symbol BSE blocks required to send PSDU;
          no padding is necessary when the number of symbols is not a multiple of 32 *****/
Input parameter: number_of_PSDU_octets, rate;          /* rate is 1 or 2*/
number_of_symbols= (number_of_PSDU_octets *8) /rate;
number_of_blocks_in_packet = truncate((number_of_symbols + 31) / 32);

/***** Accumulate the bias in the header to use in calculating the inversion state of the first
          block of PSDU data *****/
Read in header {b(1),...,b(32)};          /* b(1) is first bit in */
header_bias = Sum{weight(b(1)),...,weight(b(32))};
/* calculate bias in header; weights are defined in Table 31*/
Transmit {b(1),...,b(32)};          /* no stuffing on header */
accum=header_bias;          /* initialize accum */
Initialize scrambler to all ones;

/***** Whiten the PSDU data with scrambler and BSE encoder *****/
For n = 1 to number_of_blocks_in_packet
{
    b(0) = 0 for 1 Mbit/s; b(0)=00 for 2 Mbit/s;          /* b(0) is the stuff symbol */
    N = min(32, number_of_symbols);          /* N= block size in symbols */
    Read in next symbol block {b(1),...,b(N)};          /* b(n) = {0,1} or {0,1,2,3};
    1 - 8 octets, use PHY-DATA.req(DATA), PHY-DATA.confirm for each octet*/
    Scramble {b(1),...,b(N)};          /* see 14.3.2.3*/
    bias_next_block = Sum{weight(b(0)),...,weight(b(N))};          /* calculate bias with b(0)=0 */

    /**** if accum and bias of next block has the same sign, then invert block;
           if accum=0 or bias_next_block=0, don't invert *****/
    If {[accum * bias_next_block > 0] then
    {
        Invert {b(0),...,b(N)};          /* Invert deviation, or, negate msb of symbol */
        bias_next_block = - bias_next_block;
    }

    accum = accum + bias_next_block;
    transmit {b(0),...,b(N)};          /* b(0) is first symbol out */
    number_of_symbols = number_of_symbols - N
}
}

```

Figure 75—Data whitener encoding procedure

After the last MPDU octet is passed to the PLCP, the MAC will indicate the end of the frame with a *PHY-TXEND.request*. After the last bit of the PSDU data has completed propagation through the radio and been transmitted on the air, the PLCP shall complete the transmit procedure by sending a *PHY-TXEND.confirm* to the MAC sublayer, ramp off the power amplifier in the manner prescribed in 14.6, and switch the PHY PMD circuitry from transmit to receive state. The execution shall then return to the CS/CCA procedure.

The weights assigned to each value of the symbols are defined in Table 31 for the 1 Mbit/s (2GFSK) and 2 Mbit/s (4GFSK) symbols.

Table 31—PLCP field bit descriptions

2GFSK	4GFSK	Weight
—	10	3
1	—	2
—	11	1
Center	Center	0
—	01	-1
0	—	-2
—	00	-3

14.3.3.1.2 Transmit state timing

The transmit timing illustrated in Figure 76 is defined from the instant that the *PHY-TXSTART.request(TXVECTOR)* is received from the MAC sublayer. The PLCP shall switch the PMD circuitry from receive to transmit, turn on and settle the transmitter, and begin transmitting the first bit of the preamble at the antenna within a maximum of 20 μ s of receiving the *PHY-TXSTART.request(TXVECTOR)*. The PLCP Preamble shall be transmitted at 1 Mbit/s and be completed in 96 μ s. The PLCP Header shall be transmitted at 1 Mbit/s and be completed in 32 μ s. The variable length PSDU shall be transmitted at the selected data rate. After the last bit of the PSDU data has completed propagation through the radio and been transmitted on the air, the PLCP shall send the *PHY-TXEND.confirm* to the MAC sublayer. The PLCP shall turn off the transmitter, reducing the output energy to less than the specified off-mode transmit power within the time specified in 14.6. At the end of the power amplifier ramp down period, the PLCP shall switch the PMD circuitry from transmit to receive.

14.3.3.2 Carrier sense/clear channel assessment (CS/CCA) procedure

The PLCP CS/CCA procedure is executed while the receiver is turned on and the STA is not currently receiving or transmitting a packet. The CS/CCA procedure is used for two purposes: to detect the start of a network signal that can be received (CS) and to determine whether the channel is clear prior to transmitting a packet (CCA).

14.3.3.2.1 CS/CCA state machine

Timing for priority (PIFS, DIFS), contention backoff (slot times), and CS/CCA assessment windows is defined relative to the end of the last bit of the last packet on the air. The CS/CCA state machine is shown in Figure 77. The PLCP shall perform a CS/CCA assessment on a minimum of one antenna within a MAC contention back-off slot time of 50 μ s. The PLCP shall be capable of detecting within the slot time an FH PHY conformant signal that is received at the selected antenna up to 22 μ s after the start of the slot time with the synchronous detection performance specified in 14.6.15.3. Subclause 14.6.15.3 specifies detection performance with zero-one sync patterns and with random data patterns. If a start of a transmission is asynchronous with the BSS and arrives after the start of the slot but at least 16 μ s prior to the end of the slot, the PLCP shall indicate a busy channel prior to the end of the slot time with the asynchronous detection performance specified in 14.6.15.3.

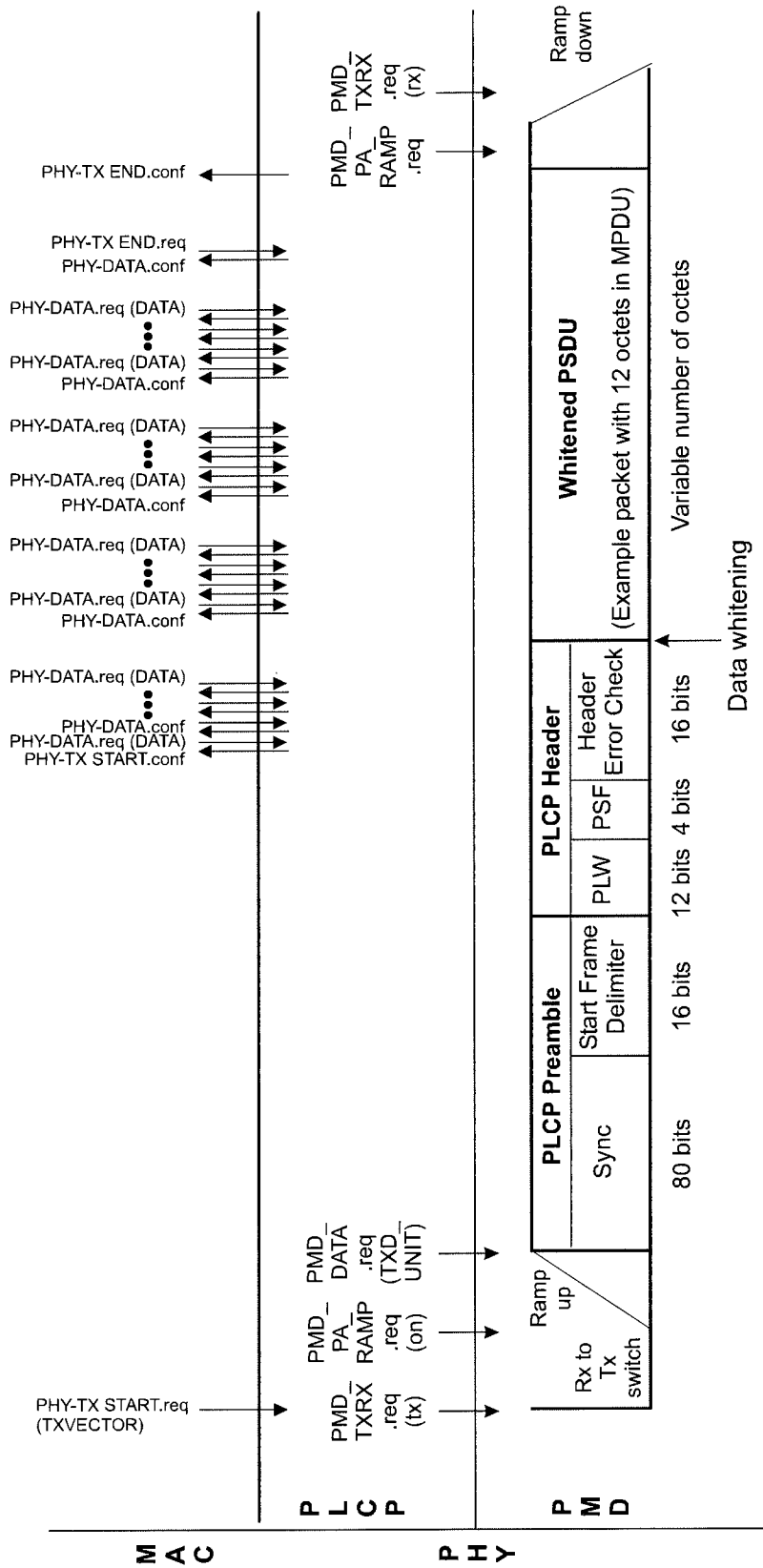


Figure 76—Transmit state timing

The CCA indication immediately prior to transmission shall be performed on an antenna with essentially the same free space gain and gain pattern as the antenna to be used for transmission. The method of determining CS/CCA is unspecified except for the detection performance of a conformant method as specified in 14.6.15.3.

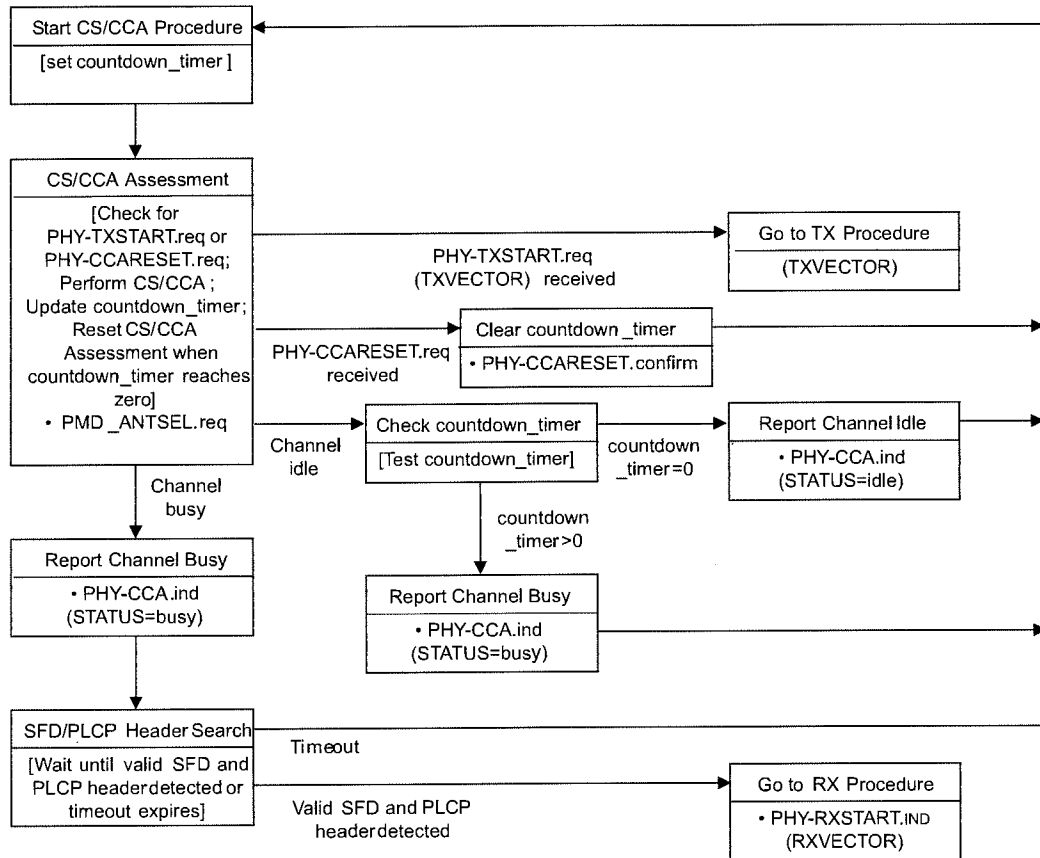


Figure 77—CS/CCA state machine

If a *PHY-TXSTART.request* (*TXVECTOR*) is received, the CS/CCA procedure shall exit to the transmit procedure within 1 μ s. If a *PHY-CCARESET.request* is received, the PLCP shall reset the CS/CCA state machine to the state appropriate for the end of a complete received frame. This service primitive is generated by the MAC at the end of a NAV period. The PHY shall indicate completion of the request by sending a *PHY-CCARESET.confirm* to the MAC.

If a CS/CCA assessment returns a channel idle result, the PHY shall send a *PHY-CCA.indicate(STATUS=idle)* to the MAC.

If a CS/CCA assessment returns a channel busy result, the PHY shall send a *PHY-CCA.indicate(STATUS=busy)* to the MAC. Upon a channel busy assessment, the PLCP shall stop any antenna switching prior to the earliest possible arrival time of the SFD and detect a valid SFD and PLCP Header if received. A valid PLCP Header is defined as containing valid PLCP Length Word and PHY Signaling field values and a valid HEC field. If a valid SFD/PLCP Header is detected, the CS/CCA procedure shall send a *PHY-RXSTART.indicate(RXVECTOR)* message to the MAC sublayer and exit to the receive procedure. The PLCP shall dwell and search for the SFD/PLCP Header for a minimum period longer than the latest possible arrival time of the SFD/PLCP Header. Indication of a busy channel does not necessarily lead to the successful reception of a frame.

The octet/bit count remaining may be a nonzero value when returning from the receive procedure if a signal in the process of being received was lost prior to the end as determined from the Length field of a valid PLCP Header. The countdown timer shall be set to the octet/bit count and used to force the CS/CCA indication to remain in the BUSY state until the predicted end of the frame regardless of actual CS/CCA indications.

However, if the CS/CCA procedure indicates the start of a new frame within the countdown timer period, it is possible to transition to the receive procedure prior to the end of the countdown timer period. If the PHY transitions to receive under these conditions, the countdown timer shall be reset to the longer of (1) the remaining time of the current frame or (2) the length of the new frame.

When a nonzero countdown timer reaches zero, the PLCP shall reset the CS/CCA state machine to the state appropriate for the end of a complete received frame and the CS/CCA indication shall reflect the state of the channel.

If the receive procedure encountered an unsupported rate error, the PLCP shall keep the CS/CCA state at Busy for the duration of the frame by setting the countdown timer to the value corresponding to the calculated time based on the information in the PLCP Header and the 33/32 expansion factor.

14.3.3.2.2 CS/CCA state timing

Timing for priority (PIFS, DIFS), contention backoff (slot times), and CS/CCA assessment windows is defined relative to the end of the last bit of the last packet on the air. The PLCP shall perform a CS/CCA assessment on a minimum of one antenna within a slot time. The appropriate CS/CCA indication shall be available prior to the end of each 50 μ s slot time with the performance specified in 14.6. See Figure 78.

If a STA has not successfully received the previous packet, the perceived packet end time and slot boundary times will have a higher uncertainty for that STA.

14.3.3.3 PLCP receive procedure

The PLCP receive procedure is invoked by the PLCP CS/CCA procedure upon detecting a portion of the preamble sync pattern followed by a valid SFD and PLCP Header.

14.3.3.3.1 Receive state machine

The PLCP receive procedure shown in Figure 79 includes functions that must be performed while the PPDU is being received. The PLCP receive procedure begins upon detection of a valid SFD and PLCP Header in the CS/CCA procedure. The PLCP shall set a PPDU octet/bit counter to indicate the last bit of the packet, receive the PPDU bits, and perform the data whitening decoding procedure shown in Figure 80 on each PPDU bit. The PLCP shall pass correctly received data octets to the MAC with a series of *PHY-DATA.indicate(DATA)*. After the last PPDU bit is received and the last octet is passed to the MAC, the PLCP shall send a *PHY-RXEND.indicate(RXERROR=no_error)* to the MAC sublayer. Upon error-free completion of a packet reception, the PLCP shall exit the receive procedure and return to the PLCP CS/CCA procedure with the octet/bit count set to 0.

If the PLCP Header was decoded without a CRC error but encountered an unsupported rate, then the PLCP shall immediately complete the receive procedure with a *PHY-RXEND.indicate(RXERROR=unsupported_rate)* to the MAC, and return to the CS/CCA procedure with the octet/bit count remaining and the data rate value contained in the PLCP Header.

If an error was detected during the reception of the PPDU, the PLCP shall immediately complete the receive procedure with a *PHY-RXEND.indicate(RXERROR=carrier_lost)* to the MAC, and return to the CS/CCA procedure with the octet/bit count remaining and the data rate value contained in the PLCP Header.

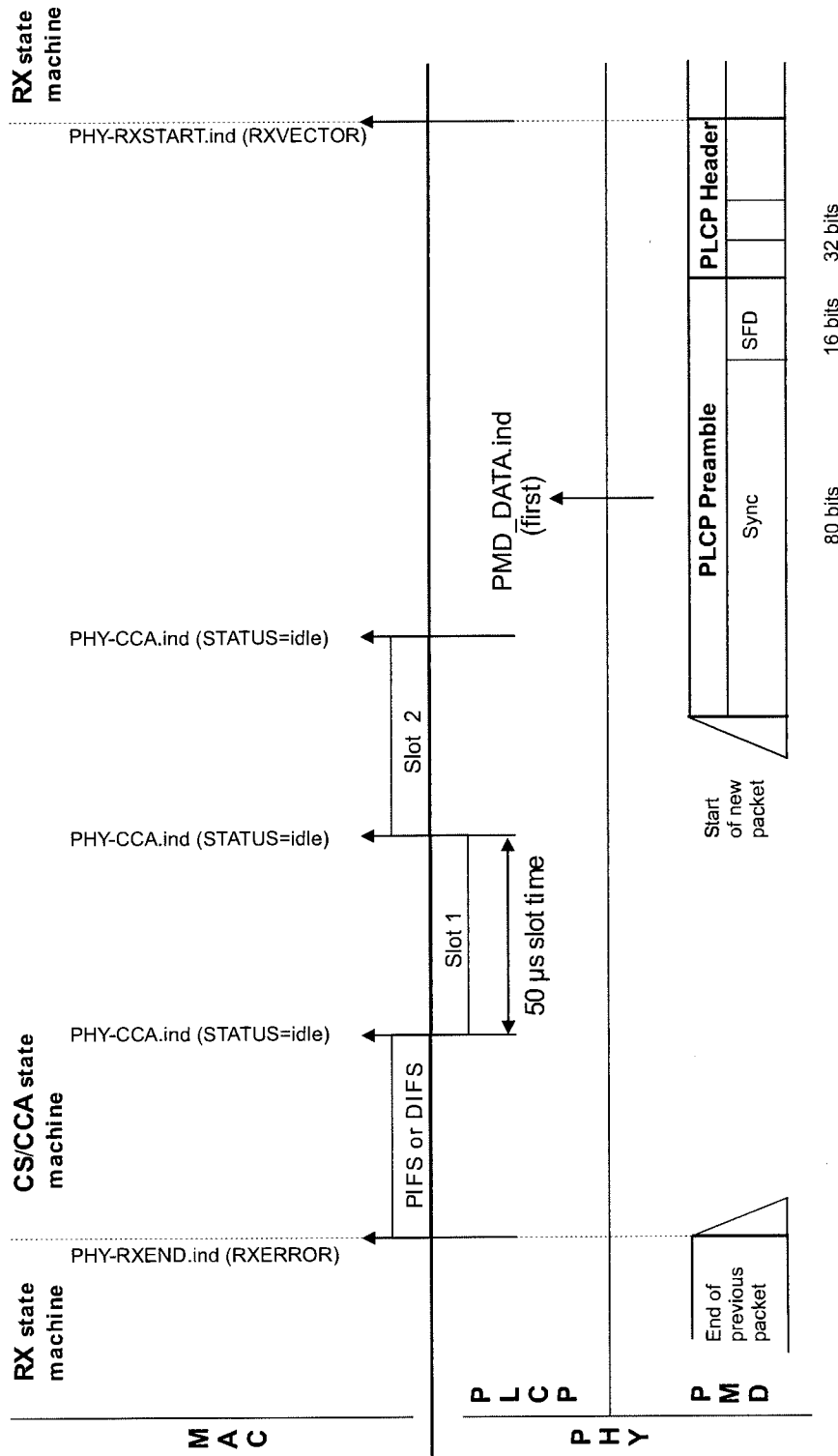


Figure 78—CS/CCA state timing

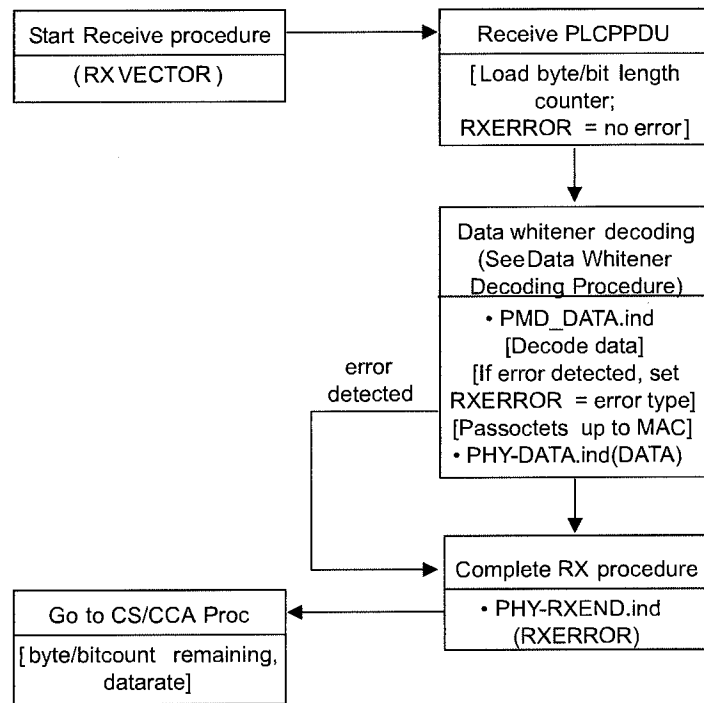


Figure 79—Receive state machine

Data whitener decoding algorithm:

```

/* If msb of stuff symbol = 1 then the next block is inverted; 0 = not inverted */
/* Stuffing begins on first symbol of PLCP Header following the SFD */
/* Algorithm begins after verifying validity of header with HEC */

/***** Read header *****/
Read in header {b(1),...,b(32)}; /* b(1) is first bit in */

Get number_of_PSDU_octets, rate from header; /* rate is 1 or 2 */
number_of_symbols = (number_of_PSDU_octets*8)/rate
number_of_blocks_in_packet = truncate((number_of_symbols + 31) / 32);
Initialize scrambler to all ones;

/***** De-whiten the PPDU data with BSE decoder and de-scrambler *****/
For n = 1 to number_of_blocks_in_packet
{
  N = min(32, # of symbols remaining); /* N= block size in symbols */
  Read in next block {b(0),...,b(N)}; /* b(n) = {0,1} or {0,1,2,3} */

  If {[msb of b(0)=1] then Invert {b(1),...,b(N)}; /* if invert bit=true */
  Descramble {b(1),...,b(N)}; /* see 14.3.2.3*/
  Send {b(1),...,b(N)} to MAC
  /* 1 - 8 octets; use PHY-DATA.ind(DATA) for each octet. */
}
  
```

Figure 80—Data whitener decoding procedure

14.3.3.3.2 Receive state timing

The receive state timing shown in Figure 81 is defined to begin upon detection of a valid SFD and PLCP Header in the CS/CCA procedure. The PLCP shall begin receiving the variable length whitened PSDU immediately after the end of the last bit of the PLCP Header. The PLCP shall send a *PHY-RXEND.indicate(RXERROR)* after receiving the last PPDU data bit.

If any error was detected during the reception of the PPDU, the PLCP may send a *PHY-RXEND.indicate(RXERROR)* and terminate the receive procedure before the last bit arrives.

14.4 PLME SAP layer management

14.4.1 Overview

This subclause describes the services provided by the FHSS PLME to the upper layer management entities. The PLME/PMD services are defined in terms of service primitives. These primitives are abstract representations of the services and are not intended to restrict implementations.

14.4.2 FH PHY specific MAC sublayer management entity (MLME) procedures

14.4.2.1 Overview

The specific MAC sublayer management entity (MLME) procedures required for operating the FHSS PHY are specified in this portion of the subclause. The relationship between the MLME and FH PLME procedures is also described.

14.4.2.2 FH synchronization

The MLME of a compliant FH PHY STA shall perform the FH time synchronization procedure as defined in 11.1.5. This procedure provides for synchronized frequency hopping for all compliant FH PHY STAs within a single BSS or ad hoc network. The FH PLME accepts *PLME-SET.request* commands from the MLME to change the tune frequency at the time determined by the MLME. The tune frequency is changed by updating any combination of the Set, Pattern, and Index PHY MIB parameters.

14.4.3 FH PHY layer management entity state machines

14.4.3.1 Overview

This portion of this subclause describes the FH PHY layer management state machines to turn the PMD on/off, reset the PLCP state machine, and change the frequency hop channel.

14.4.3.2 PLME state machine

The PLME state machine in Figure 82 begins with a *PLME-SET.request (dot11CurrentPowerState= ON)*, which turns on the PHY circuitry, resets the PLME and PLCP state machines, and sends a *PLME-SET.confirm*. The MAC then sends a series of three *PLME-SET.request* primitives to update the *dot11CurrentSet*, *dot11CurrentPattern*, and *dot11CurrentIndex* PHY MIB parameters, which together tune the PMD to the selected channel. The PLME then transfers execution to the PLCP state machine as defined in 14.3.3.

Upon receiving a PLME request from a higher-level LME, the PLCP shall return execution to the PLME state machine and process the request. A *PLME-RESET.request* shall cause a reset to the PLME and PLCP state machines. A *PLME-SET.request* updating the *dot11CurrentIndex* or a combination of the *dot11CurrentSet*, *dot11CurrentPattern*, and *dot11CurrentIndex* shall cause the PLCP to terminate a receive

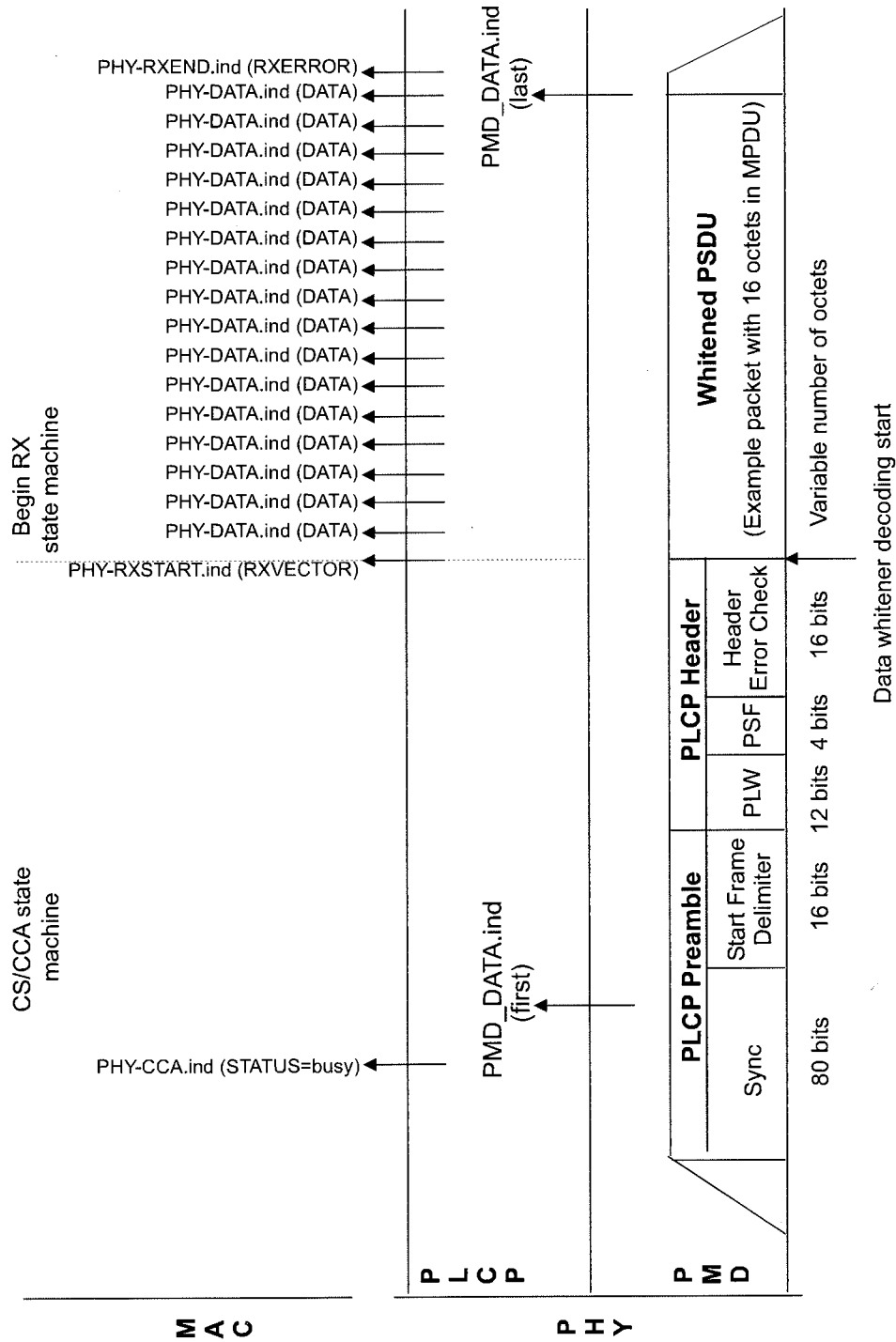


Figure 81—Receive timing

or CS/CCA process and change frequency before returning to the PLCP state machine. A *PLME-SET.request(dot11CurrentPowerState=OFF)* shall cause the PLCP to terminate a receive or CS/CCA process, power down the PMD circuitry, and return the PLME state machine to the idle state. *PLME-SET.requests* to any parameter other than the ones identified within this paragraph shall be executed and control shall be returned to the PLCP state machine. The MAC should not send a PLME request while the PLCP is in the transmit state.

All *PLME-GET.requests* shall be processed in parallel and with no interruption to the execution of any state machine in process.

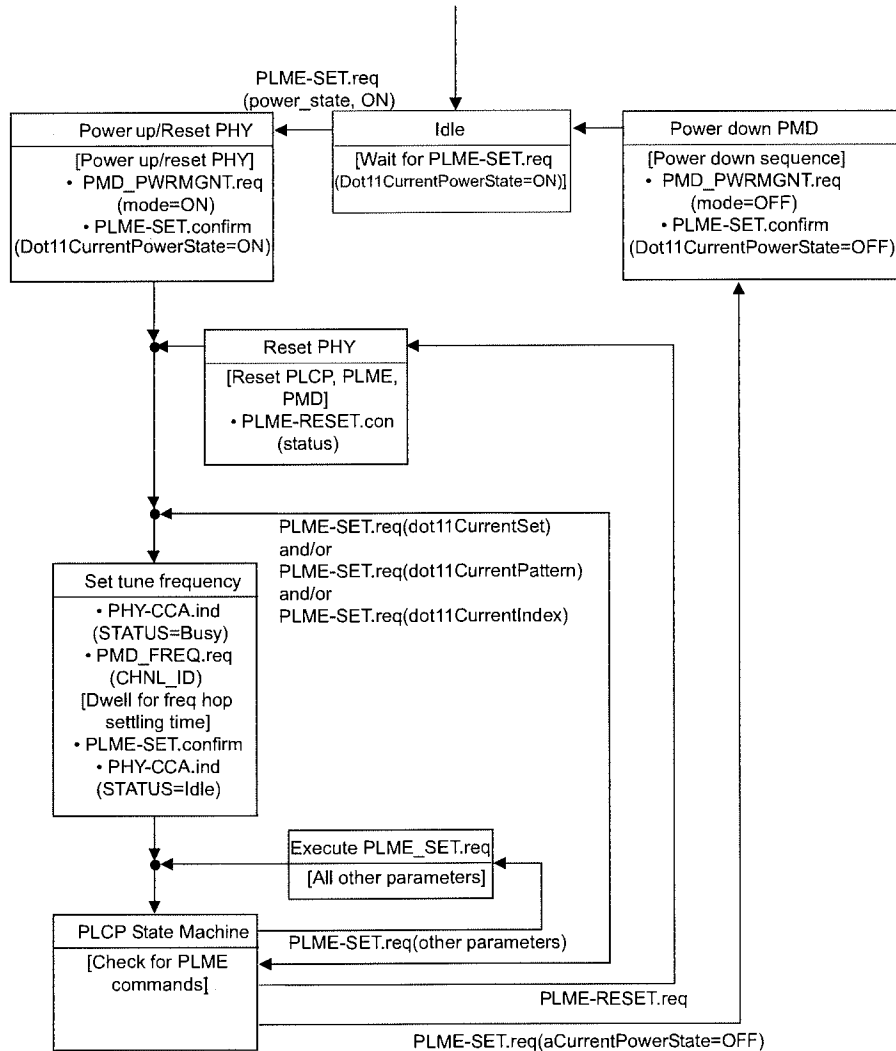


Figure 82—PLME state machine

14.4.3.3 PLME management primitives

The FH PLME uses the generic management primitives defined in 10.2 to manage all FH PHY parameters.

14.5 FHSS PMD sublayer services

14.5.1 Scope and field of application

The PMD services provided to the PLCP for the FHSS PHY are described in this subclause. Also defined in this subclause are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire FHSS PHY is shown in Figure 83.

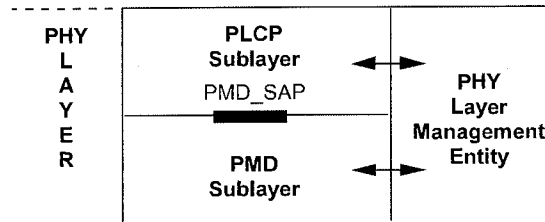


Figure 83—PMD layer reference model

14.5.2 Overview of services

In general, the FHSS PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which the signals required by these primitives are imposed onto the medium. In the FHSS PMD sublayer at the receiver the process is reversed. The combined function of the transmitting and receiving FHSS PMD sublayers results in a data stream, timing information, and receive parameter information being delivered to the receiving PLCP sublayer.

14.5.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the FHSS PMD sublayer fall into the following two basic categories:

- a) Service primitives that support PLCP peer-to-peer interactions;
- b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

14.5.4 Basic service and options

All of the service primitives described in this subclause are considered mandatory unless otherwise specified.

14.5.4.1 PMD_SAP peer-to-peer service primitives

Table 32 indicates the primitives for peer-to-peer interactions.

Table 32—PMD-SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_DATA	X	X	—	—

14.5.4.2 PMD_SAP sublayer-to-sublayer service primitives

Table 33 indicates the primitives for sublayer-to-sublayer interactions.

Table 33—PMD_SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXRX	X	—	—	—
PMD_PA_RAMP	X	—	—	—
PMD_ANTSEL	X	—	—	—
PMD_TXPWRLVL	X	—	—	—
PMD_FREQ	X	—	—	—
PMD_RSSI	—	X	—	—
PMD_PWRMGMT	X	—	—	—

14.5.4.3 PMD_SAP service primitives parameters

Table 34 shows the parameters used by one or more of the PMD_SAP service primitives.

Table 34—List of parameters for PMD primitives

Parameter	Associate primitive	Value
TXD_UNIT	PMD_DATA.request	1 Mbit/s: 0, 1 2 Mbit/s: 0, 1, 2, 3
RXD_UNIT	PMD_DATA.indicate	1 Mbit/s: 0, 1 2 Mbit/s: 0, 1, 2, 3
RF_STATE	PMD_TXRX.request	TRANSMIT, RECEIVE
RAMP_STATE	PMD_PA_RAMP.request	ON, OFF
ANTENNA_STATE	PMD_ANTSEL.request	1 to 255
TXPWR_LEVEL	PMD_TXPWRLVL.request	LEVEL1, LEVEL2, LEVEL3, LEVEL4
CHNL_ID	PMD_FREQ.request	2–80 inclusive
STRENGTH	PMD_RSSI.indicate	0 to RSSI Max
MODE	PMD_PWRMGMT.request	ON, OFF

14.5.5 PMD_SAP detailed service specification

This subclause describes the services provided by each PMD primitive.

14.5.5.1 PMD_DATA.request

14.5.5.1.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

14.5.5.1.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_DATA.request (TXD_UNIT)

The TXD_UNIT parameter can take on one of two values: one or zero. This parameter represents a single data bit. The effect of this parameter is that the PMD will properly modulate the medium to represent ones or zeros as defined in the FHSS PMD modulation specifications for a given data rate.

14.5.5.1.3 When generated

This primitive is generated by the PLCP sublayer to request the transmission of a single data bit on the PMD sublayer. The bit clock is assumed to be resident or part of the PLCP and this primitive is issued at every clock edge once the PLCP has begun transmitting data.

14.5.5.1.4 Effect of receipt

The receipt of this primitive will cause the PMD entity to encode and transmit a single data bit.

14.5.5.2 PMD_DATA.indicate

14.5.5.2.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

14.5.5.2.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_DATA.indicate (RXD_UNIT)

The RXD_UNIT parameter can take on one of two values: one or zero. This parameter represents the current state of the medium as determined by the FHSS PMD modulation specifications for a given data rate.

14.5.5.2.3 When generated

The PMD_DATA.indicate is generated to all receiving PLCP entities in the network after a PMD_DATA.request is issued.

14.5.5.2.4 Effect of receipt

The effect of receipt of this primitive by the PLCP is unspecified in this standard.

14.5.5.3 PMD_TXRX.request

14.5.5.3.1 Function

This primitive is used to place the PMD entity into the transmit or receive function.

14.5.5.3.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_TXRX.request (RF_STATE)

The RF_STATE parameter can take on one of two values: TRANSMIT or RECEIVE. When the value of the primitive is TRANSMIT, the RF state of the radio is transmit. If the value of the primitive is RECEIVE, the RF state of the radio is receive.

14.5.5.3.3 When generated

This primitive is generated whenever the mode of the radio needs to be set or when changing from transmit to receive or receive to transmit.

14.5.5.3.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the mode of the radio to be in either transmit or receive.

14.5.5.4 PMD_PA_RAMP.request

14.5.5.4.1 Function

This primitive defines the start of the ramp up or ramp down of the radio transmitter's power amplifier.

14.5.5.4.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_PA_RAMP.request (RAMP_STATE)

The RAMP_STATE parameter can take on one of two values: ON or OFF. When the value of the primitive is ON, the state of the transmit power amplifier is "on." If the value of the primitive is OFF, the state of the transmit power amplifier is "off."

14.5.5.4.3 When generated

This primitive is issued only during transmit and to establish the initial state. It is generated by the PLCP at the start of the transmit function to turn the transmitter's power amplifier "on." A power amplifier ramp-up period follows the change of state from "off" to "on." After the PLCP has transferred all required data to the PMD entity, this primitive again will be issued by the PLCP to place the transmit power amplifier back into the "off" state. A power amplifier ramp-down period follows the change of state from "on" to "off."

14.5.5.4.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the transmit power amplifier to turn on or off.

14.5.5.5 PMD_ANTSEL.request

14.5.5.5.1 Function

This primitive is used to select which antenna the PMD entity will use to transmit or receive data.

14.5.5.5.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_ANTSEL.request (ANTENNA_STATE)

The ANTENNA_STATE parameter can take on values from one to N (where N is the number of antennas supported). When the value of the primitive is a ONE, the PMD will switch to antenna 1 for receive or transmit; if the value of the primitive is TWO, the PMD entity will switch to antenna 2 for receive or transmit, etc.

14.5.5.5.3 When generated

This primitive is generated at various times by the PLCP entity to select an antenna. During receive, this primitive can be used to manage antenna diversity. During transmit, this primitive can be used to select a transmit antenna. This primitive will also be used during CCA.

14.5.5.5.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the radio to select the antenna specified.

14.5.5.6 PMD_TXPWRLVL.request**14.5.5.6.1 Function**

This primitive defines the power level the PMD entity will use to transmit data.

14.5.5.6.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_TXPWRLVL.request (TXPOWER_LEVEL)

The TXPOWER_LEVEL parameter can be one of the values listed in Table 35.

Table 35—Transmit power levels

TXPWR_LEVEL	Level description
LEVEL1	Defined as TxPowerLevel1 in MIB
LEVEL2	Defined as TxPowerLevel2 in MIB
LEVEL3	Defined as TxPowerLevel3 in MIB
LEVEL4	Defined as TxPowerLevel4 in MIB
LEVEL5	Defined as TxPowerLevel5 in MIB
LEVEL6	Defined as TxPowerLevel6 in MIB
LEVEL7	Defined as TxPowerLevel7 in MIB
LEVEL8	Defined as TxPowerLevel8 in MIB

14.5.5.6.3 When generated

This primitive is generated as part of the transmit sequence.

14.5.5.6.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the transmit power level to be modify.

14.5.5.7 PMD_FREQ.request**14.5.5.7.1 Function**

This primitive defines the frequency the PMD entity will use to receive or transmit data. Since changing the radio frequency is not an immediate function, this primitive serves also as an indication of the start of this process. The completion of this process is dictated by other PMD specifications.

14.5.5.7.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_FREQ.request (CHANNEL_ID)

The CHANNEL_ID parameter can be one of the values listed in Table 38, Table 39, Table 40, or Table 41.

14.5.5.7.3 When generated

This primitive is generated by the PLCP whenever a change to a new frequency is required.

14.5.5.7.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the radio to change to a new frequency defined by the value of the CHNL_ID.

14.5.5.8 PMD_RSSI.indicate**14.5.5.8.1 Function**

This primitive transfers a receiver signal strength indication of the physical medium from the PMD sublayer to the PLCP sublayer. This value will be used by the PLCP to perform any diversity or clear channel assessment functions required by the PLCP or other sublayers.

14.5.5.8.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_RSSI.indicate (STRENGTH)

The STRENGTH parameter can be a value from 0 to 15. This parameter is an indication by the PMD sublayer of the magnitude of the energy observed at the selected antenna. This reported value is used to generate the RSSI term in the PHY-RXSTART.ind(RXVECTOR) primitive and might also be used by any diversity function. Since RSSI is only used in a relative manner by the MAC sublayer, this parameter is defined to have no more than 16 values, ranging from 0 through RSSI_Max. The value zero is the weakest signal strength, while RSSI_Max is the strongest signal strength.

14.5.5.8.3 When generated

This primitive is generated continually by the PMD entity to transfer a receive signal strength indication to the PLCP.

14.5.5.8.4 Effect of receipt

The effect of receipt of this primitive by the PLCP is unspecified in this standard.

14.5.5.9 PMD_PWRMGMT.request

14.5.5.9.1 Function

This primitive is used by the higher-layer entities to manage or control the power consumption of the PMD when not in use. This allows higher-layer entities to put the radio into a sleep or standby mode when receipt or sending of any data is not expected.

14.5.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_PWRMGMT.request (MODE)

The MODE parameter can have one of two values: ON or OFF. When the value of the parameter is ON, the PMD entity will enter into a fully functional mode that allows it to send or receive data. When the value of the parameter is OFF, the PMD entity will place itself in a standby or power-saving mode. In the low-power mode, the PMD entity is not expected to be able to perform any request by the PLCP, nor is it expected to indicate any change in PMD state or status.

14.5.5.9.3 When generated

This primitive is delivered by the PLCP but actually is generated by a higher-layer management entity.

14.5.5.9.4 Effect of receipt

Upon receipt of this primitive, the PMD entity will enter a fully functional or low power consumption state depending on the value of the primitive's parameter.

14.6 FHSS PMD sublayer, 1.0 Mbit/s

14.6.1 1 Mbit/s PMD operating specifications, general

In general, the PMD accepts convergence layer service primitives and provides the actual means by which the signals required by these primitives are imposed on the medium. In the PMD sublayer at the receiver, the process is reversed. The combined function of the transmitting and receiving PMD sublayers results in a data stream, timing information, and receive parameter information being delivered to the receiving convergence sublayer.

14.6.2 Regulatory requirements

Wireless LANs implemented in accordance with this standard are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations for Europe, Japan, and North America at the time this standard was issued. These regulations are subject to revision, or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed within this standard. Implementors are referred to the following regulatory sources for further information.

Operation in countries within Europe, or other regions outside Japan or North America, may be subject to additional or alternative national regulations.

The documents listed below specify the current regulatory requirements for various geographic areas at the time this standard was developed. They are provided for information only, and are subject to change or revision at any time.

Geographic area	Approval standards	Documents	Approval authority
Europe	European Telecommunications Standards Institute (ETSI)	ETS 300-328, ETS 300-339	National type approval authorities
France	Règle technique applicable aux équipements radioélectriques de transmission de données à large bande fonctionnant dans la bande de fréquences à 2,4 GHz et utilisant la technique de l'étalement de spectre (Édition février 1995)	SP/DGPT/ATAS/23, ETS 300-328, ETS 300-339	Direction Générale des Postes et Télécommunications (DGPT)
Japan	Association of Radio Industries and Businesses (ARIB)	RCR STD-33A	Ministry of Telecommunications (MKK)
North America Canada USA	Industry Canada (IC) Federal Communications Commission (FCC)	GL36 CFR47, Part 15, Sections 15.205, 15.209, 15.247	IC FCC
Spain	Suplemento Del Numero 164 Del Boletín Oficial Del Estado (Published 10 July 1991, Revised 25 June 1993)	ETS 300-328, ETS 300-339	Cuadro Nacional De Atribucion De Frecuencias

14.6.3 Operating frequency range

A conformant PMD implementation shall be able to select the carrier frequency (F_c) from the full geographic-specific set of available carrier frequencies. Table 36 summarizes these frequencies for a number of geographic locations.

Table 36—Operating frequency range

Lower Limit	Upper limit	Regulatory range	Geography
2.402 GHz	2.480 GHz	2.400–2.4835 GHz	North America
2.402 GHz	2.480 GHz	2.400–2.4835 GHz	Europe ^a
2.473 GHz	2.495 GHz	2.471–2.497 GHz	Japan
2.447 GHz	2.473 GHz	2.445–2.475 GHz	Spain
2.448 GHz	2.482 GHz	2.4465–2.4835 GHz	France

NOTE—The frequency ranges in this table are subject to the geographic-specific regulatory authorities.

^aExcluding Spain and France.

14.6.4 Number of operating channels

The number of transmit and receive frequency channels used for operating the PMD entity is 79 for the USA and Europe, and 23 for Japan. Table 37 summarizes these frequencies for a number of geographic locations. This is more fully defined in Table 38 through Table 41.

Table 37—Number of operating channels

Minimum	Hopping set	Geography
75	79	North America
20	79	Europe ^a
Not applicable	23	Japan
20	27	Spain
20	35	France

NOTE—The number of required hopping channels is subject to the geographic-specific regulatory authorities.

^aExcluding Spain and France.

14.6.5 Operating channel center frequency

The channel center frequency is defined in sequential 1.0 MHz steps beginning with the first channel, channel 2.402 GHz for the USA and Europe excluding Spain and France, as listed in Table 38. The channel centers for Japan, starting at 2.473 GHz with 1 MHz increments, are listed in Table 39. The channel centers for Spain and France are listed in Table 40 and Table 41, respectively.

Table 38—Requirements in North America and Europe (excluding Spain and France; values specified in GHz)

Channel #	Value	Channel #	Value	Channel #	Value
2	2.402	28	2.428	54	2.454
3	2.403	29	2.429	55	2.455
4	2.404	30	2.430	56	2.456
5	2.405	31	2.431	57	2.457
6	2.406	32	2.432	58	2.458
7	2.407	33	2.433	59	2.459
8	2.408	34	2.434	60	2.460
9	2.409	35	2.435	61	2.461
10	2.410	36	2.436	62	2.462
11	2.411	37	2.437	63	2.463
12	2.412	38	2.438	64	2.464

**Table 38—Requirements in North America and Europe
(excluding Spain and France; values specified in GHz) (continued)**

Channel #	Value	Channel #	Value	Channel #	Value
13	2.413	39	2.439	65	2.465
14	2.414	40	2.440	66	2.466
15	2.415	41	2.441	67	2.467
16	2.416	42	2.442	68	2.468
17	2.417	43	2.443	69	2.469
18	2.418	44	2.444	70	2.470
19	2.419	45	2.445	71	2.471
20	2.420	46	2.446	72	2.472
21	2.421	47	2.447	73	2.473
22	2.422	48	2.448	74	2.474
23	2.423	49	2.449	75	2.475
24	2.424	50	2.450	76	2.476
25	2.425	51	2.451	77	2.477
26	2.426	52	2.452	78	2.478
27	2.427	53	2.453	79	2.479
—	—	—	—	80	2.480

**Table 39—Requirements in Japan
(values specified in GHz)**

Channel #	Value	Channel #	Value	Channel #	Value
73	2.473	81	2.481	89	2.489
74	2.474	82	2.482	90	2.490
75	2.475	83	2.483	91	2.491
76	2.476	84	2.484	92	2.492
77	2.477	85	2.485	93	2.493
78	2.478	86	2.486	94	2.494
79	2.479	87	2.487	95	2.495
80	2.480	88	2.488	—	—

**Table 40—Requirements in Spain
(values specified in GHz)**

Channel #	Value	Channel #	Value	Channel #	Value
47	2.447	56	2.456	65	2.465
48	2.448	57	2.457	66	2.466
49	2.449	58	2.458	67	2.467
50	2.450	59	2.459	68	2.468
51	2.451	60	2.460	69	2.469
52	2.452	61	2.461	70	2.470
53	2.453	62	2.462	71	2.471
54	2.454	63	2.463	72	2.472
55	2.455	64	2.464	73	2.473

**Table 41—Requirements in France
(values specified in GHz)**

Channel #	Value	Channel #	Value	Channel #	Value
48	2.448	60	2.460	72	2.472
49	2.449	61	2.461	73	2.473
50	2.450	62	2.462	74	2.474
51	2.451	63	2.463	75	2.475
52	2.452	64	2.464	76	2.476
53	2.453	65	2.465	77	2.477
54	2.454	66	2.466	78	2.478
55	2.455	67	2.467	79	2.479
56	2.456	68	2.468	80	2.480
57	2.457	69	2.469	81	2.481
58	2.458	70	2.470	82	2.482
59	2.459	71	2.471	—	—

14.6.6 Occupied channel bandwidth

Occupied channel bandwidth shall meet all applicable local geographic regulations for 1 MHz channel spacing. The rate at which the PMD entity will hop is governed by the MAC. The hop rate is an attribute with a maximum dwell time subject to local geographic regulations.

14.6.7 Minimum hop rate

The minimum hop rate shall be governed by the regulatory authorities.

14.6.8 Hop sequences

The hopping sequence of an individual PMD entity is used to create a pseudorandom hopping pattern utilizing uniformly the designated frequency band. Sets of hopping sequences are used to co-locate multiple PMD entities in similar networks in the same geographic area and to enhance the overall efficiency and throughput capacity of each individual network.

An FH pattern, F_x , consists of a permutation of all frequency channels defined in Table 38 and Table 39. For a given pattern number, x , the hopping sequence can be written as follows:

$$F_x = \{f_x(1), f_x(2), \dots, f_x(p)\} \tag{1}$$

where

- $f_x(i)$ is the channel number (as defined in 14.6.4) for i^{th} frequency in x^{th} hopping pattern;
- p is the number of frequency channels in hopping pattern (79 for North America and most of Europe, 23 for Japan, 35 for France, 27 for Spain).

Given the hopping pattern number, x , and the index for the next frequency, i (in the range 1 to p), the channel number shall be defined to be as follows:

- $f_x(i) = [b(i) + x] \bmod (79) + 2$ in North America and most of Europe, with $b(i)$ defined in Table 42.
- $= [(i - 1) \times x] \bmod (23) + 73$ in Japan.
- $= [b(i) + x] \bmod (27) + 47$ in Spain with $b(i)$ defined in Table 43.
- $= [b(i) + x] \bmod (35) + 48$ in France with $b(i)$ defined in Table 44.

Table 42—Base-Hopping sequence $b(i)$ for North America and most of Europe

i	$b(i)$	i	$b(i)$	i	$b(i)$	i	$b(i)$	i	$b(i)$	i	$b(i)$	i	$b(i)$	i	$b(i)$
1	0	11	76	21	18	31	34	41	14	51	20	61	48	71	55
2	23	12	29	22	11	32	66	42	57	52	73	62	15	72	35
3	62	13	59	23	36	33	7	43	41	53	64	63	5	73	53
4	8	14	22	24	71	34	68	44	74	54	39	64	17	74	24
5	43	15	52	25	54	35	75	45	32	55	13	65	6	75	44
6	16	16	63	26	69	36	4	46	70	56	33	66	67	76	51
7	71	17	26	27	21	37	60	47	9	57	65	67	49	77	38
8	47	18	77	28	3	38	27	48	58	58	50	68	40	78	30
9	19	19	31	29	37	39	12	49	78	59	56	69	1	79	46
10	61	20	2	30	10	40	25	50	45	60	42	70	28	—	—

Table 43—Base-Hopping sequence $b(i)$ for Spain

i	$b(i)$	i	$b(i)$	i	$b(i)$
1	13	10	19	19	14
2	4	11	8	20	1
3	24	12	23	21	20
4	18	13	15	22	7
5	5	14	22	23	16
6	12	15	9	24	2
7	3	16	21	25	11
8	10	17	0	26	17
9	25	18	6	27	26

Table 44—Base-Hopping sequence $b(i)$ for France

i	$b(i)$	i	$b(i)$	i	$b(i)$
1	17	13	31	25	15
2	5	14	20	26	3
3	18	15	29	27	11
4	32	16	22	28	30
5	23	17	12	29	24
6	7	18	6	30	9
7	16	19	28	31	27
8	4	20	14	32	19
9	13	21	25	33	2
10	33	22	0	34	21
11	26	23	8	35	34
12	10	24	1	—	—

The sequences are designed to ensure some minimum distance in frequency between contiguous hops. The minimum hop size is 6 MHz for North America and Europe, including Spain and France, and 5 MHz for Japan.

The hopping pattern numbers x are divided into three sets. The sets are designed to avoid prolonged collision periods between different hopping sequences in a set. Hopping sequence sets contain 26 sequences for North America and Europe, and 4 sequences per set for Japan:

For North America and most of Europe:

- $x = \{0,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,60,63,66,69,72,75\}$ Set 1
- $x = \{1,4,7,10,13,16,19,22,25,28,31,34,37,40,43,46,49,52,55,58,61,64,67,70,73,76\}$ Set 2
- $x = \{2,5,8,11,14,17,20,23,26,29,32,35,38,41,44,47,50,53,56,59,62,65,68,71,74,77\}$ Set 3

For Japan:

- $x = \{6,9,12,15\}$ Set 1
- $x = \{7,10,13,16\}$ Set 2
- $x = \{8,11,14,17\}$ Set 3

For Spain:

- $x = \{0,3,6,9,12,15,18,21,24\}$ Set 1
- $x = \{1,4,7,10,13,16,19,22,25\}$ Set 2
- $x = \{2,5,8,11,14,17,20,23,26\}$ Set 3

For France:

- $x = \{0,3,6,9,12,15,18,21,24,27,30\}$ Set 1
- $x = \{1,4,7,10,13,16,19,22,25,28,31\}$ Set 2
- $x = \{2,5,8,11,14,17,20,23,26,29,32\}$ Set 3

The three sets of hopping sequences for North America and most of Europe, of 26 patterns each, are listed in Tables B.1, B.2, and B.3 in Annex B. Similarly, there are three sets for Japan of four patterns each. The three sets for Spain have nine patterns each. The three sets for France have 11 patterns each. The channel numbers listed under each pattern refer to the actual frequency values listed in Table 38 and Table 39.

14.6.9 Unwanted emissions

Conformant PMD implementations shall limit the emissions that fall outside of the operating frequency range, defined in Table 36, to the geographically applicable limits.

14.6.10 Modulation

The minimum set of requirements for a PMD to be compliant with the IEEE 802.11 FHSS PHY shall be as follows.

The PMD shall be capable of operating using two-level Gaussian frequency shift key (GFSK) modulation with a nominal bandwidth bit-period (BT)=0.5. The PMD shall accept symbols from the set $\{\{1\},\{0\}\}$ from the PLCP. The symbol $\{1\}$ shall be encoded with a peak deviation of $(+f_d)$, giving a peak transmit frequency of (F_c+f_d) , which is greater than the carrier center frequency (F_c) . The symbol $\{0\}$ shall be encoded with a peak frequency deviation of $(-f_d)$, giving a peak transmit frequency of (F_c-f_d) .

An incoming bit stream at 1 Mbit/s will be converted to symbols at $F_{clk} = 1$ Msymbols/s, as shown in Table 45.

Table 45—Symbol encoding into carrier deviation (1 Mbit/s, 2-GFSK)

Symbol	Carrier deviation
1	$1/2 \times h2 \times F_{clk}$
0	$-1/2 \times h2 \times F_{clk}$
NOTE—These deviation values are measured using the center symbol of 7 consecutive symbols of the same value. The instantaneous deviation will vary due to Gaussian pulse shaping.	

The deviation factor h_2 for 2GFSK (measured as difference between frequencies measured in the middle of 0000 and 1111 patterns encountered in the SFD, divided by 1 MHz) will nominally be 0.32.

The minimum frequency deviation, as shown in Figure 84, shall be greater than 110 kHz relative to the nominal center frequency F_c . F_d is the average center frequency of the last 8 bits of the Preamble Sync field, measured as the deviation at the midsymbol. Midsymbol is defined as the point that is midway between the zero crossings derived from a best fit to the last 8 bits of the Sync field. Maximum deviation is not specified, but modulation is subject to the occupied bandwidth limits of 14.6.5.

The zero crossing error shall be less than $\pm 1/8$ of a symbol period. The zero crossing error is the time difference between the ideal symbol periods and measured crossings of F_c . This is illustrated in Figure 84.

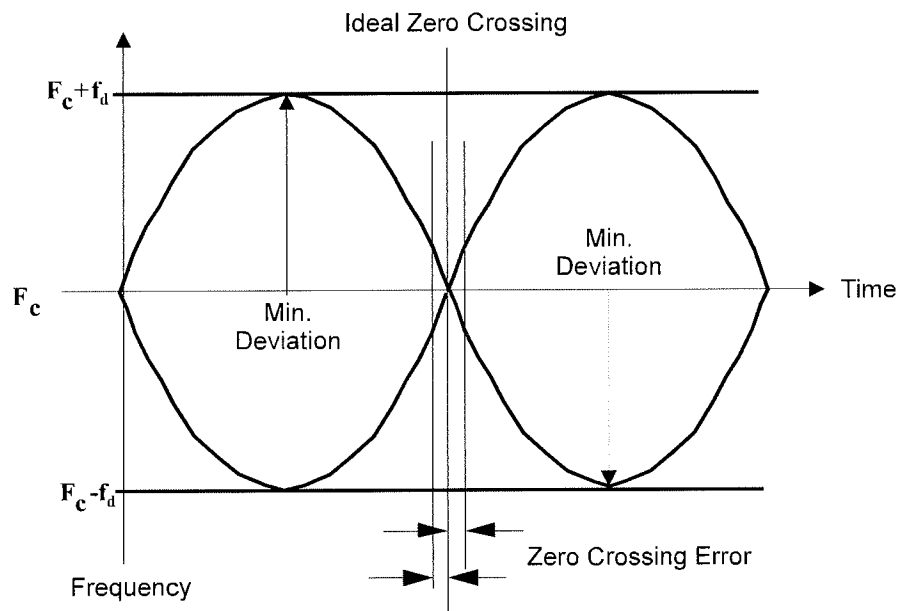


Figure 84—Transmit modulation mask

14.6.11 Channel data rate

A compliant IEEE 802.11 FHSS PMD shall be capable of transmitting and receiving at a nominal data rate of 1.0 Mbit/s \pm 50 parts per million (ppm).

14.6.12 Channel switching/settling time

The time to change from one operating channel frequency, as specified in 14.6.3, is defined as 224 μ s. A conformant PMD meets this switching time specification when the operating channel center frequency has settled to within ± 60 kHz of the nominal channel center frequency as outlined in 14.6.3.

14.6.13 Receive to transmit switch time

The maximum time for a conformant PMD to switch the radio from the receive state to the transmit state and place the start of the first bit on the air shall be 19 μ s. At the end of this 19 μ s, the RF carrier shall be within the nominal transmit power level range, and within the described modulation specifications.

14.6.14 PMD transmit specifications

The following portion of this subclause describes the transmit functions and parameters associated with the PMD sublayer. In general, these are specified by primitives from the PLCP, and the transmit PMD entity provides the actual means by which the signals required by the PLCP primitives are imposed onto the medium.

14.6.14.1 Nominal transmit power

The nominal transmit power of a frame is defined as the power averaged between the start of the first symbol in the PLCP Header to the end of the last symbol in the PLCP Header. When in the transmit state, the transmit power shall be within 2 dB of the nominal transmit power from the start of the Preamble SYNC field to the last symbol at the end of the frame.

14.6.14.2 Transmit power levels

Unless governed by more stringent local geographic regulations, the radiated emissions from compliant devices shall meet IEEE Std C95.1-1991 limits for controlled or uncontrolled environments, in accordance with their intended usage. In addition, all conformant PMD implementations shall support at least one power level with a minimum equivalent isotropically radiated power (EIRP) of 10 mW.

14.6.14.3 Transmit power level control

If a conformant PMD implementation has the ability to transmit in a manner that results in the EIRP of the transmit signal exceeding the level of 100 mW, at least one level of transmit power control shall be implemented. This transmit power control shall be such that the level of the emission is reduced to a level at or below 100 mW under the influence of said power control.

14.6.14.4 Transmit spectrum shape

Within the operational frequency band the transmitter shall pass a spectrum mask test. The duty cycle between Tx and Rx is nominally 50% and the transmit frame length is nominally 400 μ s. The adjacent channel power is defined as the sum of the power measured in a 1 MHz band. For a pseudorandom data pattern, the adjacent channel power shall be a function of the offset between channel number N and the assigned transmitter channel M , where M is the actual transmitted center frequency and N is a channel separated from it by an integer number of megahertz.

Channel offset:

$|N-M|=2$ -20 dBm or -40 dBc, whichever is the lower power.

$|N-M|\geq 3$ -40 dBm or -60 dBc, whichever is the lower power.

The levels given in dBc are measured relative to the transmitter power measured in a 1 MHz channel centered on the transmitter center frequency. The adjacent channel power and the transmitter power for this subclause of the specification shall be measured with a resolution bandwidth of 100 kHz, a video bandwidth of 300 kHz, and a peak detector, and with the measurement device set to maximum hold.

For any transmit center frequency M , two exceptions to the spectrum mask requirements are permitted within the operational frequency band, provided the exceptions are less than -50 dBc, where each offset channel exceeded counts as a separate exception. An exception occurs when the total energy within a given 1 MHz channel as defined in 14.6.5 exceeds the levels specified above.

14.6.14.5 Transmit center frequency tolerance

The PMD transmit center frequency shall be within ± 60 kHz of the nominal center frequency as specified in 14.6.5.

14.6.14.6 Transmitter ramp periods

The transmitter shall go from off to within 2 dB of the nominal transmit power in 8 μ s or less. The transmitter shall go from within 2 dB of the nominal transmit power to off (less than -50 dBm) in 8 μ s or less.

14.6.15 PMD receiver specifications

The following portion of this subclause describes the receive functions and parameters associated with the PMD sublayer. In general, these are specified by primitives from the PLCP. The Receive PMD entity provides the actual means by which the signals required by the PLCP primitives are recovered from the medium. The PMD sublayer monitors signals on the medium and will return symbols from the set $\{\{1\}, \{0\}\}$ to the PLCP sublayer.

14.6.15.1 Input signal range

The PMD shall be capable of recovering a conformant PMD signal from the medium, as described in related subclauses, with a frame error ratio (FER) $\leq 3\%$ for PSDUs of 400 octets generated with pseudorandom data, for receiver input signal levels in the range from -20 dBm to the receiver sensitivity (as specified in 14.6.15.4), across the frequency band of operation.

14.6.15.2 Receive center frequency acceptance range

An IEEE 802.11 FHSS compliant PMD shall meet all specifications with an input signal having a center frequency range of ± 60 kHz from nominal.

14.6.15.3 CCA power threshold

In the presence of any IEEE 802.11 compliant 1 Mbit/s FH PMD signal above -85 dBm that starts synchronously with respect to slot times as specified in 14.3.3.2.1, the PHY shall signal busy, with a 90% probability of detection, during the preamble within the CCA assessment window. In the presence of any IEEE 802.11 compliant 1 Mbit/s FH PMD signal above -85 dBm that starts asynchronously with respect to slot times as specified in 14.3.3.2.1, the PHY shall signal busy, with a 70% probability of detection, during the preamble within the CCA window. In the presence of any IEEE 802.11 compliant 1 Mbit/s FH PMD signal above -65 dBm, the PHY shall signal busy, with a 70% probability of detection, during random data within the CCA window. This specification applies to a PMD operating with a nominal EIRP of < 100 mW. A compliant PMD operating at a nominal output power greater than 100 mW shall use the following equation to define the CCA threshold, where P_t represents transmit power.

$$\text{CCA threshold (preamble)} = -85 \text{ dBm} - \left[5 \times \log_{10} \left(\frac{P_t}{100 \text{ mW}} \right) \right] \text{ dBm}$$

$$\text{CCA threshold (random data)} = \text{CCA threshold (preamble)} + 20 \text{ dB}$$

14.6.15.4 Receiver sensitivity

The sensitivity is defined as the minimum signal level required for an FER of 3% for PSDUs of 400 octets generated with pseudorandom data. The sensitivity shall be less than or equal to -80 dBm. The reference sensitivity is defined as -80 dBm for the 1 Mbit/s FH PHY specifications.

14.6.15.5 Intermodulation

Intermodulation protection (IMp) is defined as the ratio of the minimum amplitude of one of two equal interfering signals to the desired signal amplitude, where the interfering signals are spaced 4 MHz and 8 MHz removed from the center frequency of the desired signal, both on the same side of center frequency. The IMp protection ratio is established at the interfering signal level that causes the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudorandom data, when the desired signal is -77 dBm. Each interfering signal is modulated with the FH PMD modulation uncorrelated in time to each other or the desired signal. The PMD shall have the IMp for the interfering signal at 4 MHz and 8 MHz be ≥ 30 dB.

14.6.15.6 Desensitization

Desensitization (Dp) is defined as the ratio to measured sensitivity of the minimum amplitude of an interfering signal that causes the FER at the output of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudorandom data, when the desired signal is -77 dBm. The interfering signal shall be modulated with the FHSS PMD modulation uncorrelated in time to the desired signal. The minimum Dp shall be as given in Table 46. The spectral purity of the interferer shall be sufficient to ensure that the measurement is limited by the receiver performance.

Table 46—1 Mbit/s Dp

Interferer frequency ^a	Dp minimum
$M = N \pm 2$	30 dB
$M = N \pm 3$ or more	40 dB

^aWhere M is the interferer frequency and N is the desired channel frequency.

14.6.15.7 Receiver radiation

The signal leakage when receiving shall not exceed -50 dBm EIRP in the operating frequency range. The FHSS PHY shall conform with out-of-band spurious emissions by regulatory bodies.

14.6.16 Operating temperature range

Two temperature ranges for full operation compliance to the FH PHY are specified. Type 1 is defined as 0 °C to 40 °C and is designated for office environments. Type 2 is defined as -30 °C to $+70$ °C and is designated for industrial environments.

14.7 FHSS PMD sublayer, 2.0 Mbit/s**14.7.1 Overview**

This subclause details the RF specification differences of the optional 2 Mbit/s operation from the baseline 1 Mbit/s PMD as contained in 14.6. Unless otherwise specified in this subclause, the compliant PMD shall also meet all requirements of 14.6 when transmitting at 2 Mbit/s. When implementing the 2 Mbit/s option, the preamble and PHY Header shall be transmitted at 1 Mbit/s. STAs implementing the 2 Mbit/s option shall also be capable of transmitting and receiving PPDU's at 1 Mbit/s.

14.7.2 Four-Level GFSK modulation

For an FHSS 2 Mbit/s PMD, the modulation scheme shall be four-level Gaussian frequency shift keying (4GFSK), with a nominal symbol-period bandwidth product (BT) of 0.5. The four-level deviation factor, defined as the frequency separation of adjacent symbols divided by symbol rate, h_4 , shall be related to the deviation factor of the 2GFSK modulation, h_2 , by the following equation:

$$h_4/h_2 = 0.45 \pm 0.01$$

An incoming bit stream at 2 Mbit/s will be converted to 2-bit words or symbols, with a rate of $F_{clk} = 1$ Msymbol/s. The first received bit will be encoded as the LMB of the symbol in Table 47. The bits will be encoded into symbols as shown in Table 47.

Table 47—Symbol encoding into carrier deviation

1 Mbit/s, 2GFSK	
Symbol	Carrier deviation
1	$1/2 \times h_2 \times F_{clk}$
0	$-1/2 \times h_2 \times F_{clk}$
2 Mbit/s, 4GFSK	
Symbol	Carrier deviation
10	$3/2 \times h_4 \times F_{clk}$
11	$1/2 \times h_4 \times F_{clk}$
01	$-1/2 \times h_4 \times F_{clk}$
00	$-3/2 \times h_4 \times F_{clk}$
NOTE—These deviation values are measured using the center symbol of 7 consecutive symbols of the same value. The instantaneous deviation will vary due to Gaussian pulse shaping.	

The deviation factor h_2 for 2GFSK (measured as the difference between frequencies measured in the middle of 0000 and 1111 patterns encountered in the SFD, divided by 1 MHz) will nominally be 0.32. The deviation factor h_2 will be no less than 0.30 (with maximum dictated by regulatory bandwidth requirement). Accordingly, h_4 (measured as a difference between the outermost frequencies, divided by 3, divided by 1 MHz) is nominally $0.45 \times 0.32 = 0.144$, and it will be no less than $0.45 \times 0.3 = 0.135$.

The modulation error shall be less than ± 15 kHz at the midsymbol time for 4GFSK, from the frequency deviations specified above, for a symbol surrounded by identical symbols, and less than ± 25 kHz for any symbol. The deviation is relative to the actual center frequency of the RF carrier. For definition purposes, the actual center frequency is the midfrequency between symbols 11 and 01. The actual center frequency shall be within ± 60 kHz of the nominal channel center frequency defined in 14.6.5 and shall not vary by more than ± 10 kHz/ms, from the start to end of the PPDU. The peak-to-peak variation of the actual center frequency over the PPDU shall not exceed 15 kHz. Symbols and terms used within this subclause are illustrated in Figure 85.

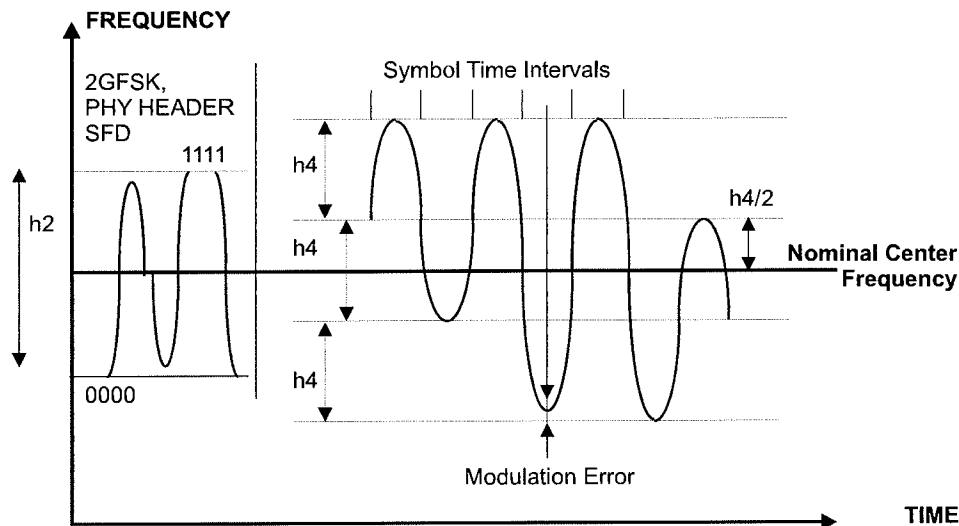


Figure 85—Four-Level GFSK transmit modulation

14.7.2.1 Frame structure for HS FHSS PHY

The high rate FHSS PPDU consists of PLCP Preamble, PLCP Header, and whitened PSDU. The PLCP Preamble and PLCP Header format are identical to the 1 Mbit/s PHY, as described in 14.3.2. The whitened PSDU is transmitted in 2GFSK, 4GFSK, or potentially a higher-rate format, according to the rate chosen. The rate is indicated in a 3-bit field in a PLCP Header, having a value of 1 or 2 bits/symbol (or Mbit/second).

The PPDU is transmitted as four-level symbols, with the amount determined by $\text{number_of_symbols} = (\text{number_of_PSDU_octets} \times 8) / \text{rate}$.

The input bits are scrambled according to the method in 14.3.2.3.

The scrambled bit stream is divided into groups of rate (1 or 2) consecutive bits. The bits are mapped into symbols according to Table 47.

A bias suppression algorithm is applied to the resulting symbol stream. The bias suppression algorithm is defined in 14.3.2.3, Figure 72, and Figure 75. A polarity control symbol is inserted prior to each block of 32 symbols (or less for the last block). The polarity control signals are 4GFSK symbols 10 or 00. The algorithm is equivalent to the case of 2GFSK, with the polarity symbol 2GFSK “1” replaced with 4GFSK symbol “10,” and the 2GFSK polarity symbol “0” replaced with a 4GFSK symbol “00.”

14.7.3 Channel data rate

The data rate for the whitened PSDU at the optional rate shall be $2.0 \text{ Mbit/s} \pm 50 \text{ ppm}$.

14.7.3.1 Input dynamic range

The PMD shall be capable of recovering a conformant PMD signal from the medium, as described in related subclauses, with an FER $\leq 3\%$ for PSDUs of 400 octets generated with pseudorandom data, for receiver input signal levels in the range from -20 dBm to the receiver sensitivity (as specified in 14.7.3.2), across the frequency band of operation.

14.7.3.2 Receiver sensitivity

The sensitivity is defined as the minimum signal level required for an FER of 3% for PSDUs of 400 octets generated with pseudorandom data. The sensitivity shall be less than or equal to -75 dBm. The reference sensitivity is defined as -75 dBm for the 2 Mbit/s FH PHY specifications.

14.7.3.3 IMp

IMp is defined as the ratio to -77 dBm of the minimum amplitude of one of the two equal-level interfering signals at 4 MHz and 8 MHz removed from center frequency, both on the same side of center frequency, that cause the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudorandom data, when the desired signal is -72 dBm (3 dB above the specified sensitivity specified in 14.7.3.2). Each interfering signal is modulated with the FH 1 Mbit/s PMD modulation uncorrelated in time to each other or the desired signal. The FHSS optional 2 Mbit/s rate IMp shall be ≥ 25 dB.

14.7.3.4 Dp

Dp is defined as the ratio to measured sensitivity of the minimum amplitude of an interfering signal that causes the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudorandom data, when the desired signal is -72 dB (3 dB above sensitivity specified in 14.7.3.2). The interfering signal shall be modulated with the FHSS PMD modulation uncorrelated in time to the desired signal. The minimum Dp shall be as given in Table 48.

Table 48—2 Mbit/s Dp

Interferer frequency ^a	DP minimum
$M = N \pm 2$	20 dB
$M = N \pm 3$ or more	30 dB

^aWhere M is the interferer frequency and N is the desired channel frequency.

14.8 FHSS PHY management information base (MIB)

14.8.1 Overview

The following is the MIB for the FHSS PHY.

14.8.2 FH PHY attributes

This subclause defines the attributes for the FHSS MIB. Table 49 lists these attributes and the default values. Following the table is a description of each attribute.

Table 49—FHSS PHY attributes

Attribute	Default value	Operational semantics	Operational behavior
dot11PHYType	FHSS = X'01'	Static	Identical for all FH PHYs
dot11RegDomainsSupported	FCC = X'10' IC = X'20' ETSI = X'30' Spain = X'31' France = X'32' MKK = X'40'	Static	Implementation dependent
dot11CurrentRegDomain	X'00'	Dynamic LME	Implementation dependent
dot11TempType	Type 1 = X'01' Type 2 = X'02' Type 3 = X'03'	Static	Implementation dependent
dot11SupportedDataRatesTX	1 Mbit/s = X'02' mandatory 2 Mbit/s = X'04' optional	Static	Identical for all FH PHYs
dot11SupportedDataRatesRX	1 Mbit/s = X'02' mandatory 2 Mbit/s = X'04' optional	Static	Identical for all FH PHYs
dot11SupportedTxAntennas	Ant 1 = X'01' Ant 2 = X'02' Ant 3 = X'03' Ant n = n	Static	Implementation dependent
dot11CurrentTxAntenna	Ant 1 = default	Dynamic LME	Implementation dependent
dot11SupportedRxAntennas	Ant 1 = X'01' Ant 2 = X'02' Ant 3 = X'03' Ant n = n	Static	Implementation dependent
dot11DiversitySupport	Available = X'01' Not avail. = X'02' Control avail. = X'03'	Static	Implementation dependent
dot11DiversitySelectionRx	Ant 1 = X'01' Ant 2 = X'02' Ant 3 = X'03' Ant 4 = X'04' Ant 5 = X'05' Ant 6 = X'06' Ant 7 = X'07' Ant 8 = X'08'	Dynamic LME	Implementation dependent
dot11NumberSupportedPowerLevels	Lvl1 = X'01' Lvl2 = X'02' Lvl3 = X'03' Lvl4 = X'04' Lvl5 = X'05' Lvl6 = X'06' Lvl7 = X'07' Lvl8 = X'08'	Static	Implementation dependent
dot11TxPowerLevel1	Factory default	Static	Implementation dependent
dot11TxPowerLevel2	Factory default	Static	Implementation dependent
dot11TxPowerLevel3	Factory default	Static	Implementation dependent

Table 49—FHSS PHY attributes (continued)

Attribute	Default value	Operational semantics	Operational behavior
dot11TxPowerLevel4	Factory def.	Static	Implementation dependent
dot11TxPowerLevel5	Factory def.	Static	Implementation dependent
dot11TxPowerLevel6	Factory def.	Static	Implementation dependent
dot11TxPowerLevel7	Factory def.	Static	Implementation dependent
dot11TxPowerLevel8	Factory def.	Static	Implementation dependent
dot11CurrentTxPowerLevel	TxPowerLevel1	Dynamic LME	Implementation dependent
dot11HopTime	224 μ s	Static	Identical for all FH PHYs
dot11CurrentChannelNumber	X'00'	Dynamic PLME	
dot11MaxDwellTime	390 TU	Static	Regulatory domain dependent
dot11CurrentSet	X'00'	Dynamic PLME	
dot11CurrentPattern	X'00'	Dynamic PLME	
dot11CurrentIndex	X'00'	Dynamic PLME	
dot11CurrentPowerState	X'01' off X'02' on	Dynamic LME	

NOTE—The column titled “Operational semantics” contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. MIB attributes defined as dynamic can be modified by some management entity. Whenever an attribute is defined as dynamic, the column also shows which entity has control over the attribute. LME refers to the MAC sublayer management entity (MLME), while PHY refers to the physical layer management entity (PLME).

14.8.2.1 FH PHY attribute definitions

14.8.2.1.1 dot11PHYType

The dot11PHYType is FHSS. The LME uses this attribute to determine what PLCP and PMD are providing services to the MAC. It also is used by the MAC to determine what MAC sublayer management state machines must be invoked to support the PHY. The value of this attribute is defined as the integer 01 to indicate the FHSS PHY.

14.8.2.1.2 dot11RegDomainsSupported

Operational requirements for FHSS PHY are defined by agencies representing certain geographical regulatory domains. These regulatory agencies may define limits on various parameters that differ from region to region. This parameters may include dot11TxPowerLevels, and dot11MaxDwellTime, as well as the total number of frequencies in the hopping pattern. The values shown in Table 50 indicate regulatory agencies supported by this document.

Since a PLCP and PMD might be designed to support operation in more than one regulatory domain, this attribute can actually represent a list of agencies. This list can be one or more of the above agencies and must be terminated using the null terminator. Upon activation of the PLCP and PMD, the information in this list must be used to set the value of the dot11CurrentRegDomain attribute.

14.8.2.1.3 dot11CurrentRegDomain

The dot11CurrentRegDomain attribute for the FHSS PHY is defined as the regulatory domain under which the PMD is currently operating. This value must be one of the values listed in the dot11RegDomainsSupported list. This MIB attribute is managed by the LME.

Table 50—Regulatory domain codes

Code point	Regulatory agency	Region
X'10'	FCC	United States
X'20'	IC	Canada
X'30'	ETSI	Most of Europe
X'31'	Spain	Spain
X'32'	France	France
X'40'	MKK	Japan

14.8.2.1.4 dot11TempType

The parameter dot11TempType defines the temperature range supported by the PHY. Type 1 equipment (X'01') supports a temperature range of 0 °C to 40 °C. Type 2 equipment (X'02') supports a temperature range of –20 °C to +55 °C. Type 3 equipment (X'03') supports a temperature range of –30 °C to +70 °C.

14.8.2.1.5 dot11CurrentPowerState

The dot11CurrentPowerState attribute for the FHSS PHY allows the MAC sublayer management entity to control the power state of the PHY. This attribute can be updated using the PLMESET.request. The permissible values are ON and OFF.

14.8.2.1.6 dot11SupportedDataRatesTX

The dot11SupportedDataRatesTX attribute for the FHSS PHY is defined as a null terminated list of supported data rates in the transmit mode for this implementation. Table 51 shows the possible values appearing in the list.

Table 51—Supported data rate codes (dot11SupportedDataRatesTX)

Code point	Data rate
X'02'	1 Mbit/s
X'04'	2 Mbit/s
X'00'	Null terminator

14.8.2.1.7 dot11SupportedDataRatesRX

The dot11SupportedDataRatesRX attribute for the FHSS PHY is defined as a null terminated list of supported data rates in the receive mode for this implementation. Table 52 shows the possible values appearing in the list.

14.8.2.1.8 aMPDUMaxLength

The aMPDUMaximumLength attribute for the FHSS PHY is defined as the maximum PSDU, in octets, that the PHY shall ever be capable of accepting. This value for the FHSS PHY is set at 4095 octets. The recommended value for maximum PSDU length in an FHSS PHY system is 400 octets at 1 Mbit/s and 800 octets at 2 Mbit/s, which corresponds to a frame duration less than 3.5 ms. These values are optimized to achieve

Table 52—Supported data rate codes (dot11SupportedDataRatesRX)

Code point	Data rate
X'02'	1 Mbit/s
X'04'	2 Mbit/s
X'00'	Null terminator

high performance in a variety of RF channel conditions, particularly with respect to indoor multipath, channel stability for moving STAs, and interference in the 2.4 GHz band.

14.8.2.1.9 dot11SupportedTxAntennas

The dot11SupportedTxAntennas attribute for the FHSS PHY is defined as a null terminated list of antennas that this implementation can use to transmit data. Table 53 shows the possible values appearing in the list, where $N \leq 255$.

Table 53—Number of transmit antennas

Code point	Antenna number
X'01'	Tx Antenna 1
X'02'	Tx Antenna 2
X'03'	Tx Antenna 3
...	...
N	Tx Antenna N
X'00'	Null terminator

14.8.2.1.10 dot11CurrentTxAntenna

The dot11CurrentTxAntenna attribute for the FHSS PHY is used to describe the current antenna the implementation is using for transmission. This value should represent one of the antennas appearing in the dot11SupportedTxAntennas list.

14.8.2.1.11 dot11SupportedRxAntenna

The dot11SupportedRxAntennas attribute for the FHSS PHY is defined as a null terminated list of antennas that this implementation can use to receive data. In the FHSS PHY primitives, one of these values is passed as part of the PHY-RXSTART.indicate to the MAC sublayer for every received packet. Table 54 shows the possible values appearing in the list, where $N \leq 255$.

Table 54—Number of receive antennas

Code point	Antenna number
X'01'	Rx Antenna 1
X'02'	Rx Antenna 2
X'03'	Rx Antenna 3
...	...
<i>N</i>	Rx Antenna <i>N</i>
X'00'	Null terminator

14.8.2.1.12 dot11DiversitySupport

The dot11DiversitySupport attribute for the FHSS PHY is used to describe the implementation's diversity support. Table 55 shows the possible values appearing in the list.

Table 55—Diversity support codes

Code point	Diversity support
X'01'	Diversity available
X'02'	No diversity
X'03'	Control available

The value X'01' indicates that this implementation uses two or more antennas for diversity. The value X'02' indicates that the implementation has no diversity support. The value X'03' indicates that the choice of antennas used during diversity is programmable. (See 14.8.2.1.13.)

14.8.2.1.13 dot11DiversitySelectionRx

The dot11DiversitySelectionRx attribute for the FHSS PHY is a null terminated list describing the receive antenna or antennas currently in use during diversity and packet reception. Table 56 shows the possible values appearing in the list, where $N \leq 255$.

Table 56—Diversity select antenna codes

Code point	Antenna number
X'01'	Rx Antenna 1
X'02'	Rx Antenna 2
X'03'	Rx Antenna 3
...	...
<i>N</i>	Rx Antenna <i>N</i>
X'00'	Null terminator

The null terminated list can consist of one or more of the receive antennas listed in the dot11SupportedRxAntennas attribute. This attribute can be changed dynamically by the LME.

14.8.2.1.14 dot11NumberSupportedPowerLevels

The dot11NumberSupportedPowerLevels attribute for the FHSS PHY describes the number of power levels this implementation supports. This attribute can be an integer of value 1 through 8, inclusive.

14.8.2.1.15 dot11TxPowerLevel1-8

Some implementations may provide up to eight different transmit power levels. The dot11TxPowerLevels attribute for the FHSS PHY is a list of up to eight power levels supported. Table 57 describes the list.

Table 57—Transmit power levels

Attribute	Power level
TxPowerLevel1	Default setting
TxPowerLevel2	Level 2
TxPowerLevel3	Level 3
TxPowerLevel4	Level 4
TxPowerLevel5	Level 5
TxPowerLevel6	Level 6
TxPowerLevel7	Level 7
TxPowerLevel8	Level 8

14.8.2.1.16 dot11CurrentTxPowerLevel

The dot11CurrentTxPowerLevel attribute for the FHSS PHY is defined as the current transmit output power level. This level shall be one of the levels implemented in the list of attributes called dot11TxPowerLevel N (where N is 1–8). This MIB attribute is also used to define the sensitivity of the CCA mechanism when the output power exceeds 100 mW. This MIB attribute is managed by the LME.

14.8.2.1.17 dot11HopTime

The dot11HopTime attribute for the FHSS PHY describes the time allocated for the PHY to change to a new frequency. For the FHSS PHY, this time period is 224 μ s.

14.8.2.1.18 dot11CurrentChannelNumber

The dot11CurrentChannelNumber attribute for the FHSS PHY is defined as the current operating channel number of the PMD. The values of this attribute correspond to the values shown in Table 38. This MIB attribute is managed by the PLME and is updated as the result of a PLMESET.request to dot11CurrentSet, dot11CurrentPattern, or dot11CurrentIndex.

14.8.2.1.19 dot11MaxDwellTime

The dot11MaxDwellTime attribute for the FHSS PHY is defined as the maximum time the PMD can dwell on a channel and meet the requirements of the current regulatory domain. For the FCC regulatory domain, this number is 390 TU (FCC = 400 ms). The recommended dwell time for the FHSS PHY is 19 TU.

14.8.2.1.20 dot11CurrentSet

The FHSS PHY contains three sets of hopping patterns. The dot11CurrentSet attribute for the FHSS PHY defines what set the STA is using to determine the hopping pattern. Its value can be 1, 2, or 3. This attribute is managed by the PLME.

14.8.2.1.21 dot11CurrentPattern

There are up to 78 patterns in each hopping set used by the FHSS PHY. The dot11CurrentPattern attribute for the FHSS PHY defines the x value used in Equation (1) in 14.6.8 to calculate the current channel number. Its value has various ranges, always within the overall range of 0 to 77, depending on the dot11CurrentRegDomain. This attribute is managed by the PLME.

14.8.2.1.22 dot11CurrentIndex

The FHSS PHY addresses each channel in the selected hopping pattern through an index. The dot11CurrentIndex attribute for the FHSS PHY defines the i value used in the equation for $f_x(i)$ in 14.6.8 to calculate the current channel number. Its value has various ranges, always within the overall range of 1 to 79, depending on the dot11CurrentRegDomain. This attribute is managed by the PLME.

14.8.2.1.23 dot11CurrentPowerState

The parameter dot11CurrentPowerState defines the operational state of the FHSS PHY. When this attribute has a value of X'01', the PHY is "OFF." When this attribute has a value of X'02', the PHY is "ON." This attribute is managed by the PLME.

14.9 FH PHY characteristics

Following are the static FH PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive. The definitions of these characteristics are in 10.4.3.

Table 57a—FH PHY characteristics

Characteristic	Value	Notes
aSlotTime	50 μ s	—
aSIFSTime	28 μ s	In order to account for variations between implementations, this value has a tolerance as specified in 9.2.3.1.
aCCATime	27 μ s	This period includes the aRxRFDelay and the aRxPLCPDelay.
aRxTxTurnaroundTime	20 μ s	—
aTxPLCPDelay	1 μ s	Implementors may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	2 μ s	Implementors may choose to increase or decrease this delay as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	10 μ s	Implementors may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOnTime	8 μ s	Implementors may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOffTime	8 μ s	—
aTxRFDelay	1 μ s	Implementors may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	4 μ s	Implementors may choose to increase or decrease this delay as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	1 μ s	Variations in the actual propagation time are accounted for in the allowable range of aSIFSTime.
aMACProcessingDelay	2 μ s	Implementors may choose to increase or decrease this delay as long as the requirements of aSIFSTime are met.
aPreambleLength	96 μ s	—
aPLCPHeaderLength	32 μ s	—
aMPDUDurationFactor	31250000	This factor is calculated as $[(33/32) - 1] \times 10^9$ to account for the expansion due to the data whitener encoding algorithm.
aMPDUMaxLength	4095	The recommended value for maximum PSDU length in an FHSS PHY system is 400 octets at 1 Mbit/s and 800 octets at 2 Mbit/s, which corresponds to a frame duration less than 3.5 ms. These values are optimized to achieve high performance in a variety of RF channel conditions, particularly with respect to indoor multipath, channel stability for moving STAs, and interference in the 2.4 GHz band.
aCWmin	15	—
aCWmax	1023	—

15. Direct sequence spread spectrum (DSSS) PHY specification for the 2.4 GHz band designated for ISM applications

15.1 Overview

The PHY for the direct sequence spread spectrum (DSSS) system is described in this clause. The RF LAN system is initially aimed for the 2.4 GHz band designated for ISM applications as provided in the USA according to FCC 15.247, in Europe by ETS 300–328, and in other countries according to 15.4.6.2.

The DSSS system provides a wireless LAN with both a 1 Mbit/s and a 2 Mbit/s data payload communication capability. According to the FCC regulations, the DSSS system shall provide a processing gain of at least 10 dB. This shall be accomplished by chipping the baseband signal at 11 MHz with an 11-chip PN code. The DSSS system uses baseband modulations of differential binary phase shift keying (DBPSK) and differential quadrature phase shift keying (DQPSK) to provide the 1 Mbit/s and 2 Mbit/s data rates, respectively.

15.1.1 Scope

The PHY services provided to the IEEE 802.11 wireless LAN MAC by the 2.4 GHz DSSS system are described in this clause. The DSSS PHY layer consists of two protocol functions:

- a) A physical layer convergence function, which adapts the capabilities of the physical medium dependent (PMD) system to the PHY service. This function shall be supported by the physical layer convergence procedure (PLCP), which defines a method of mapping the IEEE 802.11 MAC sublayer protocol data units (MPDU) into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.
- b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a wireless medium (WM) between two or more STAs each using the DSSS system.

15.1.2 DSSS PHY functions

The 2.4 GHz DSSS PHY architecture is depicted in the reference model shown in Figure 11. The DSSS PHY contains three functional entities: the PMD function, the physical layer convergence function, and the layer management function. Each of these functions is described in detail in the following subclauses.

The DSSS PHY service shall be provided to the MAC through the PHY service primitives described in Clause 12.

15.1.2.1 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a physical layer convergence sublayer is defined. This function simplifies the PHY service interface to the IEEE 802.11 MAC services.

15.1.2.2 PMD sublayer

The PMD sublayer provides a means to send and receive data between two or more STAs. This clause is concerned with the 2.4 GHz ISM bands using direct sequence modulation.

15.1.2.3 Physical layer management entity (PLME)

The PLME performs management of the local PHY functions in conjunction with the MAC management entity.

15.1.3 Service specification method and notation

The models represented by figures and state diagrams are intended to be illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the IEEE 802.11 DSSS PHY compliant developer.

The service of a layer or sublayer is a set of capabilities that it offers to a user in the next-higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation.

15.2 DSSS PLCP sublayer

15.2.1 Overview

This clause provides a convergence procedure in which MPDUs are converted to and from PPDU. During transmission, the MPDU shall be prepended with a PLCP Preamble and Header to create the PPDU. At the receiver, the PLCP Preamble and header are processed to aid in demodulation and delivery of the MPDU.

15.2.2 PLCP frame format

Figure 86 shows the format for the PPDU including the DSSS PLCP Preamble, the DSSS PLCP Header, and the MPDU. The PLCP Preamble contains the following fields: Synchronization (Sync) and Start Frame Delimiter (SFD). The PLCP Header contains the following fields: IEEE 802.11 Signaling (Signal), IEEE 802.11 Service (Service), LENGTH (Length), and CCITT CRC-16. Each of these fields is described in detail in 15.2.3.

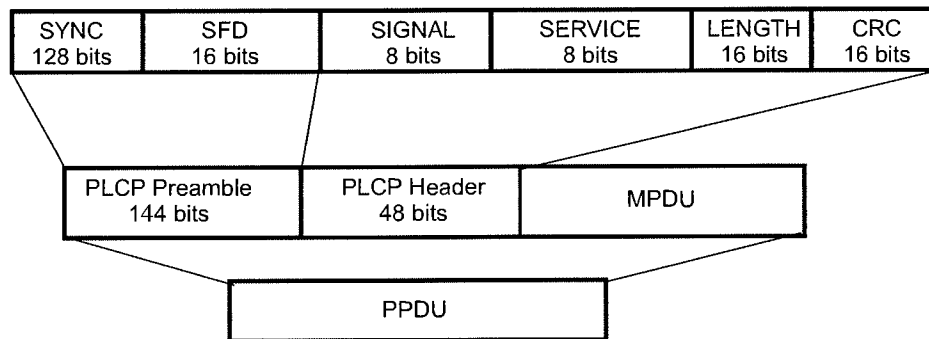


Figure 86—PLCP frame format

15.2.3 PLCP field definitions

The entire PLCP Preamble and Header shall be transmitted using the 1 Mbit/s DBPSK modulation described in 15.4.7. All transmitted bits shall be scrambled using the feedthrough scrambler described in 15.2.4.

15.2.3.1 PLCP Synchronization (SYNC) field

The SYNC field shall consist of 128 bits of scrambled ones. This field shall be provided so that the receiver can perform the necessary operations for synchronization.

15.2.3.2 PLCP Start Frame Delimiter (SFD)

The SFD shall be provided to indicate the start of PHY-dependent parameters within the PLCP Preamble. The SFD shall be a 16-bit field, X'F3A0' (msb to lsb). The lsb shall be transmitted first in time.

15.2.3.3 PLCP IEEE 802.11 Signal (SIGNAL) field

The 8-bit IEEE 802.11 signal field indicates to the PHY the modulation that shall be used for transmission (and reception) of the MPDU. The data rate shall be equal to the signal field value multiplied by 100 kbit/s. The DSSS PHY currently supports two mandatory modulation services given by the following 8-bit words, where the lsb shall be transmitted first in time:

- a) X'0A' (msb to lsb) for 1 Mbit/s DBPSK
- b) X'14' (msb to lsb) for 2 Mbit/s DQPSK

The DSSS PHY rate change capability is described in 15.2.5. This field shall be protected by the CCITT CRC-16 frame check sequence described in 15.2.3.6.

15.2.3.4 PLCP IEEE 802.11 Service (SERVICE) field

The 8-bit IEEE 802.11 service field shall be reserved for future use. The value of X'00' signifies IEEE 802.11 device compliance. The lsb shall be transmitted first in time. This field shall be protected by the CCITT CRC-16 frame check sequence described in 15.2.3.6.

15.2.3.5 PLCP Length (LENGTH) field

The PLCP Length field shall be an unsigned 16-bit integer that indicates the number of microseconds (16 to $2^{16}-1$ as defined by `aMPDUMaxLength`) required to transmit the MPDU. The transmitted value shall be determined from the LENGTH parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 12.3.5.4. The length field provided in the TXVECTOR is in bytes and is converted to microseconds for inclusion in the PLCP LENGTH field. The lsb shall be transmitted first in time. This field shall be protected by the CCITT CRC-16 frame check sequence described in 15.2.3.6.

15.2.3.6 PLCP CRC (CCITT CRC-16) field

The IEEE 802.11 SIGNAL, IEEE 802.11 SERVICE, and LENGTH fields shall be protected with a CCITT CRC-16 frame check sequence (FCS). The CCITT CRC-16 FCS shall be the one's complement of the remainder generated by the modulo 2 division of the protected PLCP fields by the polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

The protected bits shall be processed in transmit order. All FCS calculations shall be made prior to data scrambling.

As an example, the SIGNAL, SERVICE, and LENGTH fields for a DBPSK signal with a packet length of 192 μ s (24 bytes) would be given by the following:

0101 0000 0000 0000 0000 0011 0000 0000 (leftmost bit transmitted first in time)

The one's complement FCS for these protected PLCP Preamble bits would be the following:

0101 1011 0101 0111 (leftmost bit transmitted first in time)

Figure 87 depicts this example.

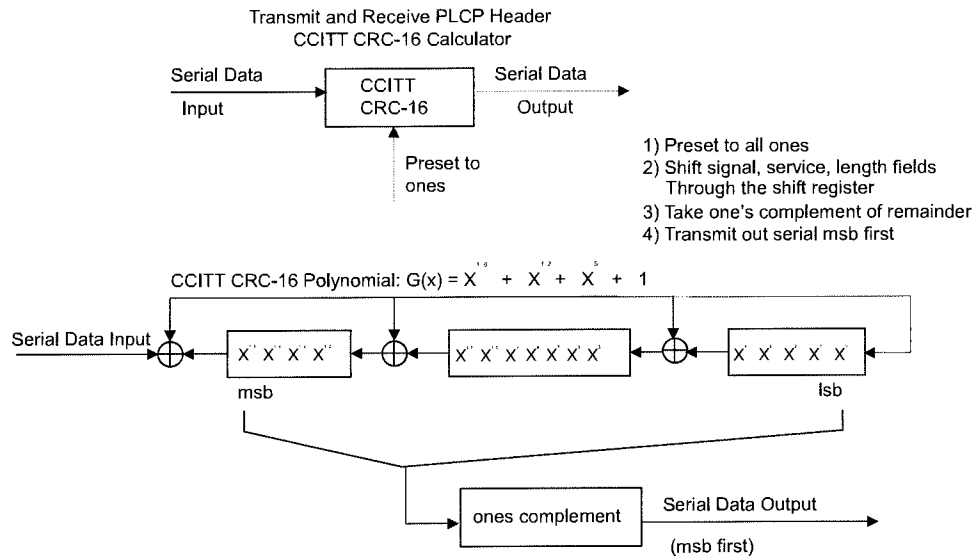


Figure 87—CCITT CRC-16 implementation

An illustrative example of the CCITT CRC-16 FCS using the information from Figure 87 follows in Figure 88.

Data	CRC registers	
	msb lsb	
	1111111111111111	; initialize preset to 1's
0	1110111111011111	
1	1101111111011110	
0	10101111110101101	
1	01011111010111010	
0	1011110101110100	
0	0110101011001001	
0	1101010110010010	
0	1011101100000101	
0	0110011000101011	
0	1100110001010110	
0	1000100010001101	
0	0000000100111011	
0	0000001001110110	
0	000010011101100	
0	0000100111011000	
0	0010011101100000	
0	0100111011000000	
0	1001110110000000	
0	0010101100100001	
0	0101011001000010	
0	1010110010000100	
1	0101100100001000	
1	1010001000110001	
0	0101010001000011	
0	1010100010000110	
0	0100000100101101	
0	1000001001011010	
0	0001010010010101	
0	0010100100101010	
0	0101001001010100	
0	1010010010101000	
	0101101101010111	; one's complement, result = CRC FCS parity

Figure 88—Example CRC calculation

15.2.4 PLCP/DSSS PHY data scrambler and descrambler

The polynomial $G(z) = z^{-7} + z^{-4} + 1$ shall be used to scramble all bits transmitted by the DSSS PHY. The feedthrough configuration of the scrambler and descrambler is self-synchronizing, which requires no prior knowledge of the transmitter initialization of the scrambler for receive processing. Figure 89 and Figure 90 show typical implementations of the data scrambler and descrambler, but other implementations are possible.

The scrambler should be initialized to any state except all ones when transmitting.

Scrambler Polynomial; $G(z)=Z^{-7} + Z^{-4} + 1$

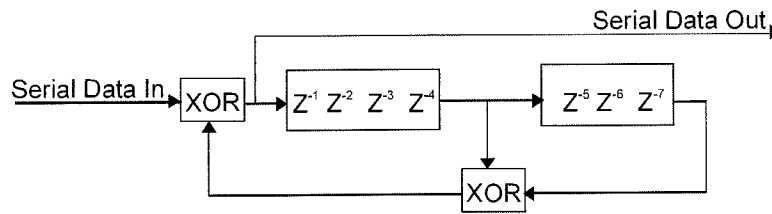


Figure 89—Data scrambler

Descrambler Polynomial; $G(z)=Z^{-7} + Z^{-4} + 1$

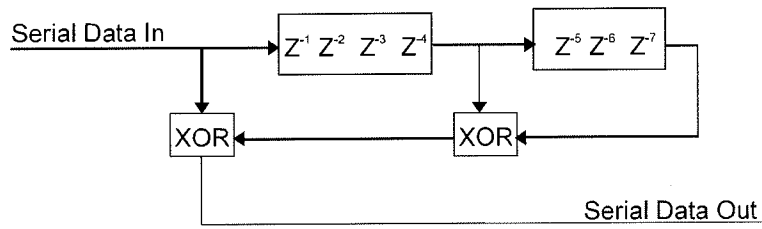


Figure 90—Data descrambler

15.2.5 PLCP data modulation and modulation rate change

The PLCP Preamble shall be transmitted using the 1 Mbit/s DBPSK modulation. The IEEE 802.11 SIGNAL field shall indicate the modulation that shall be used to transmit the MPDU. The transmitter and receiver shall initiate the modulation indicated by the IEEE 802.11 SIGNAL field starting with the first symbol (1 bit for DBPSK or 2 bits for DQPSK) of the MPDU. The MPDU transmission rate shall be set by the DATA-RATE parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 15.4.4.1.

15.2.6 PLCP transmit procedure

The PLCP transmit procedure is shown in Figure 91.

In order to transmit data, PHY-TXSTART.request shall be enabled so that the PHY entity shall be in the transmit state. Further, the PHY shall be set to operate at the appropriate channel through station management via the PLME. Other transmit parameters such as DATARATE, TX antenna, and TX power are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR) as described in 15.4.4.2.

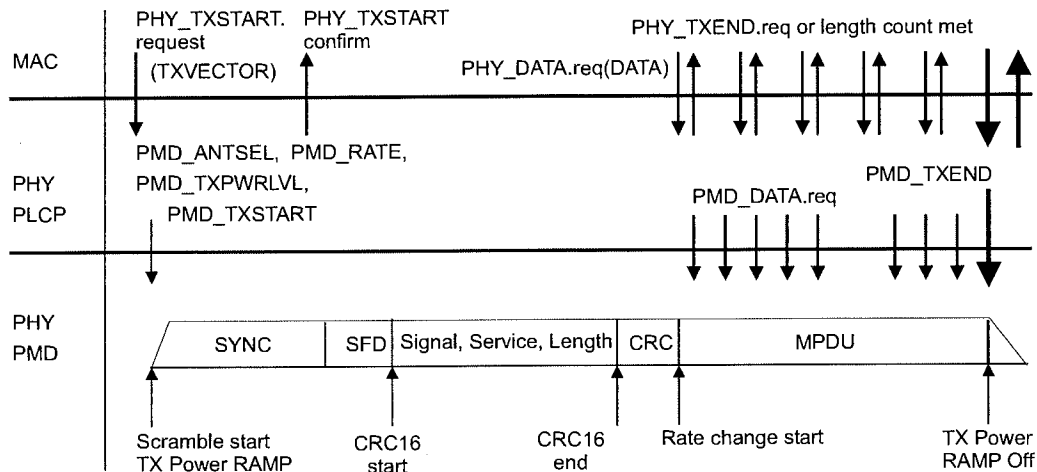


Figure 91—PLCP transmit procedure

Based on the status of clear channel assessment (CCA) indicated by PHY-CCA.indicate, the MAC will assess that the channel is clear. A clear channel shall be indicated by PHY-CCA.indicate(IDLE). If the channel is clear, transmission of the PPDU shall be initiated by issuing the PHY-TXSTART.request (TXVECTOR) primitive. The TXVECTOR elements for the PHY-TXSTART.request are the PLCP Header parameters SIGNAL (DATARATE), SERVICE, and LENGTH, and the PMD parameters of TX_ANTENNA and TX_PWR_LEVEL. The PLCP Header parameter LENGTH is calculated from the TXVECTOR element by multiplying by 8 for 1 Mbit/s and by 4 for 2 Mbit/s.

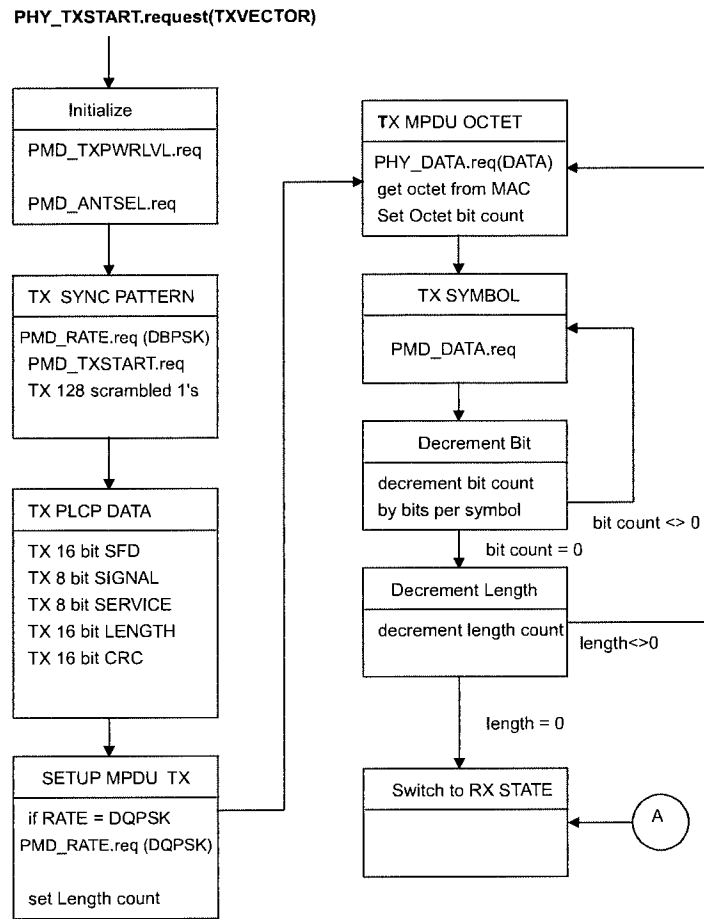
The PLCP shall issue PMD_ANTSEL, PMD_RATE, and PMD_TXPWRLVL primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request and the PHY entity shall immediately initiate data scrambling and transmission of the PLCP Preamble based on the parameters passed in the PHY-TXSTART.request primitive. The time required for TX power-up ramp described in 15.4.7.7 shall be included in the PLCP synchronization field. Once the PLCP Preamble transmission is complete, data shall be exchanged between the MAC and the PHY by a series of PHY-DATA.request(DATA) primitives issued by the MAC and PHY-DATA.confirm primitives issued by the PHY. The modulation rate change, if any, shall be initiated with the first data symbol of the MPDU as described in 15.2.5. The PHY proceeds with MPDU transmission through a series of data octet transfers from the MAC. At the PMD layer, the data octets are sent in lsb to msb order and presented to the PHY layer through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the primitive PHY-TXEND.request. PHY-TXSTART shall be disabled by the issuance of the PHY-TXEND.request. Normal termination occurs after the transmission of the final bit of the last MPDU octet according to the number supplied in the DSSS PHY preamble LENGTH field. The packet transmission shall be completed and the PHY entity shall enter the receive state (i.e., PHY-TXSTART shall be disabled). It is recommended that chirping continue during power-down. Each PHY-TXEND.request is acknowledged with a PHY-TXEND.confirm primitive from the PHY.

A typical state machine implementation of the PLCP transmit procedure is provided in Figure 92.

15.2.7 PLCP receive procedure

The PLCP receive procedure is shown in Figure 93.

In order to receive data, PHY-TXSTART.request shall be disabled so that the PHY entity is in the receive state. Further, through station management via the PLME, the PHY is set to the appropriate channel and the CCA method is chosen. Other receive parameters such as receive signal strength indication (RSSI), signal quality (SQ), and indicated DATARATE may be accessed via the PHY-SAP.



A At any stage in the above flow diagram, if a PHY_THEND. Request is received.

Figure 92—PLCP transmit state machine

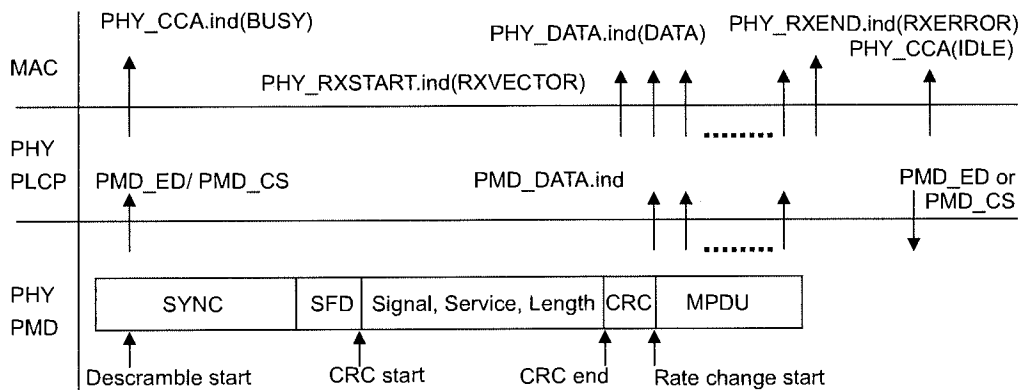


Figure 93—PLCP receive procedure

Upon receiving the transmitted energy, according to the selected CCA mode, the PMD_ED shall be enabled (according to 15.4.8.4) as the RSSI strength reaches the ED_THRESHOLD and/or PMD_CS shall be enabled after code lock is established. These conditions are used to indicate activity to the MAC via PHY-CCA.indicate according to 15.4.8.4. PHY-CCA.indicate(BUSY) shall be issued for energy detection and/or code lock prior to correct reception of the PLCP frame. The PMD primitives PMD_SQ and PMD_RSSI are issued to update the RSSI and SQ parameters reported to the MAC.

After PHY-CCA.indicate is issued, the PHY entity shall begin searching for the SFD field. Once the SFD field is detected, CCITT CRC-16 processing shall be initiated and the PLCP IEEE 802.11 SIGNAL, IEEE 802.11 SERVICE and LENGTH fields are received. The CCITT CRC-16 FCS shall be processed. If the CCITT CRC-16 FCS check fails, the PHY receiver shall return to the RX Idle state as depicted in Figure 94. Should the status of CCA return to the IDLE state during reception prior to completion of the full PLCP processing, the PHY receiver shall return to the RX Idle state.

If the PLCP Header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indicate(RXVECTOR) shall be issued. The RXVECTOR associated with this primitive includes the SIGNAL field, the SERVICE field, the MPDU length in bytes (calculated from the LENGTH field in microseconds), the antenna used for receive (RX_ANTENNA), RSSI, and SQ.

The received MPDU bits are assembled into octets and presented to the MAC using a series of PHY-DATA.indicate(DATA) primitive exchanges. The rate change indicated in the IEEE 802.11 SIGNAL field shall be initiated with the first symbol of the MPDU as described in 15.2.5. The PHY proceeds with MPDU reception. After the reception of the final bit of the last MPDU octet indicated by the PLCP Preamble LENGTH field, the receiver shall be returned to the RX Idle state as shown in Figure 94. A PHY-RXEND.indicate(NoError) primitive shall be issued. A PHY-CCA.indicate(IDLE) primitive shall be issued following a change in PHYCS (PHY carrier sense) and/or PHYED (PHY energy detection) according to the selected CCA method.

In the event that a change in PHYCS or PHYED would cause the status of CCA to return to the IDLE state before the complete reception of the MPDU as indicated by the PLCP LENGTH field, the error condition PHY-RXEND.indicate(CarrierLost) shall be reported to the MAC. The DSSS PHY will ensure that the CCA will indicate a busy medium for the intended duration of the transmitted packet.

If the PLCP Header is successful, but the indicated rate in the SIGNAL field is not receivable, a PHY-RXSTART.indicate will not be issued. The PHY shall issue the error condition PHY-RXEND.indicate(UnsupportedRate). If the PLCP Header is successful, but the SERVICE field is out of IEEE 802.11 DSSS specification, a PHY-RXSTART.indicate will not be issued. The PHY shall issue the error condition PHY-RXEND.indicate(FormatViolation). Also, in both cases, the DSSS PHY will ensure that the CCA shall indicate a busy medium for the intended duration of the transmitted frame as indicated by the Length field. The intended duration is indicated by the Length field ($\text{length} \times 1 \mu\text{s}$).

A typical state machine implementation of the PLCP receive procedure is provided in Figure 94.

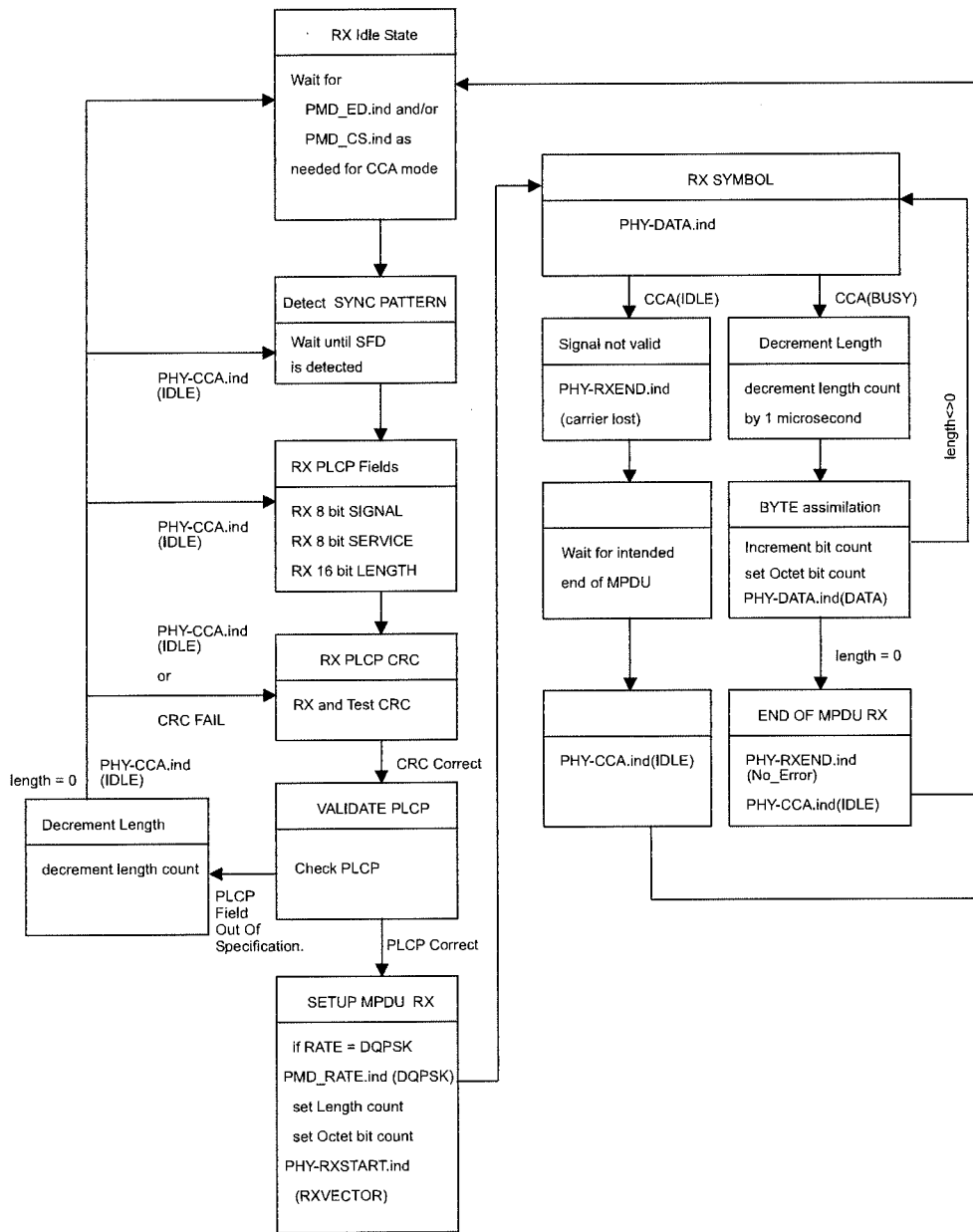


Figure 94—PLCP receive state machine

15.3 DSSS physical layer management entity (PLME)

15.3.1 PLME_SAP sublayer management primitives

Table 58 lists the MIB attributes that may be accessed by the PHY sublayer entities and intralayer of higher-layer management entities (LMEs). These attributes are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in Clause 10.

15.3.2 DSSS PHY MIB

All DSSS PHY MIB attributes are defined in Clause 12, with specific values defined in Table 58.

Table 58—MIB attribute default values/ranges

Managed object	Default value/range	Operational semantics
dot11PhyOperationComplianceGroup		
dot11PHYdot11TempType	DSSS-2.4 (02)	Static
dot11TempType	Implementation dependent	Static
dot11RegDomainsSupported	Implementation dependent	Static
dot11CurrentRegDomain	Implementation dependent	Static
dot11PhyRateGroup		
dot11SupportedDataRatesTx	X'02', X'04'	Static
dot11SupportedDataRatesRx	X'02', X'04'	Static
dot11PhyAntennaComplianceGroup		
dot11CurrentTxAntenna	Implementation dependent	Dynamic
dot11DiversitySupport	Implementation dependent	Static
dot11CurrentRxAntenna	Implementation dependent	Dynamic
dot11PhyTxPowerComplianceGroup		
dot11NumberSupportedPowerLevels	Implementation dependent	Static
dot11TxPowerLevel1	Implementation dependent	Static
dot11TxPowerLevel2	Implementation dependent	Static
dot11TxPowerLevel3	Implementation dependent	Static
dot11TxPowerLevel4	Implementation dependent	Static
dot11TxPowerLevel5	Implementation dependent	Static
dot11TxPowerLevel6	Implementation dependent	Static
dot11TxPowerLevel7	Implementation dependent	Static
dot11TxPowerLevel8	Implementation dependent	Static
dot11CurrentTxPowerLevel	Implementation dependent	Dynamic
dot11PhyDSSSComplianceGroup		
dot11CurrentChannel	Implementation dependent	Dynamic
dot11CCAModeSupported	Implementation dependent	Static
dot11CurrentCCAMode	Implementation dependent	Dynamic
dot11EDThreshold	Implementation dependent	Dynamic
dot11AntennasListGroup		
dot11SupportedTxAntenna	Implementation dependent	Static
dot11SupportedRxAntenna	Implementation dependent	Static
dot11DiversitySelectionRx	Implementation dependent	Dynamic
NOTE—The column titled “Operational semantics” contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. MIB attributes defined as dynamic can be modified by some management entities.		

15.3.3 DS PHY characteristics

The static DS PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 59. The definitions of these characteristics are in 10.4.3.

Table 59—DS PHY characteristics

Characteristic	Value
aSlotTime	20 μ s
aSIFSTime	10 μ s
aCCATime	≤ 15 μ s
aRxTxTurnaroundTime	≤ 5 μ s
aTxPLCPDelay	Implementors may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	Implementors may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	≤ 5 μ s
aTxRampOnTime	Implementors may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOffTime	Implementors may choose any value for this delay as long as the requirements of aSIFSTime are met.
aTxRFDelay	Implementors may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	Implementors may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	1 μ s
aMACProcessingDelay	0 (not applicable)
aPreambleLength	144 μ s
aPLCPHeaderLength	48 μ s
aMPDUDurationFactor	0
aMPDUMaxLength	$4 \leq x \leq (2^{13} - 1)$
aCWmin	31
aCWmax	1023

15.4 DSSS PMD sublayer

15.4.1 Scope and field of application

This subclause describes the PMD services provided to the PLCP for the DSSS PHY. Also defined in this subclause are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this standard. The relationship of this standard to the entire DSSS physical layer is shown in Figure 95.

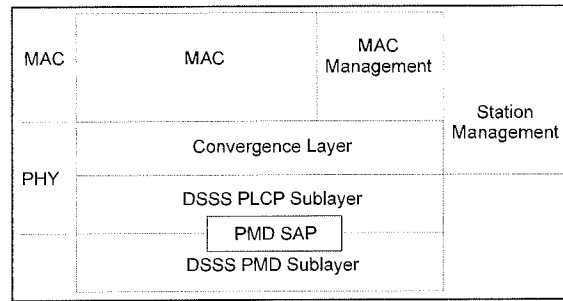


Figure 95—PMD layer reference model

15.4.2 Overview of service

The DSSS PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data shall be transmitted or received from the medium. The combined function of DSSS PMD sublayer primitives and parameters for the receive function results in a data stream, timing information, and associated received signal parameters being delivered to the PLCP sublayer. A similar functionality shall be provided for data transmission.

15.4.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the DSSS PMD fall into two basic categories:

- a) Service primitives that support PLCP peer-to-peer interactions, and
- b) Service primitives that have local significance and that support sublayer-to-sublayer interactions.

15.4.4 Basic service and options

All of the service primitives described in this clause are considered mandatory unless otherwise specified.

15.4.4.1 PMD_SAP peer-to-peer service primitives

Table 60 indicates the primitives for peer-to-peer interactions.

Table 60—PMD_SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PHY-RXSTART		X		—
PHY-RXEND		X		—
PHY-CCA		X		—
PHY-TXSTART	X		X	—
PHY-TXEND	X		X	—
PHY-DATA	X	X	X	—

15.4.4.2 PMD_SAP peer-to-peer service primitive parameters

Several service primitives include a parameter vector. This vector shall be a list of parameters that may vary depending on PHY type. Table 61 indicates the parameters required by the MAC or DSSS PHY in each of the parameter vectors used for peer-to-peer interactions.

Table 61—DSSS PMD_SAP peer-to-peer service primitives

Parameter	Associated primitive	Value
LENGTH	RXVECTOR, TXVECTOR	0 to $2^{13} - 1$
DATARATE	RXVECTOR, TXVECTOR	1, 2 Mbit/s
SERVICE	RXVECTOR, TXVECTOR	1, 2 Mbit/s
TXPWR_LEVEL	TXVECTOR	1, 2 Mbit/s
TX_ANTENNA	TXVECTOR	1, 2 Mbit/s
RSSI	RXVECTOR	1, 2 Mbit/s
SQ	RXVECTOR	1, 2 Mbit/s
RX_ANTENNA	RXVECTOR	1, 2 Mbit/s

15.4.4.3 PMD_SAP sublayer-to-sublayer service primitives

Table 62 indicates the primitives for sublayer-to-sublayer interactions.

Table 62—PMD_SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXSTART	X		—	—
PMD_TXEND	X		—	—
PMD_ANTSEL	X	X	—	—
PMD_TXPWRLVL	X		—	—
PMD_RATE	X	X	—	—
PMD_RSSI		X	—	—
PMD_SQ		X	—	—
PMD_CS		X	—	—
PMD_ED	X	X	—	—

15.4.4.4 PMD_SAP service primitive parameters

Table 63 indicates the parameters for the PMD primitives.

Table 63—List of parameters for the PMD primitives

Parameter	Associate primitive	Value
DATA	PHY-DATA.request PHY-DATA.indicate	Octet value: X'00'–X'FF'
TXVECTOR	PHY-DATA.request	A set of parameters
RXVECTOR	PHY-DATA.indicate	A set of parameters
TXD_UNIT	PMD_DATA.request	One(1), Zero(0): DBPSK dibit combinations 00,01,11,10: DQPSK
RXD_UNIT	PMD_DATA.indicate	One(1), Zero(0): DBPSK dibit combinations 00,01,11,10: DQPSK
RF_STATE	PMD_TXE.request	Receive, Transmit
ANT_STATE	PMD_ANTSEL.indicate PMD_ANTSEL.request	1 to 256
TXPWR_LEVEL	PHY-TXSTART	0, 1, 2, 3 (max of 4 levels)
RATE	PMD_RATE.indicate PMD_RATE.request	X'0A' for 1 Mbit/s DBPSK X'14' for 2 Mbit/s DQPSK
RSSI	PMD_RSSI.indicate	0–8 bits of RSSI
SQ	PMD_SQ.indicate	0–8 bits of SQ

15.4.5 PMD_SAP detailed service specification

The following subclauses describe the services provided by each PMD primitive.

15.4.5.1 PMD_DATA.request

15.4.5.1.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

15.4.5.1.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_DATA.request(TXD_UNIT)

The TXD_UNIT parameter takes on the value of either one(1) or zero(0) for DBPSK modulation or the dibit combination 00, 01, 11, or 10 for DQPSK modulation. This parameter represents a single block of data, which, in turn, shall be used by the PHY to be differentially encoded into a DBPSK or DQPSK transmitted symbol. The symbol itself shall be spread by the PN code prior to transmission.

15.4.5.1.3 When generated

This primitive shall be generated by the PLCP sublayer to request transmission of a symbol. The data clock for this primitive shall be supplied by the PMD layer based on the PN code repetition.

15.4.5.1.4 Effect of receipt

The PMD performs the differential encoding, PN code modulation, and transmission of the data.

15.4.5.2 PMD_DATA.indicate**15.4.5.2.1 Function**

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

15.4.5.2.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_DATA.indicate(RXD_UNIT)

The RXD_UNIT parameter takes on the value of one (1) or zero (0) for DBPSK modulation or as the dibit 00, 01, 11, or 10 for DQPSK modulation. This parameter represents a single symbol that has been demodulated by the PMD entity.

15.4.5.2.3 When generated

This primitive, which is generated by the PMD entity, forwards received data to the PLCP sublayer. The data clock for this primitive shall be supplied by the PMD layer based on the PN code repetition.

15.4.5.2.4 Effect of receipt

The PLCP sublayer either interprets the bit or bits that are recovered as part of the PLCP convergence procedure or passes the data to the MAC sublayer as part of the MPDU.

15.4.5.3 PMD_TXSTART.request**15.4.5.3.1 Function**

This primitive, which is generated by the PHY PLCP sublayer, initiates PPDU transmission by the PMD layer.

15.4.5.3.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_TXSTART.request

15.4.5.3.3 When generated

This primitive shall be generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-DATA.request primitive shall be provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

15.4.5.3.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

15.4.5.4 PMD_TXEND.request

15.4.5.4.1 Function

This primitive, which is generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

15.4.5.4.2 Semantics of the service primitive

The semantics of the primitive are as follows:

PMD_TXEND.request

15.4.5.4.3 When generated

This primitive shall be generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

15.4.5.4.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.

15.4.5.5 PMD_ANTSEL.request

15.4.5.5.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the antenna used by the PHY for transmission or reception (when diversity is disabled).

15.4.5.5.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_ANTSEL.request(ANT_STATE)

ANT_STATE selects which of the available antennas should be used for transmit. The number of available antennas shall be determined from the MIB table parameters aSuprtRxAntennas and aSuprtTxAntennas.

15.4.5.5.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific antenna for transmission or reception (when diversity is disabled).

15.4.5.5.4 Effect of receipt

PMD_ANTSEL immediately selects the antenna specified by ANT_STATE.

15.4.5.6 PMD_ANTSEL.indicate

15.4.5.6.1 Function

This primitive, which is generated by the PHY PLCP sublayer, reports the antenna used by the PHY for reception of the most recent packet.

15.4.5.6.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_ANTSEL.indicate(ANT_STATE)

ANT_STATE reports which of the available antennas was used for reception of the most recent packet.

15.4.5.6.3 When generated

This primitive shall be generated by the PLCP sublayer to report the antenna used for the most recent packet reception.

15.4.5.6.4 Effect of receipt

PMD_ANTSEL immediately reports the antenna specified by ANT_STATE.

15.4.5.7 PMD_TXPWRLVL.request

15.4.5.7.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

15.4.5.7.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_TXPWRLVL.request(TXPWR_LEVEL)

TXPWR_LEVEL selects which of the optional transmit power levels should be used for the current packet transmission. The number of available power levels shall be determined by the MIB parameter dot11NumberSupportedPowerLevels. Subclause 15.4.7.3 provides further information on the optional DSSS PHY power-level-control capabilities.

15.4.5.7.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific transmit power. This primitive shall be applied prior to setting PMD_TXSTART to the transmit state.

15.4.5.7.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level given by TXPWR_LEVEL.

15.4.5.8 PMD_RATE.request

15.4.5.8.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the modulation rate that shall be used by the DSSS PHY for transmission.

15.4.5.8.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_RATE.request(RATE)

The RATE parameter selects which of the DSSS PHY data rates shall be used for MPDU transmission. Subclause 15.4.6.4 provides further information on the DSSS PHY modulation rates. The DSSS PHY rate change capability is fully described in 15.2.

15.4.5.8.3 When generated

This primitive shall be generated by the PLCP sublayer to change or set the current DSSS PHY modulation rate used for the MPDU portion of a PPDU.

15.4.5.8.4 Effect of receipt

The receipt of PMD_RATE selects the rate that shall be used for all subsequent MPDU transmissions. This rate shall be used for transmission only. The DSSS PHY shall still be capable of receiving all the required DSSS PHY modulation rates.

15.4.5.9 PMD_RATE.indicate

15.4.5.9.1 Function

This primitive, which is generated by the PMD sublayer, indicates which modulation rate was used to receive the MPDU portion of the PPDU. The modulation shall be indicated in the PLCP Preamble IEEE 802.11 SIGNALING field.

15.4.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_RATE.indicate(RATE)

In receive mode, the RATE parameter informs the PLCP layer which of the DSSS PHY data rates was used to process the MPDU portion of the PPDU. Subclause 15.4.6.4 provides further information on the DSSS PHY modulation rates. The DSSS PHY rate change capability is fully described in 15.2.

15.4.5.9.3 When generated

This primitive shall be generated by the PMD sublayer when the PLCP Preamble IEEE 802.11 SIGNALING field has been properly detected.

15.4.5.9.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only.

15.4.5.10 PMD_RSSI.indicate

15.4.5.10.1 Function

This optional primitive, which is generated by the PMD sublayer, provides to the PLCP and MAC entity the received signal strength.

15.4.5.10.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_RSSI.indicate(RSSI)

The RSSI shall be a measure of the RF energy received by the DSSS PHY. RSSI indications of up to 8 bits (256 levels) are supported.

15.4.5.10.3 When generated

This primitive shall be generated by the PMD when the DSSS PHY is in the receive state. It shall be continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

15.4.5.10.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RSSI may be used in conjunction with signal quality (SQ) as part of a CCA scheme.

15.4.5.11 PMD_SQ.indicate

15.4.5.11.1 Function

This optional primitive, which is generated by the PMD sublayer, provides to the PLCP and MAC entity the SQ of the DSSS PHY PN code correlation. The SQ shall be sampled when the DSSS PHY achieves code lock and shall be held until the next code lock acquisition.

15.4.5.11.2 Semantics of the service primitive

The primitive shall provide the following parameter:

PMD_SQ.indicate(SQ)

The SQ shall be a measure of the PN code correlation quality received by the DSSS PHY. SQ indications of up to 8 bits (256 levels) are supported.

15.4.5.11.3 When generated

This primitive shall be generated by the PMD when the DSSS PHY is in the receive state and code lock is achieved. It shall be continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

15.4.5.11.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The SQ may be used in conjunction with RSSI as part of a CCA scheme.

15.4.5.12 PMD_CS.indicate**15.4.5.12.1 Function**

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the PN code and data is being demodulated.

15.4.5.12.2 Semantics of the service primitive

The PMD_CS (carrier sense) primitive in conjunction with PMD_ED provides CCA status through the PLCP layer PHYCCA primitive. PMD_CS indicates a binary status of ENABLED or DISABLED. PMD_CS shall be ENABLED when the correlator SQ indicated in PMD_SQ is greater than the CS_THRESHOLD parameter. PMD_CS shall be DISABLED when the PMD_SQ falls below the correlation threshold.

15.4.5.12.3 When generated

This primitive shall be generated by the PHY sublayer when the DSSS PHY is receiving a PPDU and the PN code has been acquired.

15.4.5.12.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PHYCCA indicator. This parameter shall indicate that the RF medium is busy and occupied by a DSSS PHY signal. The DSSS PHY should not be placed into the transmit state when PMD_CS is ENABLED.

15.4.5.13 PMD_ED.indicate**15.4.5.13.1 Function**

This optional primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has detected RF energy indicated by the PMD_RSSI primitive that is above a predefined threshold.

15.4.5.13.2 Semantics of the service primitive

The PMD_ED (energy detect) primitive, along with the PMD_SQ, provides CCA status at the PLCP layer through the PHYCCA primitive. PMD_ED indicates a binary status of ENABLED or DISABLED. PMD_ED shall be ENABLED when the RSSI indicated in PMD_RSSI is greater than the ED_THRESHOLD parameter. PMD_ED shall be DISABLED when the PMD_RSSI falls below the energy detect threshold.

15.4.5.13.3 When generated

This primitive shall be generated by the PHY sublayer when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter.

15.4.5.13.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PMD_ED indicator. This parameter shall indicate that the RF medium may be busy with an RF energy source that is not DSSS PHY compliant. If a DSSS PHY source is being received, the PMD_CS function shall be enabled shortly after the PMD_ED function is enabled.

15.4.5.14 PMD_ED.request**15.4.5.14.1 Function**

This optional primitive, which is generated by the PHY PLCP, sets the energy detect ED_THRESHOLD value.

15.4.5.14.2 Semantics of the service primitive

The primitive shall provide the following parameters:

PMD_ED.request(ED_THRESHOLD)

ED_THRESHOLD is the value that the RSSI indicated shall exceed for PMD_ED to be enabled.

15.4.5.14.3 When generated

This primitive shall be generated by the PLCP sublayer to change or set the current DSSS PHY energy detect threshold.

15.4.5.14.4 Effect of receipt

The receipt of PMD_ED immediately changes the energy detection threshold as set by the ED_THRESHOLD parameter.

15.4.5.15 PHY-CCA.indicate**15.4.5.15.1 Function**

This primitive, which is generated by the PMD, indicates to the PLCP layer that the receiver has detected RF energy that adheres to the CCA algorithm.

15.4.5.15.2 Semantics of the service primitive

The PHY-CCA primitive provides CCA status at the PLCP layer to the MAC.

15.4.5.15.3 When generated

This primitive shall be generated by the PHY sublayer when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter (PMD_ED is active), and optionally is a valid correlated DSSS PHY signal whereby PMD_CS would also be active.

15.4.5.15.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PHY-CCA indicator. This parameter indicates that the RF medium may be busy with an RF energy source that may or may not be DSSS PHY compliant. If a DSSS PHY source is being received, the PMD_CS function shall be enabled shortly after the PMD_ED function is enabled.

15.4.6 PMD operating specifications, general

The following subclauses provide general specifications for the DSSS PMD sublayer. These specifications apply to both the Receive and the Transmit functions and general operation of a DSSS PHY.

15.4.6.1 Operating frequency range

The DSSS PHY shall operate in the frequency range of 2.4 GHz to 2.4835 GHz as allocated by regulatory bodies in the USA and Europe or in the 2.471 GHz to 2.497 GHz frequency band as allocated by regulatory authority in Japan.

15.4.6.2 Number of operating channels

The channel center frequencies and CHNL_ID numbers shall be as shown in Table 64. The FCC (US), IC (Canada), and ETSI (Europe) specify operation from 2.4 GHz to 2.4835 GHz. For MKK (Japan), operation is specified as 2.471 GHz to 2.497 GHz. France allows operation from 2.4465 GHz to 2.4835 GHz, and Spain allows operation from 2.445 GHz to 2.475 GHz. For each supported regulatory domain, all channels in Table 64 marked with “X” shall be supported.

Table 64—DSSS PHY frequency channel plan

CHNL_ID	Frequency	Regulatory domains					
		X'10' FCC	X'20' IC	X'30' ETSI	X'31' Spain	X'32' France	X'40' MKK
1	2412 MHz	X	X	X	—	—	—
2	2417 MHz	X	X	X	—	—	—
3	2422 MHz	X	X	X	—	—	—
4	2427 MHz	X	X	X	—	—	—
5	2432 MHz	X	X	X	—	—	—
6	2437 MHz	X	X	X	—	—	—
7	2442 MHz	X	X	X	—	—	—
8	2447 MHz	X	X	X	—	—	—
9	2452 MHz	X	X	X	—	—	—
10	2457 MHz	X	X	X	X	X	—
11	2462 MHz	X	X	X	X	X	—
12	2467 MHz	—	—	X	—	X	—
13	2472 MHz	—	—	X	—	X	—
14	2484 MHz	—	—	—	—	—	X

In a multiple cell network topology, overlapping and/or adjacent cells using different channels can operate simultaneously without interference if the distance between the center frequencies is at least 30 MHz. Channel 14 shall be designated specifically for operation in Japan.

15.4.6.3 Spreading sequence

The following 11-chip Barker sequence shall be used as the PN code sequence:

+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1

The leftmost chip shall be output first in time. The first chip shall be aligned at the start of a transmitted symbol. The symbol duration shall be exactly 11 chips long.

15.4.6.4 Modulation and channel data rates

Two modulation formats and data rates are specified for the DSSS PHY: a *basic access rate* and an *enhanced access rate*. The basic access rate shall be based on 1 Mbit/s DBPSK modulation. The DBPSK encoder is specified in Table 65. The enhanced access rate shall be based on 2 Mbit/s DQPSK. The DQPSK encoder is specified in Table 66. (In the tables, $+j\omega$ shall be defined as counterclockwise rotation.)

Table 65—1 Mbit/s DBPSK encoding table

Bit input	Phase change ($+j\omega$)
0	0
1	π

Table 66—2 Mbit/s DQPSK encoding table

Dibit pattern (d0,d1) d0 is first in time	Phase change ($+j\omega$)
00	0
01	$\pi/2$
11	π
10	$3\pi/2$ ($-\pi/2$)

15.4.6.5 Transmit and receive in-band and out-of-band spurious emissions

The DSSS PHY shall conform with in-band and out-of-band spurious emissions as set by regulatory bodies. For the USA, refer to FCC 15.247, 15.205, and 15.209. For Europe, refer to ETS 300–328.

15.4.6.6 Transmit-to-receive turnaround time

The TX-to-RX turnaround time shall be less than 10 μs , including the power-down ramp specified in 15.4.7.7.

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol to valid CCA detection of the incoming signal. The CCA should occur within 25 μs (10 μs for turnaround time plus 15 μs for energy detect) or by the next slot boundary occurring after 25 μs has elapsed (refer to 15.4.8.4). A receiver input signal 3 dB above the ED threshold described in 15.4.8.4 shall be present at the receiver.

15.4.6.7 Receive-to-transmit turnaround time

The RX-to-TX turnaround time shall be measured at the MAC/PHY interface, using PHYTXSTART.request and shall be ≤ 5 μs . This includes the transmit power-up ramp described in 15.4.7.7.

15.4.6.8 Slot time

The slot time for the DSSS PHY shall be the sum of the RX-to-TX turnaround time (5 μ s) and the energy detect time (15 μ s specified in 15.4.8.4). The propagation delay shall be regarded as being included in the energy detect time.

15.4.6.9 Transmit and receive antenna port impedance

The impedance of the transmit and receive antenna port(s) shall be 50 Ω if the port is exposed.

15.4.6.10 Transmit and receive operating temperature range

Three temperature ranges for full operation compliance to the DSSS PHY are specified in Clause 13. Type 1 shall be defined as 0 $^{\circ}$ C to 40 $^{\circ}$ C, and is designated for office environments. Type 2 shall be defined as -20 $^{\circ}$ C to +50 $^{\circ}$ C, and Type 3 shall be defined as -30 $^{\circ}$ C to +70 $^{\circ}$ C. Types 2 and 3 are designated for industrial environments.

15.4.7 PMD transmit specifications

The following subclauses describe the transmit functions and parameters associated with the PMD sublayer.

15.4.7.1 Transmit power levels

The maximum allowable output power as measured in accordance with practices specified by the regulatory bodies is shown in Table 67. In the USA, the radiated emissions should also conform with the IEEE uncontrolled radiation emission standard (IEEE Std C95.1-1991).

Table 67—Transmit power levels

Maximum output power	Geographic location	Compliance document
1000 mW	USA	FCC 15.247
100 mW (EIRP)	Europe	ETS 300-328
10 mW/MHz	Japan	MPT ordinance for Regulating Radio Equipment, Article 49-20

15.4.7.2 Minimum transmitted power level

The minimum transmitted power shall be no less than 1 mW.

15.4.7.3 Transmit power level control

Power control shall be provided for transmitted power greater than 100 mW. A maximum of four power levels may be provided. At a minimum, a radio capable of transmission greater than 100 mW shall be capable of switching power back to 100 mW or less.

15.4.7.4 Transmit spectrum mask

The transmitted spectral products shall be less than -30 dBr (dB relative to the SINx/x peak) for $f_c - 22$ MHz $< f < f_c - 11$ MHz, $f_c + 11$ MHz $< f < f_c + 22$ MHz, -50 dBr for $f < f_c - 22$ MHz, and $f > f_c + 22$ MHz, where f_c

is the channel center frequency. The transmit spectral mask is shown in Figure 96. The measurements shall be made using 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

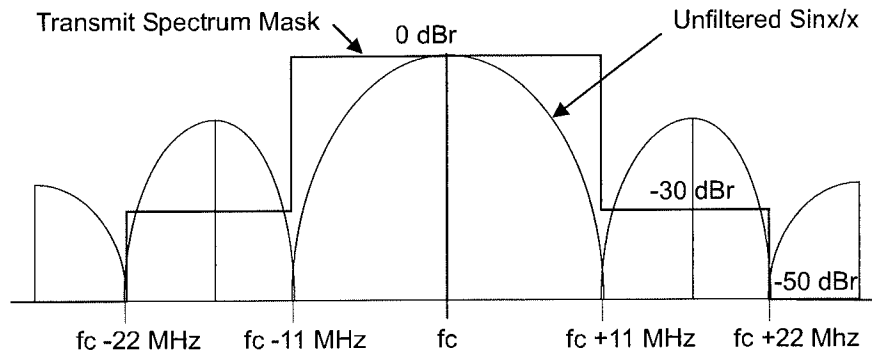


Figure 96—Transmit spectrum mask

15.4.7.5 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ± 25 ppm maximum.

15.4.7.6 Chip clock frequency tolerance

The PN code chip clock frequency tolerance shall be better than ± 25 ppm maximum.

15.4.7.7 Transmit power-on and power-down ramp

The transmit power-on ramp for 10% to 90% of maximum power shall be no greater than 2 μ s. The transmit power-on ramp is shown in Figure 97.

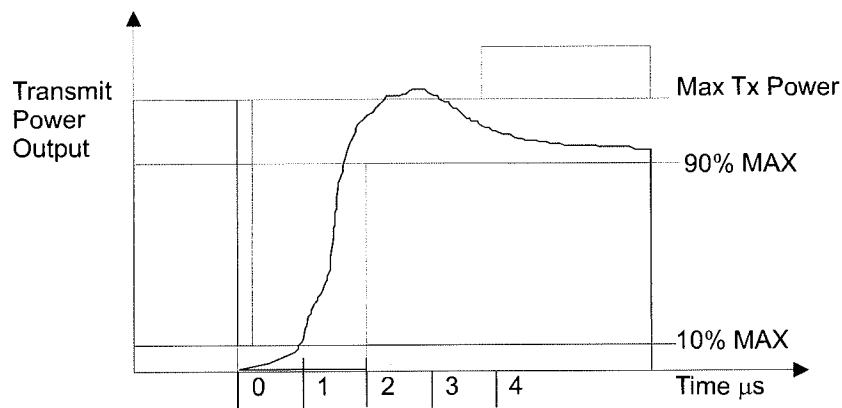


Figure 97—Transmit power-on ramp

The transmit power-down ramp for 90% to 10% maximum power shall be no greater than 2 μ s. The transmit power down ramp is shown in Figure 98.

The transmit power ramps shall be constructed such that the DSSS PHY emissions conform with the spurious frequency product specification defined in 15.4.6.5.

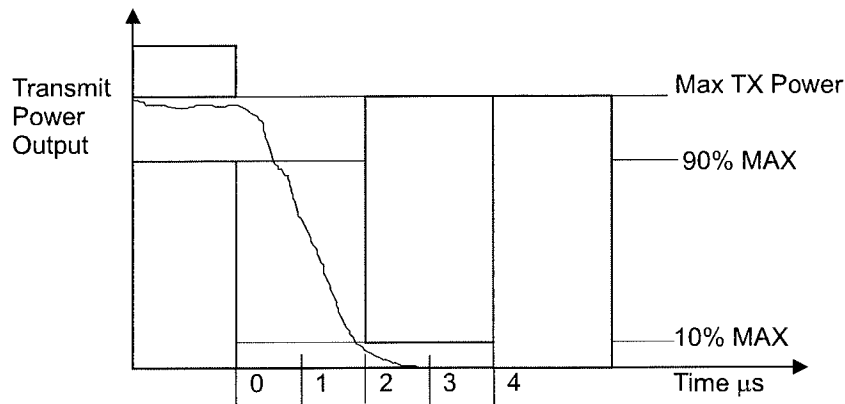


Figure 98—Transmit power-down ramp

15.4.7.8 RF carrier suppression

The RF carrier suppression, measured at the channel center frequency, shall be at least 15 dB below the peak $\text{SIN}(x)/x$ power spectrum. The RF carrier suppression shall be measured while transmitting a repetitive 01 data sequence with the scrambler disabled using DQPSK modulation. A 100 kHz resolution bandwidth shall be used to perform this measurement.

15.4.7.9 Transmit modulation accuracy

The transmit modulation accuracy requirement for the DSSS PHY shall be based on the difference between the actual transmitted waveform and the ideal signal waveform. Modulation accuracy shall be determined by measuring the peak vector error magnitude measured during each chip period. Worst-case vector error magnitude shall not exceed 0.35 for the normalized sampled chip data. The ideal complex I and Q constellation points associated with DQPSK modulation $(0.707, 0.707)$, $(0.707, -0.707)$, $(-0.707, 0.707)$, $(-0.707, -0.707)$ shall be used as the reference. These measurements shall be from baseband I and Q sampled data after recovery through a reference receiver system.

Figure 99 illustrates the ideal DQPSK constellation points and range of worst-case error specified for modulation accuracy.

Error vector measurement requires a reference receiver capable of carrier lock. All measurements shall be made under carrier lock conditions. The distortion induced in the constellation by the reference receiver shall be calibrated and measured. The test data error vectors described below shall be corrected to compensate for the reference receiver distortion.

The IEEE 802.11 vendor compatible radio shall provide an exposed TX chip clock, which shall be used to sample the I and Q outputs of the reference receiver.

The measurement shall be made under the conditions of continuous DQPSK transmission using scrambled all 1s.

The eye pattern of the I channel shall be used to determine the I and Q sampling point. The chip clock provided by the vendor radio shall be time delayed such that the samples fall at a 1/2 chip period offset from the mean of the zero crossing positions of the eye (see Figure 100). This is the ideal center of the eye and may not be the point of maximum eye opening.

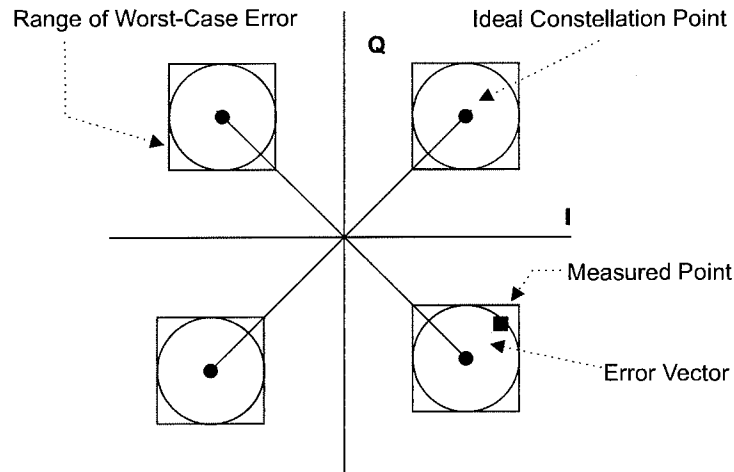


Figure 99—Modulation accuracy measurement example

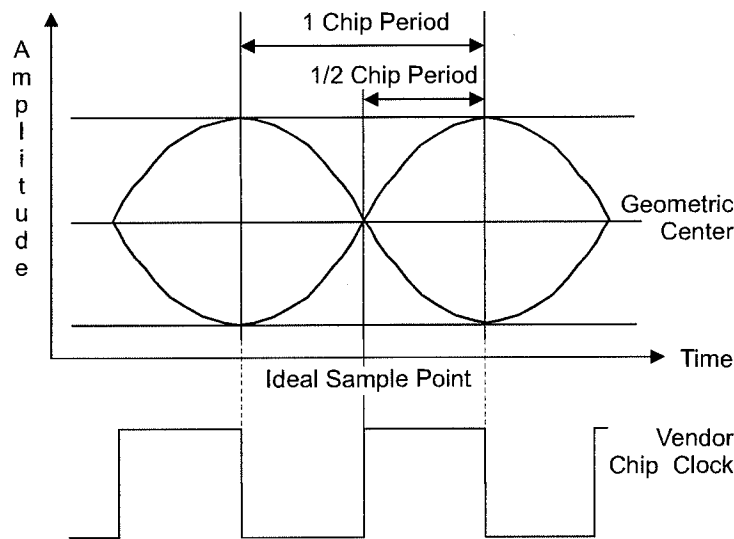


Figure 100—Chip clock alignment with baseband eye pattern

Using the aligned chip clock, 1000 samples of the *I* and *Q* baseband outputs from the reference receiver are captured. The vector error magnitudes shall be calculated as follows:

Calculate the dc offsets for *I* and *Q* samples.

$$I_{\text{mean}} = \sum_{n=1}^{1000} |I(n)| / 1000$$

$$Q_{\text{mean}} = \sum_{n=1}^{1000} |Q(n)| / 1000$$

Calculate the dc corrected I and Q samples for all $n=1000$ sample pairs.

$$I_{dc}(n) = I(n) - I_{\text{mean}}$$

$$Q_{dc}(n) = Q(n) - Q_{\text{mean}}$$

Calculate the average magnitude of I and Q samples.

$$I_{\text{mag}} = \sum_{n=1}^{1000} |I_{dc}(n)| / 1000$$

$$Q_{\text{mag}} = \sum_{n=1}^{1000} |Q_{dc}(n)| / 1000$$

Calculate the normalized error vector magnitude for the $I_{dc}(n)/Q_{dc}(n)$ pairs.

$$V_{\text{err}}(n) = \left[\frac{1}{2} \times (\{|I_{dc}(n)|/I_{\text{mag}}\}^2 + \{|Q_{dc}(n)|/Q_{\text{mag}}\}^2) \right]^{\frac{1}{2}} - V_{\text{correction}}$$

with $V_{\text{correction}}$ = error induced by the reference receiver system.

A vendor DSSS PHY implementation shall be compliant if for all $n=1000$ samples the following condition is met:

$$V_{\text{err}}(n) < 0.35$$

15.4.8 PMD receiver specifications

The following subclauses describe the receive functions and parameters associated with the PMD sublayer.

15.4.8.1 Receiver minimum input level sensitivity

The frame error ratio (FER) shall be less than 8×10^{-2} at an MPDU length of 1024 bytes for an input level of -80 dBm measured at the antenna connector. This FER shall be specified for 2 Mbit/s DQPSK modulation. The test for the minimum input level sensitivity shall be conducted with the energy detection threshold set ≤ -80 dBm.

15.4.8.2 Receiver maximum input level

The receiver shall provide a maximum FER of 8×10^{-2} at an MPDU length of 1024 bytes for a maximum input level of -4 dBm measured at the antenna. This FER shall be specified for 2 Mbit/s DQPSK modulation.

15.4.8.3 Receiver adjacent channel rejection

Adjacent channel rejection is defined between any two channels with ≥ 30 MHz separation in each channel group defined in 15.4.6.2.

The adjacent channel rejection shall be ≥ 35 dB with an FER of 8×10^{-2} using 2 Mbit/s DQPSK modulation described in 15.4.6.4 and an MPDU length of 1024 bytes.

The adjacent channel rejection shall be measured using the following method:

Input a 2 Mbit/s DQPSK modulated signal at a level 6 dB greater than specified in 15.4.8.1. In an adjacent channel (≥ 30 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion that adheres to the transmit mask specified in 15.4.7.4 to a level 41 dB above the level specified in 15.4.8.1. The adjacent channel signal shall be derived from a separate signal source. It cannot be a frequency shifted version of the reference channel. Under these conditions, the FER shall be no worse than 8×10^{-2} .

15.4.8.4 CCA

The DSSS PHY shall provide the capability to perform CCA according to at least one of the following three methods:

- *CCA Mode 1*: Energy above threshold. CCA shall report a busy medium upon detection of any energy above the ED threshold.
- *CCA Mode 2*: Carrier sense only. CCA shall report a busy medium only upon detection of a DSSS signal. This signal may be above or below the ED threshold.
- *CCA Mode 3*: Carrier sense with energy above threshold. CCA shall report a busy medium upon detection of a DSSS signal with energy above the ED threshold.

The energy detection status shall be given by the PMD primitive, PMD_ED. The carrier sense status shall be given by PMD_CS. The status of PMD_ED and PMD_CS is used in the PLCP convergence procedure to indicate activity to the MAC through the PHY interface primitive PHY-CCA.indicate.

A busy channel shall be indicated by PHY-CCA.indicate of class BUSY.

A clear channel shall be indicated by PHY-CCA.indicate of class IDLE.

The PHY MIB attribute dot11CCAModeSupported shall indicate the appropriate operation modes. The PHY shall be configured through the PHY MIB attribute dot11CurrentCCAMode.

The CCA shall be TRUE if there is no energy detect or carrier sense. The CCA parameters are subject to the following criteria:

- a) The energy detection threshold shall be ≤ -80 dBm for TX power > 100 mW, -76 dBm for 50 mW $<$ TX power ≤ 100 mW, and -70 dBm for TX power ≤ 50 mW.
- b) With a valid signal (according to the CCA mode of operation) present at the receiver antenna within $5 \mu\text{s}$ of the start of a MAC slot boundary, the CCA indicator shall report channel busy before the end of the slot time. This implies that the CCA signal is available as an exposed test point. Refer to Figure 47 for a definition of slot time boundary.
- c) In the event that a correct PLCP Header is received, the DSSS PHY shall hold the CCA signal inactive (channel busy) for the full duration as indicated by the PLCP LENGTH field. Should a loss of carrier sense occur in the middle of reception, the CCA shall indicate a busy medium for the intended duration of the transmitted packet.

Conformance to DSSS PHY CCA shall be demonstrated by applying a DSSS compliant signal, above the appropriate ED threshold a), such that all conditions described in b) and c) above are demonstrated.

16. Infrared (IR) PHY specification

16.1 Overview

The physical layer for the infrared system is specified in this clause. The IR PHY uses near-visible light in the 850 nm to 950 nm range for signaling. This is similar to the spectral usage of both common consumer devices such as infrared remote controls, as well as other data communications equipment, such as Infrared Data Association (IrDA) devices.

Unlike many other infrared devices, however, the IR PHY is not directed. That is, the receiver and transmitter do not have to be aimed at each other and do not need a clear line-of-sight. This permits the construction of a true LAN system, whereas with an aimed system, it would be difficult or impossible to install a LAN because of physical constraints.

A pair of conformant infrared devices would be able to communicate in a typical environment at a range up to about 10 m. This standard allows conformant devices to have more sensitive receivers, and this may increase range up to about 20 m.

The IR PHY relies on both reflected infrared energy as well as line-of-sight infrared energy for communications. Most designs anticipate that *all* of the energy at the receiver is reflected energy. This reliance on reflected infrared energy is called *diffuse infrared* transmission.

This standard specifies the transmitter and receiver in such a way that a conformant design will operate well in most environments where there is no line-of-sight path from the transmitter to the receiver. However, in an environment that has few or no reflecting surfaces, and where there is no line-of-sight, an IR PHY system may suffer reduced range.

The IR PHY will operate only in indoor environments. Infrared radiation does not pass through walls, and is significantly attenuated passing through most exterior windows. This characteristic can be used to “contain” an IR PHY in a single physical room, like a classroom or conference room. Different LANs using the IR PHY can operate in adjacent rooms separated only by a wall without interference, and without the possibility of eavesdropping.

At the time of this standard’s preparation, the only known regulatory standards that apply to the use of infrared radiation are safety regulations, such as IEC 60825-1: 1998 [B2] and ANSI Z136.1-1993 [B1]. While a conformant IR PHY device can be designed to also comply with these safety standards, conformance with this standard does not ensure conformance with other standards.

Worldwide, there are currently no frequency allocation or bandwidth allocation regulatory restrictions on infrared emissions.

Emitter (typically LED) and detector (typically PIN diode) devices for infrared communications are relatively inexpensive at the infrared wavelengths specified in the IR PHY, and at the electrical operating frequencies required by this PHY.

While many other devices in common use also use infrared emissions in the same optical band, these devices usually transmit infrared intermittently and do not interfere with the proper operation of a compliant IR PHY. If such a device does interfere, by transmitting continuously and with a very strong signal, it can be physically isolated (placing it in a different room) from the IEEE 802.11 LAN.

16.1.1 Scope

The PHY services provided to the IEEE 802.11 wireless LAN MAC by the IR system are described in this clause. The IR PHY layer consists of two protocol functions as follows:

- a) A physical layer convergence function, which adapts the capabilities of the physical medium dependent (PMD) system to the PHY service. This function is supported by the physical layer convergence procedure (PLCP), which defines a method of mapping the IEEE 802.11 MAC sublayer protocol data units (MPDU) into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.
- b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, the wireless medium (WM) between two or more STAs.

16.1.2 IR PHY functions

The IR PHY contains three functional entities: the PMD function, the physical layer convergence function, and the layer management function. Each of these functions is described in detail below.

The IR PHY service is provided to the MAC entity at the STA through a service access point (SAP) as described in Clause 12. For a visual guide to the relationship of the IR PHY to the remainder of the system, refer to Figure 11.

16.1.2.1 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a physical layer convergence sublayer is defined. This function simplifies the PHY service interface to the IEEE 802.11 MAC services. The PHY-specific preamble is normally associated with this convergence layer.

16.1.2.2 PMD sublayer

The PMD sublayer provides a clear channel assessment (CCA) mechanism, transmission mechanism, and reception mechanism that are used by the MAC via the PLCP to send or receive data between two or more STAs.

16.1.2.3 PHY management entity (PLME)

The PLME performs management of the local PHY functions in conjunction with the MAC management entity. Subclause 16.4 lists the MIB variables that may be accessed by the PHY sublayer entities and intra-layer of higher-layer management entities (LMEs). These variables are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in Clause 10.

16.1.3 Service specification method and notation

The models represented by figures and state diagrams are intended as illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the IEEE 802.11 IR PHY compliant developer. Conformance to this standard is not dependent on following the model, and an implementation that follows the model closely may not be conformant.

Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation. In particular, the PHY-SAP operations are defined and described as instantaneous; however, this may be difficult to achieve in an implementation.

16.2 IR PLCP sublayer

While the PLCP sublayer and the PMD sublayer are described separately, the separation and distinction between these sublayers is artificial, and is not meant to imply that the implementation must separate these functions. This distinction is made primarily to provide a point of reference from which to describe certain functional components and aspects of the PMD. The functions of the PLCP can be subsumed by a PMD sublayer; in this case, the PMD will incorporate the PHY-SAP as its interface, and will not offer a PMD-SAP.

16.2.1 Overview

A convergence procedure is provided by which MPDUs are converted to and from PLCPDUs. During transmission, the MPDU (PLCSDU) is prepended with a PLCP Preamble and PLCP Header to create the PLCPDU. At the receiver, the PLCP Preamble is processed and the internal data fields are processed to aid in demodulation and delivery of the MPDU (PSDU).

16.2.2 PLCP frame format

Figure 101 shows the format for the PLCPDU including the PLCP Preamble, the PLCP Header, and the PSDU. The PLCP Preamble contains the following fields: Synchronization (SYNC) and Start Frame Delimiter (SFD). The PLCP Header contains the following fields: Data Rate (DR), DC Level Adjustment (DCLA), Length (LENGTH), and Cyclic Redundancy Check (CRC). Each of these fields is described in detail in 16.2.4.

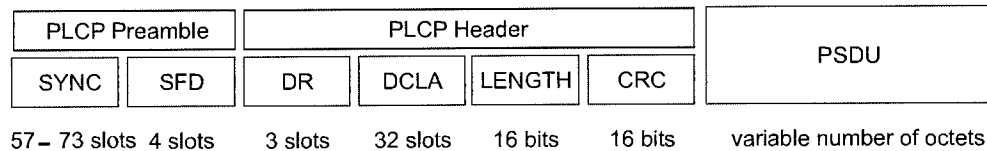


Figure 101—PLCPDU frame format

16.2.3 PLCP modulation and rate change

The PLCP Preamble shall be transmitted using the basic pulse defined in 16.3.3.2. The PLCSDU, LENGTH, and CRC fields shall be transmitted using pulse position modulation (PPM). PPM maps bits in the octet into symbols: 16-PPM maps four bits into a 16-position symbol, and 4-PPM maps two bits into a 4-position symbol. The basic L-PPM time unit is the slot. A slot corresponds to one of the L positions of a symbol and has a 250 ns duration. The PLCSDU, LENGTH, and CRC fields are transmitted at one of two bit rates: 1 Mbit/s or 2 Mbit/s. The Data Rate field indicates the data rate that will be used to transmit the PLCSDU, LENGTH, and CRC fields. The 1 Mbit/s data rate uses 16-PPM (basic access rate), and the 2 Mbit/s data rate uses 4-PPM (enhanced access rate). The transmitter and receiver will initiate the modulation or demodulation indicated by the DR field starting with the first 4 bits (in 16-PPM) or 2 bits (in 4-PPM) of the LENGTH field. The PSDU transmission rate is set by the DATARATE parameter in the PHY-TXSTART.request primitive. Any conformant IR PHY shall be capable of receiving at 1 Mbit/s and 2 Mbit/s. Transmission at 2 Mbit/s is optional.

A PHY-TXSTART.request that specifies a data rate that is not supported by a PHY instance will cause the PHY to indicate an error to its MAC instance. A PHY is not permitted under any circumstance to transmit at a different rate than the requested rate.

16.2.4 PLCP field definitions

16.2.4.1 PLCP Synchronization (SYNC) field

The SYNC field consists of a sequence of alternated presence and absence of a pulse in consecutive slots. The SYNC field has a minimum length of 57 L-PPM slots and a maximum length of 73 L-PPM slots and shall terminate with the absence of a pulse in the last slot. This field is provided so that the receiver can perform clock recovery (slot synchronization), automatic gain control (optional), signal-to-noise ratio estimation (optional), and diversity selection (optional).

The SYNC field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol. See 16.3.2.1 for legal symbols.

16.2.4.2 PLCP Start Frame Delimiter (SFD) field

The SFD field length is four L-PPM slots and consists of the binary sequence 1001, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot. The leftmost bit shall be transmitted first. The SFD field is provided to indicate the start of the PLCP Preamble and to perform bit and symbol synchronization.

The SFD field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

16.2.4.3 PLCP Data Rate (DR) field

The DR field indicates to the PHY the data rate that shall be used for the transmission or reception of the PLCSDU, LENGTH, and CRC fields. The transmitted value shall be provided by the PHY-TXSTART.request primitive as described in Clause 12. The DR field has a length of three L-PPM slots. The leftmost bit, as shown below, shall be transmitted first. The IR PHY currently supports two data rates defined by the slot pattern shown for the three L-PPM slots following the SFD, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot:

1 Mbit/s: 000
2 Mbit/s: 001

The DR field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

16.2.4.4 PLCP DC Level Adjustment (DCLA) field

The DCLA field is required to allow the receiver to stabilize the dc level after the SYNC, SFD, and DR fields. The leftmost bit, as shown below, shall be transmitted first. The length of the DCLA field is 32 L-PPM slots and consists of the contents shown, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot:

1 Mbit/s: 00000000100000000000000010000000
2 Mbit/s: 00100010001000100010001000100010

The DCLA field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

16.2.4.5 PLCP LENGTH field

The LENGTH field is an unsigned 16-bit integer that indicates the number of octets to be transmitted in the PSDU. The transmitted value shall be provided by the PHYTXSTART.request primitive as described in Clause 12. The lsb shall be transmitted first. This field is modulated and sent in L-PPM format. This field is protected by the CRC described in 16.2.4.6.

16.2.4.6 PLCP CRC field

The LENGTH field shall be protected by a 16-bit CRC-CCITT. The CRC-CCITT is the one's complement of the remainder generated by the modulo 2 division of the LENGTH field by the polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

The protected bits will be processed in transmit order. The msb of the 16-bit CRC-CCITT shall be transmitted first. This field shall be modulated and sent in L-PPM format. All CRC-CCITT calculations shall be made prior to L-PPM encoding on transmission and after L-PPM decoding on reception.

16.2.4.7 PSDU field

This field is composed of a variable number of octets. The minimum is 0 (zero) and the maximum is 2500. The lsb of each octet shall be transmitted first. All the octets of this field shall be modulated and sent in L-PPM format.

16.2.5 PLCP procedures**16.2.5.1 PLCP transmit procedure**

All commands issued by the MAC require that a confirmation primitive be issued by the PHY. The confirmation primitives provide flow control between the MAC and the PHY.

The transmit procedure is as follows:

- a) Based on the status of CCA, the MAC shall determine whether the channel is clear.
- b) If the channel is clear, transmission of the PSDU shall be initiated by a PHY-TXSTART.request with parameters LENGTH and DATARATE.
- c) The PHY entity shall immediately initiate transmission of the PLCP Preamble and PLCP Header based on the LENGTH and DATARATE parameters passed in the PHY-TXSTART.request. Once the PLCP Preamble and PLCP Header transmission is completed, the PHY entity shall issue a PHY-TXSTART.confirm.
- d) Each octet of the PSDU is passed from the MAC to the PHY by a single PHY-DATA.request primitive. Each PHY-DATA.request shall be confirmed by the PHY with a PHY-DATA.confirm before the next request can be made.
- e) At the PHY layer each PSDU octet shall be divided into symbols of 2 bits or 4 bits each. The symbols shall be modulated using L-PPM and transmitted into the medium.
- f) Transmission is terminated by the MAC through the primitive PHY-TXEND.request. The PHY shall confirm the resulting end of transmission with a PHY-TXEND.confirm.

16.2.5.2 PLCP receive procedure

The receive procedure is as follows:

- a) CCA is provided to the MAC via the PHY-CCA.indicate primitive. When the PHY senses activity on the medium, it shall indicate that the medium is busy with a PHY-CCA.indicate with a value of BUSY. This will normally occur during the SYNC field of the PLCP Preamble.
- b) The PHY entity shall begin searching for the SFD field. Once the SFD field is detected, the PHY entity shall attempt to receive the PLCP Header. After receiving the DR and DCLA fields, the PHY shall initiate processing of the received CRC and LENGTH fields. The data rate indicated in the DR field applies to all symbols in the latter part of the received PHYSDU, commencing with the first symbol of the LENGTH field. The CRC-CCITT shall be checked for correctness immediately after its reception.
- c) If the CRC-CCITT check fails, or the value received in the DR field is not one supported by the PHY, then a PHY-RXSTART.indicate shall not be issued to the MAC. When the medium is again free, the PHY shall issue a PHY-CCA.indicate with a value of IDLE.
- d) If the PLCP Preamble and PLCP Header reception is successful, the PHY shall send a PHY-RXSTART.indicate to the MAC; this includes the parameters DATARATE and LENGTH.
In the absence of errors, the receiving PHY shall report the same length to its local MAC, in the RXVECTOR parameter of the PHY-RXSTART.indicate primitive, that the peer MAC presented to its local PHY entity in the TXVECTOR parameter of its respective PHY-TXSTART.request.
- e) The received PLCSDU L-PPM symbols shall be assembled into octets and presented to the MAC using a series of PHY-DATA.indicate primitives, one per octet.
- f) Reception shall be terminated after the reception of the final symbol of the last PLCSDU octet indicated by the PLCP Header's LENGTH field. After the PHY-DATA.indicate for that octet is issued, the PHY shall issue a PHY-RXEND.indicate primitive to its MAC.
- g) After issuing the PHY-RXEND.indicate primitive, and when the medium is no longer busy, the PHY shall issue a PHY-CCA.indicate primitive with a value of IDLE.

16.2.5.3 CCA procedure

CCA is provided to the MAC via the PHY-CCA.indicate primitive.

The CCA procedure is as follows:

- a) When the PHY senses activity on the medium, a PHY-CCA.indicate primitive with a value of BUSY shall be issued. This will normally occur during reception of the SYNC field of the PLCP Preamble.
- b) When the PHY senses that the medium is free, a PHY-CCA.indicate primitive with a value of IDLE shall be issued.
- c) At any time, the MAC may issue a PHY-CCARESET.request primitive, which will reset the PHY's internal CCA detection mechanism to the medium not-busy (IDLE) state. This primitive will be acknowledged with a PHY-CCARESET.confirm primitive.

16.2.5.4 PMD_SAP peer-to-peer service primitive parameters

Several service primitives include a parameter vector. This vector shall be a list of parameters that may vary depending on PHY type. Table 68 indicates the parameters required by the MAC or IR PHY in each of the parameter vectors used for peer-to-peer interactions.

Table 68—IR PMD_SAP peer-to-peer service primitives

Parameter	Associated primitive	Value
LENGTH	RXVECTOR, TXVECTOR	4 to $2^{16} - 1$
DATARATE	RXVECTOR, TXVECTOR	PHY dependent

16.3 IR PMD sublayer

The IR PMD sublayer does not define PMD SAPs. The mechanism for communications between the PLCP and PMD sublayers, as well as the distinction between these two sublayers, if any, is left to implementors. In particular, it is possible to design and implement, in a conformant way, a single sublayer that subsumes the functions of both the PLCP and PMD, presenting only the PHY-SAP.

16.3.1 Overview

The PMD functional, electrical, and optical characteristics required for interoperability of implementations conforming to this specification are described in this subclause. The relationship of this specification to the entire IR physical layer is shown in Figure 11.

16.3.2 PMD operating specifications, general

General specifications for the IR PMD sublayer are provided in this subclause. These specifications apply to both the receive and transmit functions and general operation of a compliant IR PHY.

16.3.2.1 Modulation and channel data rates

Two modulation formats and data rates are specified for the IR PHY: a *basic access rate* and an *enhanced access rate*. The basic access rate is based on 1 Mbit/s 16-PPM modulation. The 16-PPM encoding is specified in Table 69. Each group of 4 data bits is mapped to one of the 16-PPM symbols. The enhanced access rate is based on 2 Mbit/s 4-PPM. The 4-PPM encoding is specified in Table 70. Each group of 2 data bits is mapped to one of the 4-PPM symbols. Transmission order of the symbol slots is from left to right, as shown in the table, where a 1 indicates in-band energy in the slot, and a 0 indicates the absence of in-band energy in the slot.

The data in Table 69 and Table 70 have been arranged (gray coded) so that a single out-of-position-by-one error in the medium, caused, for example, by intersymbol interference, results in only a single bit error in the received data, rather than in a multiple bit error.

Table 69—Sixteen-PPM basic rate mapping

Data	16-PPM symbol
0000	0000000000000001
0001	0000000000000010
0011	0000000000000100
0010	0000000000001000
0110	0000000000010000
0111	0000000000100000
0101	0000000001000000
0100	0000000010000000
1100	0000000100000000
1101	0000001000000000
1111	0000010000000000
1110	0000100000000000
1010	0001000000000000
1011	0010000000000000
1001	0100000000000000
1000	1000000000000000

Table 70—Four-PPM enhanced rate mapping

Data	4-PPM symbol
00	0001
01	0010
11	0100
10	1000

16.3.2.2 Octet partition and PPM symbol generation procedure

Since PPM is a block modulation method, with the block size less than a full octet, octets have to be partitioned prior to modulation (mapping into PPM symbols).

Octet partition depends on the PPM order being used.

Assume an octet is formed by eight bits numbered 7 6 5 4 3 2 1 0, where bit 0 is the lsb. Partition the octet as follows:

For 16-PPM, create two PPM symbols:

- The symbol using bits 3 2 1 0 shall be transmitted onto the medium first.
- The symbol using bits 7 6 5 4 shall be transmitted onto the medium last.

For 4-PPM, create four PPM symbols:

- The symbol using bits 1 0 shall be transmitted onto the medium first.
- The symbol using bits 3 2 shall be transmitted onto the medium second.
- The symbol using bits 5 4 shall be transmitted onto the medium third.
- The symbol using bits 7 6 shall be transmitted onto the medium last.

16.3.2.3 Operating environment

The IR PHY will operate only in indoor environments. IR PHY interfaces cannot be exposed to direct sunlight. The IR PHY relies on reflected infrared energy and does not require a line-of-sight between emitter and receiver in order to work properly. The range and bit error rate of the system may vary with the geometry of the environment and with natural and artificial illumination conditions.

16.3.2.4 Operating temperature range

The temperature range for full operation compliance with the IR PHY is specified as 0 °C to 40 °C.

16.3.3 PMD transmit specifications

The following subclauses describe the transmit functions and parameters associated with the PMD sublayer.

16.3.3.1 Transmitted peak optical power

The peak optical power of an emitted pulse shall be as specified in Table 71.

Table 71—Peak optical power as a function of emitter radiation pattern mask

Emitter radiation pattern mask	Peak optical power
Mask 1	2 W \pm 20%
Mask 2	0.55 W \pm 20%

16.3.3.2 Basic pulse shape and parameters

The basic pulse width, measured between the 50% amplitude points, shall be 250 ± 10 ns. The pulse rise time, measured between the 10% and 90% amplitude points, shall be no more than 40 ns. The pulse fall time, measured between the 10% and 90% amplitude points, shall be no more than 40 ns. The edge jitter, defined as the absolute deviation of the edge from its correct position, shall be no more than 10 ns. The basic pulse shape is shown in Figure 102.

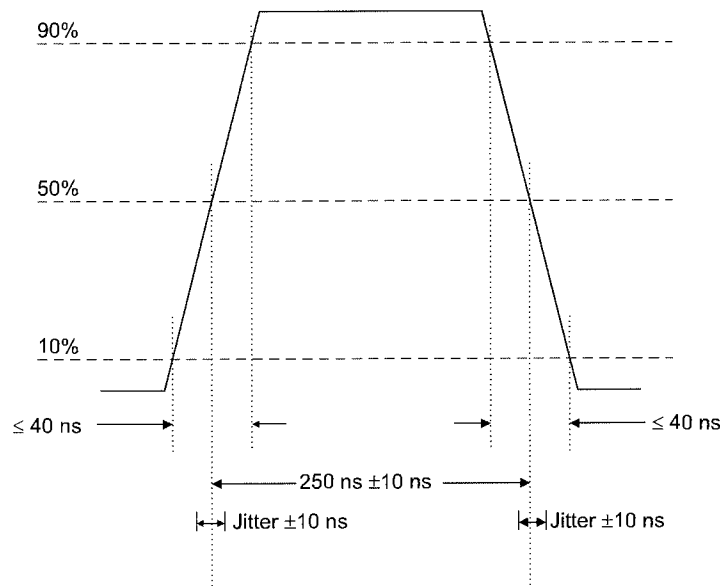


Figure 102—Basic pulse shape

16.3.3.3 Emitter radiation pattern mask

The standard contains two emitter radiation pattern masks. Mask 1 is defined in Table 72 and illustrated in Figure 103. Mask 2 is defined in Table 73 and illustrated in Figure 105.

Table 72—Definition of the emitter radiation pattern mask 1

Declination angle	Normalized irradiance
$\alpha \leq 60^\circ$	$> 3.5 \times 10^{-6}$
$\alpha \leq 29^\circ$	$\leq 2.2 \times 10^{-5}$
$29^\circ < \alpha \leq 43^\circ$	$\leq -1.06 \times 10^{-4} + (0.44 \times 10^{-5}) \alpha$
$43^\circ < \alpha \leq 57^\circ$	$\leq 1.15 \times 10^{-4} - (7.1 \times 10^{-7}) \alpha$
$57^\circ < \alpha \leq 74^\circ$	$\leq 2.98 \times 10^{-4} - (3.9 \times 10^{-6}) \alpha$
$74^\circ < \alpha \leq 90^\circ$	$\leq 4.05 \times 10^{-5} - (4.5 \times 10^{-7}) \alpha$

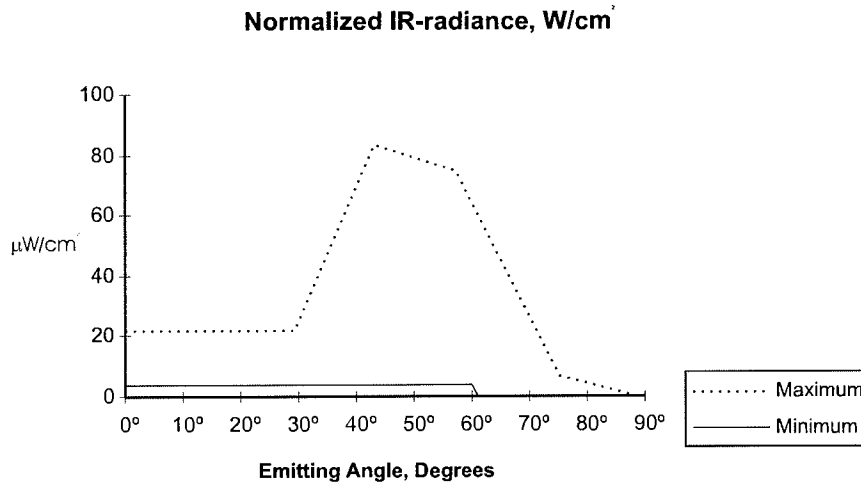


Figure 103—Emitter radiation pattern mask 1

Following is a description of how to interpret the Mask 1 table and figure. Position the conformant Mask 1 device in its recommended attitude. Define the conformant Mask 1 device axis as the axis passing through the emitter center and having the direction perpendicular to the floor. The mask represents the irradiance normalized to the total peak emitted power, as a function of the angle between the conformant Mask 1 device axis and the axis from the emitter center to the test receiver center (declination angle). The distance between emitter and test receiver is 1 m. The test receiver normal is always aimed at the emitter center. The azimuth angle is a rotation angle on the conformant device axis.

A device is conformant if for any azimuth angle its radiation pattern as a function of declination angle falls within the pattern mask.

Figure 104 is a description of how to interpret the Mask 2 table with reference to Figure 105.

Table 73—Definition of emitter radiation pattern mask 2

Declination angle	Pitch angle	Normalized irradiance
$\alpha \leq 60$	$\alpha = 0$	$0.05 \pm 15\%$
$\alpha \leq 90$	$\alpha = 0$	$0.025 \pm 15\%$
$\alpha \geq 100$	$\alpha = 0$	≤ 0.015
$0 \leq \alpha \leq 60$	$0 \leq \alpha \leq 10$	$0.035 \leq I \leq 0.055$
$0 \leq \alpha \leq 60$	$10 \leq \alpha \leq 20$	$0.0225 \leq I \leq 0.05$
$0 \leq \alpha \leq 60$	$\alpha \geq 30$	≤ 0.015

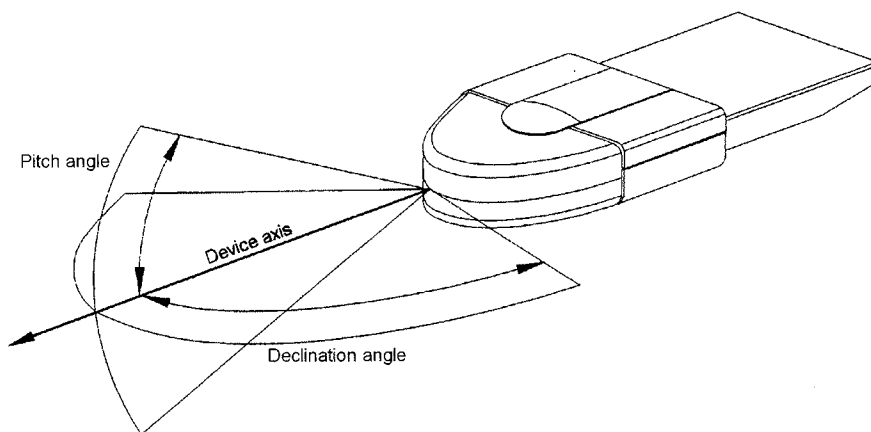


Figure 104—Mask 2 device orientation drawing

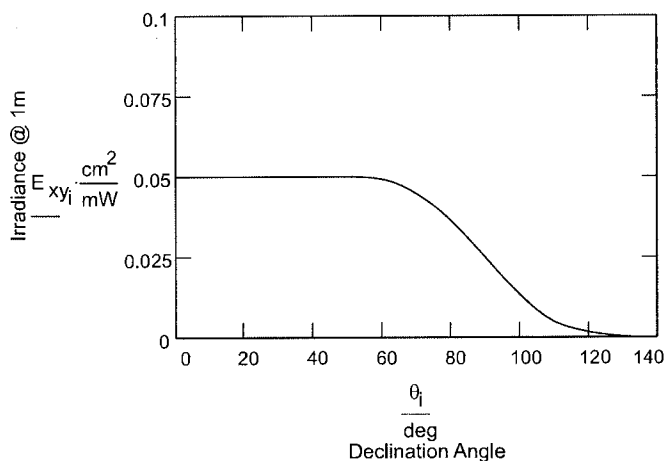


Figure 105—Emitter radiation pattern mask 2

Position the conformant Mask 2 device in its recommended attitude. Define the conformant Mask 2 device axis as passing through the emitter center and having the direction relative to the device as defined by the manufacturer. The declination angle plane is as defined by the manufacturer. The mask represents the irradiance normalized to the peak emitted power on the conformant Mask 2 device axis, as a function of the angle between the conformant device axis and the axis from the emitter center to the test receiver center (declination angle) in the declination plane. The distance between emitter and test receiver is 1 m. The test receiver normal is always aimed at the emitter center. The pitch angle is an angle relative to the conformant device axis which is perpendicular to the declination plane.

The device is conformant if, for a pitch angle of 0 degrees, at any declination angle from 0 to 100 degrees, and if, for any declination angle from 0 to 60 degrees, at any pitch angle from 0 to 20 degrees, its radiation pattern as a function of angle falls within the pattern mask.

Other radiation patterns are for future study.

16.3.3.4 Optical emitter peak wavelength

The optical emitter peak wavelength shall be between 850 nm and 950 nm.

16.3.3.5 Transmit spectrum mask

Define the transmit spectrum of a transmitter as the Fourier Transform, or equivalent, of a voltage (or current) signal whose amplitude, as a function of time, is proportional to the transmitted optical power.

The transmit spectrum of a conformant transmitter shall be 20 dB below its maximum for all frequencies above 15 MHz. The transmit spectrum mask is shown in Figure 106.

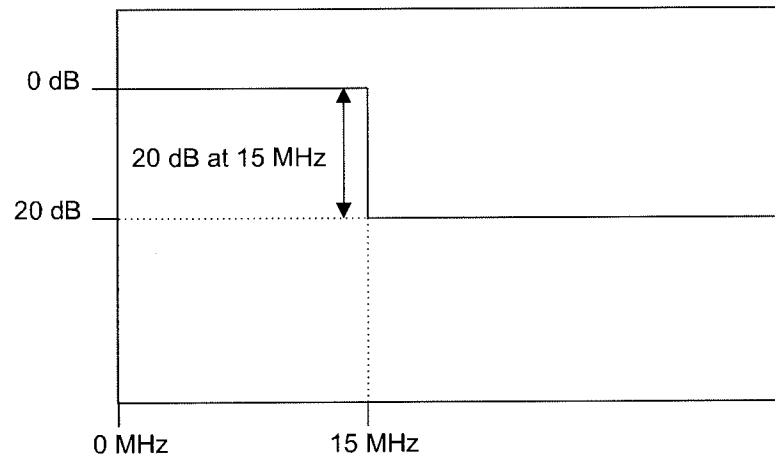


Figure 106—Transmit spectrum mask

16.3.4 PMD receiver specifications

The following subclauses describe the receive functions and parameters associated with the PMD sublayer.

16.3.4.1 Receiver sensitivity

The receiver sensitivity, defined as the minimum irradiance (in mW/cm²) at the photodetector plane required for a frame error ratio (FER) of 4×10^{-5} with a PLCSDU of 512 octets and with an unmodulated background IR source between 800 nm and 1000 nm with a level of 0.1 mW/cm², shall be

- 1 Mbit/s: 2×10^{-5} mW/cm²
- 2 Mbit/s: 8×10^{-5} mW/cm²

16.3.4.2 Receiver dynamic range

The receiver dynamic range, defined as the ratio between the maximum and minimum irradiance at the plane normal to the receiver axis that assures an FER lower than or equal to 4×10^{-5} with a PLCSDU of 512 octets and with an unmodulated background IR source between 800 nm and 1000 nm with a level of 0.1 mW/cm², shall be ≥ 30 dB.

16.3.4.3 Receiver field-of-view (FOV)

The receiver axis is defined as the direction of incidence of the optical signal at which the received optical power is maximum.

The received optical power shall be greater than the values given in Table 74, at the angles indicated, where “angle of incidence” is the angle of the optical signal relative to the receiver axis, and “received power” is the received optical power as a percentage of that measured at the receiver axis.

Table 74—Definition of the receiver field of view

Angle of incidence	Received power
$\alpha \leq 20^\circ$	$\geq 65\%$
$\alpha \leq 40^\circ$	$\geq 55\%$
$\alpha \leq 60^\circ$	$\geq 35\%$
$\alpha \leq 80^\circ$	$\geq 10\%$

16.3.5 Energy Detect, Carrier Sense, and CCA definitions

16.3.5.1 Energy Detect (ED) signal

The ED signal shall be set true when IR energy variations in the band between 1 MHz and 10 MHz exceed 0.001 mW/cm².

The ED shall operate independently of the CS. The ED shall not be asserted at the minimum signal level specified in 16.3.4.1, which is below the level specified in this subclause.

This signal is not directly available to the MAC.

16.3.5.2 Carrier Sense (CS) signal

The CS shall be asserted by the PHY when it detects and locks onto an incoming PLCP Preamble signal. Conforming PHYs shall assert this condition within the first 12 μ s of signal reception, at the minimum signal

level equal to the receiver sensitivity specified in 16.3.4.1, with a background IR level as specified in 16.3.4.1.

The CS shall be deasserted by the PHY when the receiving conformant device loses carrier lock.

NOTE—The 12 μ s specification is somewhat less than the minimum length of the PLCP SYNC interval, which is 14.25 μ s.

The CS shall operate independently of the ED and shall not require a prior ED before the acquisition and assertion of CS. This permits reception of signals at the minimum signal level specified in 16.3.4.1, even though these signals fall below the ED level.

This signal is not directly available to the MAC.

16.3.5.3 CCA

CCA shall be asserted “IDLE” by the PHY when the CS and the ED are both false, or when ED has been continuously asserted for a period of time defined by the product of `dot11CCAWatchdogTimerMax` and `dot11CCAWatchdogCountMax` without CS becoming active. When either CS or ED go true, CCA is indicated as “BUSY” to the MAC via the primitive `PHY-CCA.indicate`. CS and DE behavior are defined in 16.3.5.2.

Normally, CCA will be held “BUSY” throughout the period of the PLCP Header. After receiving the last PLCP bit and the first data octet, the PHY shall signal `PHY-RXSTART.indicate` with the parameters `LENGTH` and `RATE`. CCA shall be held “BUSY” until the number of octets specified in the decoded PLCP Header are received. At that time the PHY shall signal `PHY-RXEND.indicate`. The CCA may remain “BUSY” after the end of data if some form of energy is still being detected. The PHY will signal `PHY-CCA.indicate` with a value of `IDLE` only when the CCA goes “CLEAR.”

The transition of CCA from “BUSY” to “IDLE” is indicated to the MAC via the primitive `PHY-CCA.indicate`.

If CS and ED go false before the PHY signals `PHY-RXSTART.indicate`, CCA is set to “IDLE” and *immediately* signaled to the MAC via `PHY-CCA.indicate` with a value of `IDLE`. If CS and ED go false after the PHY has signaled `PHY-RXSTART.indicate`, implying that the PLCP Header has been properly decoded, then the PHY shall not signal a change in state of CCA until the proper interval has passed for the number of octets indicated by the received PLCP `LENGTH`. At that time, the PHY shall signal `PHY-RXEND.indicate` with an `RXERROR` parameter of `CarrierLost` followed by `PHY-CCA.indicate` with a value of `IDLE`.

The transition of CCA from “CLEAR” to “BUSY” resets the CCA watchdog timer and CCA watchdog counter. `dot11CCAWatchdogTimerMax` and `dot11CCAWatchdogCountMax` are parameters available via MIB entries and can be read and set via the LME.

Rise and fall times of CCA relative to the OR’ing of the CS and ED signals shall be less than 30 ns. CS and ED are both internal signals to the PHY and are not available directly to the MAC, nor are they defined at any exposed interface.

16.3.5.4 CHNL_ID

For the IR PHY, `CHNL_ID = X'01'` is defined as the baseband modulation method. All other values are not defined.

16.4 PHY attributes

PHY attributes have allowed values and default values that are PHY dependent. Table 75 and Table 76 describe those values, and further specify whether they are permitted to vary from implementation to implementation.

Table 75 does not provide the definition of the attributes, but only provides the IR PHY-specific values for the attributes whose definitions are in Clause 13.

Table 75—IR PHY MIB attributes

PHY MIB object	Default value	Operational semantics	Operational behavior
dot11CCAWatchdogTimerMax	Implementation dependent	Dynamic	A conformant PHY may set this via the LME
dot11CCAWatchdogCountMax	Implementation dependent	Dynamic	A conformant PHY may set this via the LME
dot11CCAWatchdogTimerMin	22 μ s	Static	Identical for all conformant PHYs
dot11CCAWatchdogCountMin	1	Static	Identical for all conformant PHYs
dot11SupportedDataRatesTx	Implementation dependent	Static	All conformant PHYs must include the value X'02' (1 Mbit/s).
dot11SupportedDataRatesRx	Implementation dependent	Static	All conformant PHYs must include the values X'02' (1 Mbit/s) and X'04' (2 Mbit/s).
dot11PhyType	03	Static	Identical for all conformant PHYs
dot11PhyTempType	X'01'	Static	Identical for all conformant PHYs

The static IR PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 76. The definitions of these characteristics are in 10.4.3.

Table 76—IR PHY characteristics

Characteristic	Value
aSlotTime	8 μ s
aSIFSTime	10 μ s
aCCATime	5 μ s
aRxTxTurnaroundTime	0 μ s
aTxPLCPDelay	Implementors may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	1 μ s
aRxTxSwitchTime	0 μ s
aTxRampOnTime	0 μ s
aTxRampOffTime	0 μ s
aTxRFDelay	Implementors may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	Implementors may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	1 μ s
aMACProcessingDelay	2 μ s
aPreambleLength	16 μ s (1 Mbit/s) 20 μ s (2 Mbit/s)
aPLCPHeaderLength	41 μ s (1 Mbit/s) 25 μ s (2 Mbit/s)
aMPDUDurationFactor	0
aMPDUMaxLength	2500
aCWmin	63
aCWmax	1023

Annex A

(normative)

Protocol Implementation Conformance Statement (PICS) proforma

A.1 Introduction

The supplier of a protocol implementation that is claimed to conform to ISO/IEC 8802.11: 1999 shall complete the following PICS proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

- a) By the protocol implementor, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- b) By the supplier and acquirer, or potential acquirer, of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- c) By the user, or potential user, of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS proformas);
- d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.2 Abbreviations and special symbols

A.2.1 Status symbols

M	mandatory
O	optional
O.<n>	optional, but support of at least one of the group of options labeled by the same numeral <n> is required
pred:	conditional symbol, including predicate identification

A.2.2 General abbreviations

N/A	not applicable
AD	address function capability
CF	implementation under test (IUT) configuration
FR	MAC frame capability
FS	frame sequence capability
PC	protocol capability
PICS	protocol implementation conformance statement

A.3 Instructions for completing the PICS proforma

A.3.1 General structure of the PICS proforma

The first part of the PICS proforma, Implementation identification and Protocol summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed questionnaire, divided into subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No) or by entering a value or a set or a range of values. (Note that there are some items where two or more choices from a set of possible answers may apply. All relevant choices are to be marked in these cases.)

Each item is identified by an item reference in the first column. The second column contains the question to be answered. The third column contains the reference or references to the material that specifies the item in the main body of ISO/IEC 8802-11: 1999. The remaining columns record the status of each item, i.e., whether support is mandatory, optional, or conditional, and provide the space for the answers (see also A.3.4). Marking an item as supported is to be interpreted as a statement that all relevant requirements of the subclauses and normative annexes, cited in the References column for the item, are met by the implementation.

A supplier may also provide, or be required to provide, further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labeled A<*I*> or X<*I*>, respectively, for cross-referencing purposes, where <*I*> is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format or presentation.

The PICS proforma for a station consists of A.4.1 through A.4.4 inclusive, and at least one of A.4.5, A.4.6, or A.4.7 corresponding to the PHY implemented.

A completed PICS proforma, including any Additional Information and Exception Information, is the PICS for the implementation in question.

NOTE—Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's capabilities, if this makes for easier and clearer presentation of the information.

A.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist in the interpretation of the PICS. It is not intended or expected that a large quantity of information will be supplied, and a PICS can be considered complete without any such information. Examples of such Additional Information might be an outline of the ways in which an (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this standard but have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

A.3.3 Exception information

It may happen occasionally that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer will be found in the Support column for this. Instead, the supplier shall write the missing answer into the Support column, together with an X</> reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception Information item itself.

An implementation for which an Exception Information item is required in this way does not conform to ISO/IEC 8802-11: 1999.

NOTE—A possible reason for the situation described above is that a defect in ISO/IEC 8802-11: 1999 has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.3.4 Conditional status

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply, mandatory or optional, are dependent upon whether or not certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the Not Applicable (N/A) answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form “<pred>:<S>”, where “<pred>” is a predicate as described below, and “<S>” is one of the status symbols M or O.

If the value of the predicate is true, the conditional item is applicable, and its status is given by S: the support column is to be completed in the usual way. Otherwise, the conditional item is not relevant and the N/A answer is to be marked.

A predicate is one of the following:

- a) An item-reference for an item in the PICS proforma: the value of the predicate is true if the item is marked as supported, and is false otherwise.
- b) A boolean expression constructed by combining item-references using the boolean operator OR: the value of the predicate is true if one or more of the items is marked as supported, and is false otherwise.

Each item referenced in a predicate, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

A.4 PICS proforma—ISO/IEC 8802-11: 1999⁷

A.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification, e.g., name(s) and version(s) of the machines and/or operating systems(s), system names	

NOTES

1—Only the first three items are required for all implementations. Other information may be completed as appropriate in meeting the requirement for full identification.

2—The terms Name and Version should be interpreted appropriately to correspond with a supplier’s terminology (e.g., Type, Series, Model).

A.4.2 Protocol summary, ISO/IEC 8802-11: 1999

Identification of protocol standard	ISO/IEC 8802-11: 1999
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	Amd. : Corr. : Amd. : Corr. :
Have any exception items been required? (See A.3.3; the answer Yes means that the implementation does not conform to ISO/IEC 8802-11: 1999.)	Yes <input type="checkbox"/> No <input type="checkbox"/>

Date of statement (dd/mm/yy)	
------------------------------	--

⁷Copyright release for PICS proforma: Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

A.4.3 IUT configuration

Item	IUT configuration	References	Status	Support
* CF1	What is the configuration of the IUT? Access Point (AP)	5.2	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/>
* CF2	Independent station (<i>not</i> an AP)	5.2	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/>
* CF3	Frequency-Hopping spread spectrum (FHSS) PHY for the 2.4 GHz band		O.2	Yes <input type="checkbox"/> No <input type="checkbox"/>
* CF4	Direct Sequence Spread Spectrum (DSSS) PHY for the 2.4 GHz band		O.2	Yes <input type="checkbox"/> No <input type="checkbox"/>
* CF5	Infrared PHY		O.2	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.4 MAC protocol

A.4.4.1 MAC protocol capabilities

Item	Protocol capability	References	Status	Support
PC1	Are the following MAC protocol capabilities supported? Authentication service	5.4.3.1, 5.4.3.2, 5.7.6, 5.7.7, 8.1, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC1.1	Authentication state	5.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC1.2	Open System authentication	8.1.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC1.3	Shared Key authentication	8.1.2, 8.3	PC2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* PC2	WEP algorithm	5.4.3.3, 8.2, Annex C	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC2.1	WEP Encryption procedure	8.2.3, 8.2.4, 8.2.5	PC2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC2.2	WEP Decryption procedure	8.2.3, 8.2.4, 8.2.5	PC2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC2.3	Security services management	8.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3	Distributed Coordination function	9.1, 9.2, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.1	Net Allocation Vector (NAV) function	9.2.1, 9.2.5, 9.3.2.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.2	Interframe space usage and timing	9.2.3, 9.2.5, 9.2.10	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.3	Random Backoff function	9.2.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.4	DCF Access procedure	9.2.5.1, 9.2.5.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.5	Random Backoff procedure	9.2.5.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.6	Recovery procedures and retransmit limits	9.2.5.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.7	RTS/CTS procedure	9.2.5.4, 9.2.5.6, 9.2.5.7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.4.1 MAC protocol capabilities (continued)

Item	Protocol capability	References	Status	Support
PC3.8	Directed MPDU transfer	9.2.6	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.9	Broadcast and multicast MPDU transfer	9.2.7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.10	MAC level acknowledgment	9.2.2, 9.2.8	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC3.11	Duplicate detection and recovery	9.2.9	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
* PC4	Point coordinator (PC)	9.1, 9.3, Annex C	CF1:O	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC4.1	Maintenance of CFP structure and timing	9.3.1, 9.3.2	PC4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC4.2	PCF MPDU transfer from PC	9.3.3	PC4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* PC4.3	PCF MPDU transfer to PC	9.3.3	PC4:O	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC4.4	Overlapping PC provisions	9.3.3.2	PC4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC4.5	Polling list maintenance	9.3.4	PC4.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* PC5	CF-Pollable	9.1, 9.3, Annex C	CF2:O	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC5.1	Interpretation of CFP structure and timing	9.3.1, 9.3.2	PC5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC5.2	PCF MPDU transfer to/from and CF-Pollable STA	9.3.3	PC5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC5.3	Polling list update	9.3.4	PC5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC6	Fragmentation	9.2, 9.4, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC7	Defragmentation	9.2, 9.5, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC8	MAC data service	9.1.5, 9.8, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC8.1	Reorderable-Multicast service class	9.8	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC8.2	StrictlyOrdered service class	9.8	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC9	Multirate support	9.6, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
* PC10	Multiple outstanding MSDU support	9.8, Annex C	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC10.1	Multiple outstanding MSDU transmission restrictions	9.8	PC10:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC11	Timing synchronization	11.1, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC11.1	Timing in an infrastructure network	11.1.1.1, 11.1.4	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC11.2	Timing in an Independent BSS (IBSS)	11.1.1.2, 11.1.4	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC11.3	Beacon Generation function	11.1.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC11.5	TSF synchronization and accuracy	11.1.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC11.5	Infrastructure BSS initialization	11.1.3	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>

A.4.4.1 MAC protocol capabilities (continued)

Item	Protocol capability	References	Status	Support
PC11.6	Independent BSS initialization	11.1.3	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC11.7	Passive scanning	11.1.3	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC11.8	Active scanning	11.1.3	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC11.9	Probe response	11.1.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC11.10	Hop Synchronization function	11.1.5	CF3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC12	Infrastructure power management	11.2.1, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC12.1	Station power management modes	11.2.1.1, 11.2.1.8	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC12.2	TIM transmission	11.2.1.2, 11.2.1.3	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC12.3	AP function during CP	11.2.1.4	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC12.4	AP function during CFP	11.2.1.5	PC4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC12.5	Receive function during CP	11.2.1.6	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC12.6	Receive function during CFP	11.2.1.7	PC5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC12.7	Aging function	11.2.1.9	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC13	IBSS power management	11.2.2, Annex C	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC13.1	Initialization of power management	11.2.2.2	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC13.2	STA power state transitions	11.2.2.3	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC13.3	ATIM and frame transmission	11.2.2.4	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC14	Association and reassociation	5.4, 5.7, 11.3, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC14.1	Association state	5.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC14.2	STA association procedure	11.3.1	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC14.3	AP association procedure	11.3.2	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC14.4	STA reassociation procedure	11.3.3	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC14.5	AP reassociation procedure	11.3.4	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC15	Management information base (MIB)	Annex D	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC15.1	dot11SMtbase, dot11SmtAuthenticationAlgorithms	Annex D	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
* PC15.2	dot11SMTprivacy	Annex D	PC2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
PC15.3	dot11MACbase, dot11CountersGroup, dot11MacGroupAddresses	Annex D	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
* PC15.4	dot11MACStatistics	Annex D	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
PC15.5	dot11ResourceTypeID	Annex D	M	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.4.2 MAC frames

Item	MAC frame	References	Status	Support
	Is transmission of the following MAC frames supported?	7, Annex C		
FT1	Association request	7	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT2	Association response	7	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT3	Reassociation request	7	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT4	Reassociation response	7	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT5	Probe request	7	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT6	Probe response	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT7	Beacon	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT8	ATIM	7	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT9	Disassociation	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT10	Authentication	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT11	Deauthentication	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT12	PS-Poll	7	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT13	RTS	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT14	CTS	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT15	ACK	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT16	CF-End	7	PC4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT17	CF End+CF-Ack	7	PC4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT18	Data	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT19	Data + CF-Ack	7	(PC4 OR PC5):M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT20	Data + CF-Poll	7	PC4.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT21	Data + CF-Ack+CF-Poll	7	PC4.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT22	Null	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FT23	CF-Ack (no data)	7	(PC4 OR PC5):M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT24	CF-Poll (no data)	7	PC4.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FT25	CF-Ack+CF-Poll (no data)	7	PC4.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	Is reception of the following MAC frames supported?	7, Annex C		
FR1	Association request	7	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR2	Association response	7	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR3	Reassociation request	7	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR4	Reassociation response	7	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR5	Probe request	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR6	Probe response	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR7	Beacon	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR8	ATIM	7	CF2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR9	Disassociation	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR10	Authentication	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.4.2 MAC frames (continued)

Item	MAC frame	References	Status	Support
FR11	Deauthentication	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR12	PS-Poll	7	CF1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR13	RTS	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR14	CTS	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR15	ACK	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR16	CF-End	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR17	CF End+CF-Ack	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR18	Data	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR19	Data + CF-Ack	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR20	Data + CF-Poll	7	PC5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR21	Data + CF-Ack+CF-Poll	7	PC5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR22	Null	7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FR23	CF-Ack (no data)	7	(PC4 OR PC5):M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR24	CF-Poll (no data)	7	PC5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FR25	CF-Ack+CF-Poll (no data)	7	PC5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>

A.4.4.3 Frame exchange sequences

Item	Frame exchange sequence	References	Status	Support
	Are the following frame sequences supported?			
FS1	Basic frame sequences	9.7, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FS2	CF-Frame sequences	9.7, Annex C	(PC4 OR PC5):M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>

A.4.4.4 MAC addressing functions

Item	MAC Address function	References	Status	Support
	Are the following MAC Addressing functions supported?			
AD1	STA universal individual IEEE 802 address	5.3.3, 7.1.3.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
AD2	BSS identifier generation	7.1.3.3, 11.1.3, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
AD3	Receive address matching	7.1.3.3, 7.2.2, Annex C	M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>

A.4.5 Frequency-Hopping PHY functions

Item	Protocol feature	References	Status	Support
FH1	Which requirements and options does the PHY support? PHY service primitive parameters			
FH1.1	TXVECTOR parameter: LENGTH	14.2.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH1.2	TXVECTOR parameter: PLCPBITRATE	14.2.2.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH1.2.1	PLCPBITRATE = X'00' (1.0 Mbit/s)	14.2.2.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
* FH1.2.2	PLCPBITRATE = X'02' (2.0 Mbit/s)	14.2.2.2	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH1.3	RXVECTOR parameter: LENGTH	14.2.3.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH1.4	RXVECTOR parameter: RSSI	14.2.3.2	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH2	PLCP frame format			
FH2.1	PLCP Preamble: Sync	14.3.2.1.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH2.2	PLCP Preamble: Start Frame Delimiter	14.3.2.1.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH2.3	PLCP Header: Length Word	14.3.2.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH2.4	PLCP Header: Signaling field	14.3.2.2.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH2.5	PLCP Header: Header Error Check	14.3.2.2.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH2.6	PLCP Data Whitener: Scrambling and bias suppression encoding	14.3.2.3, 14.3.3.1.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH3	PLCP Transmit procedure			
FH3.1	Transmit: transmit on MAC request	14.3.3.1.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH3.2	Transmit: format and whiten frame	14.3.3.1.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH3.3	Transmit: Timing	14.3.3.1.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH4	PLCP CS/CCA procedure			
FH4.1	CS/CCA: perform on a minimum of one antenna	14.3.3.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH4.2.	CS/CCA: Detect preamble starting up to 20 μ s after start of slot time	14.3.3.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH4.3	CS/CCA: Detect preamble starting at least 16 μ s prior to end of slot time	14.3.3.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH4.4	CS/CCA: Detect random data	14.3.3.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH4.5	CS/CCA: Perform on antenna with essentially same gain and pattern as transmit antenna	14.3.3.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH4.6	CS/CCA: Detect valid SFD and PLCP header	14.3.3.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH4.7	CS/CCA: Maintain BUSY indication until end of length contained in valid PLCP header	14.3.3.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH5	PLCP Receive procedure			
FH5.1	Receive: Receive and dewhiten frame	14.3.3.3.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH6	PHY LME			
FH6.1	PLME: Support FH sync	14.4.2.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH6.2	PLME: Support PLME primitives	14.4.3.2	O	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.5 Frequency-Hopping PHY functions (continued)

Item	Protocol feature	References	Status	Support
FH7	Geographic area specific requirements			
* FH7.1	Geographic areas			
FH7.1.1	North America	14.6.2	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.1.2	Most of Europe	14.6.2	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.1.3	Japan	14.6.2	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.1.4	Spain	14.6.2	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.1.5	France	14.6.2	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.2	Operating frequency range	14.6.3	FH7.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.3	Number of operating channels	14.6.4	FH7.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.4	Operating channel frequencies	14.6.5	FH7.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.5	Occupied channel bandwidth	14.6.6	FH7.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.6	Minimum hop rate	14.6.7	FH7.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.7	Hop sequences	14.6.8	FH7.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH7.8	Unwanted emissions	14.6.9	FH7.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8	1 Mbit/s PMD			
FH8.1	Modulation 2GFSK, BT=0.5, 1=positive frequency deviation, 0=negative frequency deviation	14.6.10	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.2	Peak frequency deviation	14.6.10	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.3	Zero-Crossing error	14.6.10	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.4	Nominal channel data rate	14.6.11	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.5	Channel switching/settling time	14.6.12	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.6	Receive to transmit switch time	14.6.13	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.7	Nominal transmit power	14.6.14.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.8	Transmit power levels	14.6.14.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.9	Transmit power level control to <100 mW	14.6.14.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.10	Transmit spectrum shape	14.6.14.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.11	Transmit center frequency tolerance	14.6.14.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.12	Transmitter ramp periods	14.6.14.6	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.13	Receiver input dynamic range	14.6.15.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.14	Receiver center frequency acceptance range	14.6.15.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.15	Clear channel assessment power threshold for a probability of detection of 90% (preamble)/70% (random data) for 100 mW units	14.6.15.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.16	Clear channel assessment power threshold for units >100 mW; sensitivity threshold is 1/2 dB lower for every dB above 20 dBm	14.6.15.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.17	Minimum receiver sensitivity at FER=3% with 400 octet frames	14.6.15.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.18	Intermodulation protection	14.6.15.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.5 Frequency-Hopping PHY functions (continued)

Item	Protocol feature	References	Status	Support
FH8.19	Desensitization	14.6.15.6	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.20	Operating temperature range	14.6.16	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.20.1	Temperature type 1	14.6.16	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.20.2	Temperature type 2	14.6.16	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH8.20.3	Temperature type 3	14.6.16	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH9	2 Mbit/s PMD			
FH9.1	All 1M PMD requirements	14.7.1	FH1.2.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FH9.2	Modulation 4GFSK, BT=0.5	14.7.2	FH1.2.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FH9.3	Frame structure for 2M PHY	14.7.2.1	FH1.2.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FH9.4	Nominal channel data rate	14.7.3	FH1.2.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FH9.5	Input dynamic range	14.7.4	FH1.2.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FH9.6	Minimum receiver sensitivity at FER=3% with 400 octet frames	14.7.5	FH1.2.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FH9.7	Intermodulation protection	14.7.6	FH1.2.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FH9.8	Desensitization	14.7.7	FH1.2.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
FH10	MIB	13.1, 14.8, Annex D	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
FH10.1	dot11PhyFHSSComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup	13.1,14.8	M	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.6 Direct sequence PHY functions

Item	PHY feature	References	Status	Support
DS1	PLCP sublayer procedures	15.2		
	Preamble prepend on TX	15.2.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS1.1	PLCP frame format	15.2.2, 15.2.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS1.2	PLCP integrity check generation	15.2.3, 15.2.3.6	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS1.3	TX rate change capability	15.2.3.3, 15.2.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS1.4	Supported data rates	15.1, 15.2.3.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS1.5	Data whitener scrambler	15.2.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS1.6	Scrambler initialization	15.2.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS2	Preamble process on RX	15.2.1		
DS2.1	PLCP frame format	15.2.2, 15.2.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS2.2	PLCP integrity check verify	15.2.3, 15.2.3.6	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS2.3	RX Rate change capability	15.2.3.3, 15.2.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS2.4	Data whitener descrambler	15.2.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS3	PN code sequence	15.4.6.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS4	Chipping continue on power down	15.2.6	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
*DS5	Operating channel capability	15.2.6, 15.4.6.2		

A.4.6 Direct sequence PHY functions (continued)

Item	PHY feature	References	Status	Support
* DS5.1	North America (FCC)	15.2.6, 15.4.6.2	DS5:O.1	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.1	channel 1	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.2	channel 2	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.3	channel 3	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.4	channel 4	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.5	channel 5	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.6	channel 6	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.7	channel 7	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.8	channel 8	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.9	channel 9	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.10	channel 10	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.1.11	channel 11	15.2.6, 15.4.6.2	DS5.1:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* DS5.2	Canada (IC)	15.2.6, 15.4.6.2	DS5:O.1	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.1	channel 1	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.2	channel 2	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.3	channel 3	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.4	channel 4	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.5	channel 5	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.6	channel 6	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.7	channel 7	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.8	channel 8	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.9	channel 9	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.10	channel 10	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.2.11	channel 11	15.2.6, 15.4.6.2	DS5.2:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* DS5.3	Europe (ETSI)	15.2.6, 15.4.6.2	DS5:O.1	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.1	channel 1	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.2	channel 2	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.3	channel 3	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.4	channel 4	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.5	channel 5	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.6	channel 6	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.7	channel 7	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.8	channel 8	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.9	channel 9	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.10	channel 10	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.11	channel 11	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.12	channel 12	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.3.13	channel 13	15.2.6, 15.4.6.2	DS5.3:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* DS5.4	France	15.2.6, 15.4.6.2	DS5:O.1	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.4.1	channel 10	15.2.6, 15.4.6.2	DS5.4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>

A.4.6 Direct sequence PHY functions (continued)

Item	PHY feature	References	Status	Support
DS5.4.2	channel 11	15.2.6, 15.4.6.2	DS5.4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.4.3	channel 12	15.2.6, 15.4.6.2	DS5.4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.4.4	channel 13	15.2.6, 15.4.6.2	DS5.4:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* DS5.5	Spain	15.2.6, 15.4.6.2	DS5:O.1	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.5.1	channel 10	15.2.6, 15.4.6.2	DS5.5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS5.5.2	channel 11	15.2.6, 15.4.6.2	DS5.5:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* DS5.6	Japan (RCR)	15.2.6, 15.4.6.2	DS5:O.1	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS6	Bits to symbol mapping	15.4.6.4		
DS6.1	1 Mbit/s	15.4.6.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS6.2	2 Mbit/s	15.4.6.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
*DS7	CCA functionality	15.4.8.4		
DS7.1	Energy Only (RSSI above threshold)	15.4.8.4	DS7:O.2	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS7.2	IEEE 802.11 DSSS correlation	15.4.8.4	DS7:O.2	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS7.3	Both methods	15.4.8.4	DS7:O.2	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS7.4	Hold CCA busy for packet duration of a correctly received PLCP but carrier lost during reception of MPDU	15.2.7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS7.5	Hold CCA busy for packet duration of a correctly received but out of specification PLCP	15.2.7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS8	Transmit antenna selection	15.4.5.5, 15.4.5.6	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS9	Receive antenna diversity	15.4.5.5, 15.4.5.6, 15.4.5.7	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
*DS10	Antenna port(s) availability	15.4.6.9	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS10.1	50 Ω impedance	15.4.6.9	DS10:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
*DS11	Transmit power level support	15.4.5.8, 15.4.7.3	O	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS11.1	If greater than 100 mW capability	15.4.7.3	DS11:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
*DS12	Radio type (temperature range)	15.4.6.10		
DS12.1	Type 1	15.4.6.10	DS12:O.3	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS12.2	Type 2	15.4.6.10	DS12:O.3	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
DS13	Spurious emissions conformance	15.4.6.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS14	TX-RX turnaround time	15.4.6.6	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS15	RX-TX turnaround time	15.4.6.7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS16	Slot time	15.4.6.8	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS17	ED reporting time	15.4.6.8, 15.4.8.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS18	Minimum transmit power level	15.4.7.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS19	Transmit spectral mask conformance	15.4.7.4	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS20	Transmitted center frequency tolerance	15.4.7.5	M	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.6 Direct sequence PHY functions (continued)

Item	PHY feature	References	Status	Support
DS21	Chip clock frequency tolerance	15.4.7.6	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS22	Transmit power on ramp	15.4.7.7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS23	Transmit power down ramp	15.4.7.7	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS24	RF carrier suppression	15.4.7.8	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS25	Transmit modulation accuracy	15.4.7.9	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS26	Receiver minimum input level sensitivity	15.4.8.1	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS27	Receiver maximum input level	15.4.8.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS28	Receiver adjacent channel rejection	15.4.8.3	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS29	MIB	13.1, 15.3.2, Annex D	M	Yes <input type="checkbox"/> No <input type="checkbox"/>
DS29.1	dot11PhyDSSSComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup	13.1, 15.3.2	M	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.7 Infrared baseband PHY functions

Item	Feature	References	Status	Support
IR1	Is the transmitted SYNC field length in the range of required number of PPM slots, with the absence of a pulse in the last slot of the field?	16.2.4.1	M	Yes <input type="checkbox"/>
IR2	Is the transmitted SYNC field entirely populated by alternating presence and absence of pulses in consecutive PPM slots, with the absence of a pulse in the last slot of the field?	16.2.4.1	M	Yes <input type="checkbox"/>
IR3	Is the transmitted SFD field the binary sequence 1001, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot?	16.2.4.2	M	Yes <input type="checkbox"/>
IR4	Is the transmitted DR field pulse sequence equal to the correct value for the data rate provided by the TXVECTOR parameter PLCP BITRATE, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot?	16.2.4.3	M	Yes <input type="checkbox"/>
IR5	Is the transmitted DCLA field 32 PPM slots long with the specified sequence for 1 Mbit/s, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot? 1 Mbit/s: 0000000010000000000000010000000	16.2.4.4	M	Yes <input type="checkbox"/>
* IR5a	Does the unit support 2 Mbit/s transmission?	16.2.4.4	O	Yes <input type="checkbox"/> No <input type="checkbox"/>

A.4.7 Infrared baseband PHY functions (continued)

Item	Feature	References	Status	Support
IR5b	If the unit supports 2 Mbit/s transmission, is the transmitted DCLA field 32 PPM slots long with the specified sequence for 2 Mbit/s, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot? 2 Mbit/s: 00100010001000100010001000100010	16.2.4.4	IR5a:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
IR6	Is the transmitted LENGTH field the correct PPM representation of the unsigned 16-bit binary integer, lsb transmitted first, equal to the correct value provided by the TXVECTOR parameter LENGTH?	16.2.4.5	M	Yes <input type="checkbox"/>
IR7	Is the transmitted CRC field the correct PPM representation of the CRC value calculated as per reference subclause, transmitted lsb first?	16.2.4.6	M	Yes <input type="checkbox"/>
IR8	Is the transmitted PSDU field the correct PPM representation of the PSDU, transmitted lsb first?	16.2.4.7	M	Yes <input type="checkbox"/>
IR9	When the CCA is false does transmission begin based on PHYTXSTART.request?	16.2.5.1	M	Yes <input type="checkbox"/>
IR10	Does the PHY issue a PHYTXSTART.confirm after the transmission of the PLCP header?	16.2.5.1	M	Yes <input type="checkbox"/>
IR11	Does the PHY accept each octet of the PSDU in a PHYDATA.request and answer with a PHYDATA.confirm?	16.2.5.1	M	Yes <input type="checkbox"/>
IR12	Does the PHY cease transmission in response to a PHYTXEND.request and answer with a PHYTXEND.confirm?	16.2.5.1	M	Yes <input type="checkbox"/>
IR13	Does the PHY of a receiving STA send a PHYCCA.indicate during reception of the SYNC field?	16.2.5.2	M	Yes <input type="checkbox"/>
IR14	Does the PHY of a receiving STA properly receive a transmission that changes data rate according to the DR field?	16.2.5.2	M	Yes <input type="checkbox"/>
IR15	Does the PHY of a receiving STA properly reject an incorrect CRC?	16.2.5.2	M	Yes <input type="checkbox"/>
IR16	Does the PHY of a receiving STA properly reject a DR field other than those specified in reference subclause?	16.2.5.2, 16.2.4.3	M	Yes <input type="checkbox"/>
IR17	Does the PHY of a receiving STA send PHYRXSTART.indicate with correct RATE and LENGTH parameters after proper reception of PLCP preamble and PLCP header?	16.2.5.2	M	Yes <input type="checkbox"/>
IR18	Does the PHY of a receiving STA forward receive octets in PHYDATA.indicate primitives?	16.2.5.2	M	Yes <input type="checkbox"/>
IR19	Does the PHY of a receiving STA send a PHYRXEND.indicate after the final octet indicated by the LENGTH field?	16.2.5.2	M	Yes <input type="checkbox"/>

A.4.7 Infrared baseband PHY functions (continued)

Item	Feature	References	Status	Support
IR20	Does the PHY of a receiving STA send a PHYCCA.indicate with a state value of IDLE after the PHYRXEND.indicate?	16.2.5.2	M	Yes <input type="checkbox"/>
IR21	Does the PHY reset its CCA detection mechanism upon receiving a PHYC-CARST.request, and respond with a PHYCCARST.indicate?	16.2.5.3	M	Yes <input type="checkbox"/>
IR22	When transmitting at 1 Mbit/s does the PHY transmit PPM symbols according to the 16-PPM Basic Rate Mapping table, transmitting from left to right?	16.3.2.1, 16.3.2.2	M	Yes <input type="checkbox"/>
IR23	When transmitting at 2 Mbit/s does the PHY transmit PPM symbols according to the 4-PPM Enhanced Rate Mapping table, transmitting from left to right?	16.3.2.1, 16.3.2.2	IR5a:M	Yes <input type="checkbox"/>
IR24	Does the PHY operate over a temperature range of 0 °C to 40 °C?	16.3.2.4	M	Yes <input type="checkbox"/>
* IR25	If the unit is conformant to emitter radiation mask 1, is the peak optical power of an emitted pulse within the specification range averaged over the pulse width?	16.3.3.1	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
* IR26	If the unit is conformant to emitter radiation mask 2, is the peak optical power of an emitted pulse within the specification range averaged over the pulse width?	16.3.3.1	O.1	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
IR27	Does the transmitted pulse shape conform to the description of the reference subclause?	16.3.3.2	M	Yes <input type="checkbox"/>
IR28	Does the emitter radiation pattern as a function of angle conform to the requirements of the reference subclause as applicable based on conformance to emitter radiation mask 1?	16.3.3.3	IR25:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
IR28a	Does the emitter radiation pattern as a function of angle conform to the requirements of the reference subclause as applicable based on conformance to emitter radiation mask 2?	16.3.3.3	IR26:M	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
IR29	Is the peak emitter optical output as a function of wavelength in the range specified?	16.3.3.4	M	Yes <input type="checkbox"/>
IR30	Does the spectrum of the transmit signal amplitude as a voltage or current meet the requirements of the reference subclause?	16.3.3.5	M	Yes <input type="checkbox"/>
IR31	Does the receiver sensitivity meet the requirements of the reference subclause for receive signals of both 1 Mbit/s and 2 Mbit/s?	16.3.4.1	M	Yes <input type="checkbox"/>
IR32	Does the receiver exhibit a dynamic range as specified in reference subclause?	16.3.4.2	M	Yes <input type="checkbox"/>
IR33	Does the receiver field-of-view conform to the requirements of the reference subclause?	16.3.4.3	M	Yes <input type="checkbox"/>
IR34	When it is known that the conditions are such that the Carrier Detect Signal and the Energy Detect Signal are false is the CCA asserted IDLE?	16.3.5.1	M	Yes <input type="checkbox"/>

A.4.7 Infrared baseband PHY functions (continued)

Item	Feature	References	Status	Support
IR35	When the conditions are such that Energy Detect is true for greater than the time defined in reference subclause, does CCA become IDLE?	16.3.5.1	M	Yes <input type="checkbox"/>
IR36	When conditions are such that either Carrier Detect or Energy Detect go true, does CCA go BUSY?	16.3.5.1	M	Yes <input type="checkbox"/>
IR37	Are these compliance groups implemented? dot11PhyIRComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup	16.4	M	Yes <input type="checkbox"/>

Annex B

(informative)

Hopping sequences

The following tables pertain to the hopping sequences for North America and ETSI.

Table B.1—Hopping sequence set 1

index	0	3	6	9	12	15	18	21	24	27	30	33	36
1	2	5	8	11	14	17	20	23	26	29	32	35	38
2	25	28	31	34	37	40	43	46	49	52	55	58	61
3	64	67	70	73	76	79	3	6	9	12	15	18	21
4	10	13	16	19	22	25	28	31	34	37	40	43	46
5	45	48	51	54	57	60	63	66	69	72	75	78	2
6	18	21	24	27	30	33	36	39	42	45	48	51	54
7	73	76	79	3	6	9	12	15	18	21	24	27	30
8	49	52	55	58	61	64	67	70	73	76	79	3	6
9	21	24	27	30	33	36	39	42	45	48	51	54	57
10	63	66	69	72	75	78	2	5	8	11	14	17	20
11	78	2	5	8	11	14	17	20	23	26	29	32	35
12	31	34	37	40	43	46	49	52	55	58	61	64	67
13	61	64	67	70	73	76	79	3	6	9	12	15	18
14	24	27	30	33	36	39	42	45	48	51	54	57	60
15	54	57	60	63	66	69	72	75	78	2	5	8	11
16	65	68	71	74	77	80	4	7	10	13	16	19	22
17	28	31	34	37	40	43	46	49	52	55	58	61	64
18	79	3	6	9	12	15	18	21	24	27	30	33	36
19	33	36	39	42	45	48	51	54	57	60	63	66	69
20	4	7	10	13	16	19	22	25	28	31	34	37	40
21	20	23	26	29	32	35	38	41	44	47	50	53	56
22	13	16	19	22	25	28	31	34	37	40	43	46	49
23	38	41	44	47	50	53	56	59	62	65	68	71	74
24	74	77	80	4	7	10	13	16	19	22	25	28	31
25	56	59	62	65	68	71	74	77	80	4	7	10	13
26	71	74	77	80	4	7	10	13	16	19	22	25	28
27	23	26	29	32	35	38	41	44	47	50	53	56	59
28	5	8	11	14	17	20	23	26	29	32	35	38	41
29	39	42	45	48	51	54	57	60	63	66	69	72	75
30	12	15	18	21	24	27	30	33	36	39	42	45	48
31	36	39	42	45	48	51	54	57	60	63	66	69	72
32	68	71	74	77	80	4	7	10	13	16	19	22	25
33	9	12	15	18	21	24	27	30	33	36	39	42	45
34	70	73	76	79	3	6	9	12	15	18	21	24	27
35	77	80	4	7	10	13	16	19	22	25	28	31	34
36	6	9	12	15	18	21	24	27	30	33	36	39	42
37	62	65	68	71	74	77	80	4	7	10	13	16	19
38	29	32	35	38	41	44	47	50	53	56	59	62	65
39	14	17	20	23	26	29	32	35	38	41	44	47	50

Table B.1—Hopping sequence set 1 (continued)

index	0	3	6	9	12	15	18	21	24	27	30	33	36
40	27	30	33	36	39	42	45	48	51	54	57	60	63
41	16	19	22	25	28	31	34	37	40	43	46	49	52
42	59	62	65	68	71	74	77	80	4	7	10	13	16
43	43	46	49	52	55	58	61	64	67	70	73	76	79
44	76	79	3	6	9	12	15	18	21	24	27	30	33
45	34	37	40	43	46	49	52	55	58	61	64	67	70
46	72	75	78	2	5	8	11	14	17	20	23	26	29
47	11	14	17	20	23	26	29	32	35	38	41	44	47
48	60	63	66	69	72	75	78	2	5	8	11	14	17
49	80	4	7	10	13	16	19	22	25	28	31	34	37
50	47	50	53	56	59	62	65	68	71	74	77	80	4
51	22	25	28	31	34	37	40	43	46	49	52	55	58
52	75	78	2	5	8	11	14	17	20	23	26	29	32
53	66	69	72	75	78	2	5	8	11	14	17	20	23
54	41	44	47	50	53	56	59	62	65	68	71	74	77
55	15	18	21	24	27	30	33	36	39	42	45	48	51
56	35	38	41	44	47	50	53	56	59	62	65	68	71
57	67	70	73	76	79	3	6	9	12	15	18	21	24
58	52	55	58	61	64	67	70	73	76	79	3	6	9
59	58	61	64	67	70	73	76	79	3	6	9	12	15
60	44	47	50	53	56	59	62	65	68	71	74	77	80
61	50	53	56	59	62	65	68	71	74	77	80	4	7
62	17	20	23	26	29	32	35	38	41	44	47	50	53
63	7	10	13	16	19	22	25	28	31	34	37	40	43
64	19	22	25	28	31	34	37	40	43	46	49	52	55
65	8	11	14	17	20	23	26	29	32	35	38	41	44
66	69	72	75	78	2	5	8	11	14	17	20	23	26
67	51	54	57	60	63	66	69	72	75	78	2	5	8
68	42	45	48	51	54	57	60	63	66	69	72	75	78
69	3	6	9	12	15	18	21	24	27	30	33	36	39
70	30	33	36	39	42	45	48	51	54	57	60	63	66
71	57	60	63	66	69	72	75	78	2	5	8	11	14
72	37	40	43	46	49	52	55	58	61	64	67	70	73
73	55	58	61	64	67	70	73	76	79	3	6	9	12
74	26	29	32	35	38	41	44	47	50	53	56	59	62
75	46	49	52	55	58	61	64	67	70	73	76	79	3
76	53	56	59	62	65	68	71	74	77	80	4	7	10
77	40	43	46	49	52	55	58	61	64	67	70	73	76
78	32	35	38	41	44	47	50	53	56	59	62	65	68
79	48	51	54	57	60	63	66	69	72	75	78	2	5

Table B.1—Hopping sequence set 1 (continued)

index	39	42	45	48	51	54	57	60	63	66	69	72	75
1	41	44	47	50	53	56	59	62	65	68	71	74	77
2	64	67	70	73	76	79	3	6	9	12	15	18	21
3	24	27	30	33	36	39	42	45	48	51	54	57	60
4	49	52	55	58	61	64	67	70	73	76	79	3	6
5	5	8	11	14	17	20	23	26	29	32	35	38	41
6	57	60	63	66	69	72	75	78	2	5	8	11	14
7	33	36	39	42	45	48	51	54	57	60	63	66	69
8	9	12	15	18	21	24	27	30	33	36	39	42	45
9	60	63	66	69	72	75	78	2	5	8	11	14	17
10	23	26	29	32	35	38	41	44	47	50	53	56	59
11	38	41	44	47	50	53	56	59	62	65	68	71	74
12	70	73	76	79	3	6	9	12	15	18	21	24	27
13	21	24	27	30	33	36	39	42	45	48	51	54	57
14	63	66	69	72	75	78	2	5	8	11	14	17	20
15	14	17	20	23	26	29	32	35	38	41	44	47	50
16	25	28	31	34	37	40	43	46	49	52	55	58	61
17	67	70	73	76	79	3	6	9	12	15	18	21	24
18	39	42	45	48	51	54	57	60	63	66	69	72	75
19	72	75	78	2	5	8	11	14	17	20	23	26	29
20	43	46	49	52	55	58	61	64	67	70	73	76	79
21	59	62	65	68	71	74	77	80	4	7	10	13	16
22	52	55	58	61	64	67	70	73	76	79	3	6	9
23	77	80	4	7	10	13	16	19	22	25	28	31	34
24	34	37	40	43	46	49	52	55	58	61	64	67	70
25	16	19	22	25	28	31	34	37	40	43	46	49	52
26	31	34	37	40	43	46	49	52	55	58	61	64	67
27	62	65	68	71	74	77	80	4	7	10	13	16	19
28	44	47	50	53	56	59	62	65	68	71	74	77	80
29	78	2	5	8	11	14	17	20	23	26	29	32	35
30	51	54	57	60	63	66	69	72	75	78	2	5	8
31	75	78	2	5	8	11	14	17	20	23	26	29	32
32	28	31	34	37	40	43	46	49	52	55	58	61	64
33	48	51	54	57	60	63	66	69	72	75	78	2	5
34	30	33	36	39	42	45	48	51	54	57	60	63	66
35	37	40	43	46	49	52	55	58	61	64	67	70	73
36	45	48	51	54	57	60	63	66	69	72	75	78	2
37	22	25	28	31	34	37	40	43	46	49	52	55	58
38	68	71	74	77	80	4	7	10	13	16	19	22	25
39	53	56	59	62	65	68	71	74	77	80	4	7	10

Table B.1—Hopping sequence set 1 (continued)

index	39	42	45	48	51	54	57	60	63	66	69	72	75
40	66	69	72	75	78	2	5	8	11	14	17	20	23
41	55	58	61	64	67	70	73	76	79	3	6	9	12
42	19	22	25	28	31	34	37	40	43	46	49	52	55
43	3	6	9	12	15	18	21	24	27	30	33	36	39
44	36	39	42	45	48	51	54	57	60	63	66	69	72
45	73	76	79	3	6	9	12	15	18	21	24	27	30
46	32	35	38	41	44	47	50	53	56	59	62	65	68
47	50	53	56	59	62	65	68	71	74	77	80	4	7
48	20	23	26	29	32	35	38	41	44	47	50	53	56
49	40	43	46	49	52	55	58	61	64	67	70	73	76
50	7	10	13	16	19	22	25	28	31	34	37	40	43
51	61	64	67	70	73	76	79	3	6	9	12	15	18
52	35	38	41	44	47	50	53	56	59	62	65	68	71
53	26	29	32	35	38	41	44	47	50	53	56	59	62
54	80	4	7	10	13	16	19	22	25	28	31	34	37
55	54	57	60	63	66	69	72	75	78	2	5	8	11
56	74	77	80	4	7	10	13	16	19	22	25	28	31
57	27	30	33	36	39	42	45	48	51	54	57	60	63
58	12	15	18	21	24	27	30	33	36	39	42	45	48
59	18	21	24	27	30	33	36	39	42	45	48	51	54
60	4	7	10	13	16	19	22	25	28	31	34	37	40
61	10	13	16	19	22	25	28	31	34	37	40	43	46
62	56	59	62	65	68	71	74	77	80	4	7	10	13
63	46	49	52	55	58	61	64	67	70	73	76	79	3
64	58	61	64	67	70	73	76	79	3	6	9	12	15
65	47	50	53	56	59	62	65	68	71	74	77	80	4
66	29	32	35	38	41	44	47	50	53	56	59	62	65
67	11	14	17	20	23	26	29	32	35	38	41	44	47
68	2	5	8	11	14	17	20	23	26	29	32	35	38
69	42	45	48	51	54	57	60	63	66	69	72	75	78
70	69	72	75	78	2	5	8	11	14	17	20	23	26
71	17	20	23	26	29	32	35	38	41	44	47	50	53
72	76	79	3	6	9	12	15	18	21	24	27	30	33
73	15	18	21	24	27	30	33	36	39	42	45	48	51
74	65	68	71	74	77	80	4	7	10	13	16	19	22
75	6	9	12	15	18	21	24	27	30	33	36	39	42
76	13	16	19	22	25	28	31	34	37	40	43	46	49
77	79	3	6	9	12	15	18	21	24	27	30	33	36
78	71	74	77	80	4	7	10	13	16	19	22	25	28
79	8	11	14	17	20	23	26	29	32	35	38	41	44

Table B.2—Hopping sequence set 2

index	1	4	7	10	13	16	19	22	25	28	31	34	37
1	3	6	9	12	15	18	21	24	27	30	33	36	39
2	26	29	32	35	38	41	44	47	50	53	56	59	62
3	65	68	71	74	77	80	4	7	10	13	16	19	22
4	11	14	17	20	23	26	29	32	35	38	41	44	47
5	46	49	52	55	58	61	64	67	70	73	76	79	3
6	19	22	25	28	31	34	37	40	43	46	49	52	55
7	74	77	80	4	7	10	13	16	19	22	25	28	31
8	50	53	56	59	62	65	68	71	74	77	80	4	7
9	22	25	28	31	34	37	40	43	46	49	52	55	58
10	64	67	70	73	76	79	3	6	9	12	15	18	21
11	79	3	6	9	12	15	18	21	24	27	30	33	36
12	32	35	38	41	44	47	50	53	56	59	62	65	68
13	62	65	68	71	74	77	80	4	7	10	13	16	19
14	25	28	31	34	37	40	43	46	49	52	55	58	61
15	55	58	61	64	67	70	73	76	79	3	6	9	12
16	66	69	72	75	78	2	5	8	11	14	17	20	23
17	29	32	35	38	41	44	47	50	53	56	59	62	65
18	80	4	7	10	13	16	19	22	25	28	31	34	37
19	34	37	40	43	46	49	52	55	58	61	64	67	70
20	5	8	11	14	17	20	23	26	29	32	35	38	41
21	21	24	27	30	33	36	39	42	45	48	51	54	57
22	14	17	20	23	26	29	32	35	38	41	44	47	50
23	39	42	45	48	51	54	57	60	63	66	69	72	75
24	75	78	2	5	8	11	14	17	20	23	26	29	32
25	57	60	63	66	69	72	75	78	2	5	8	11	14
26	72	75	78	2	5	8	11	14	17	20	23	26	29
27	24	27	30	33	36	39	42	45	48	51	54	57	60
28	6	9	12	15	18	21	24	27	30	33	36	39	42
29	40	43	46	49	52	55	58	61	64	67	70	73	76
30	13	16	19	22	25	28	31	34	37	40	43	46	49
31	37	40	43	46	49	52	55	58	61	64	67	70	73
32	69	72	75	78	2	5	8	11	14	17	20	23	26
33	10	13	16	19	22	25	28	31	34	37	40	43	46
34	71	74	77	80	4	7	10	13	16	19	22	25	28
35	78	2	5	8	11	14	17	20	23	26	29	32	35
36	7	10	13	16	19	22	25	28	31	34	37	40	43
37	63	66	69	72	75	78	2	5	8	11	14	17	20
38	30	33	36	39	42	45	48	51	54	57	60	63	66
39	15	18	21	24	27	30	33	36	39	42	45	48	51

Table B.2—Hopping sequence set 2 (continued)

index	1	4	7	10	13	16	19	22	25	28	31	34	37
40	28	31	34	37	40	43	46	49	52	55	58	61	64
41	17	20	23	26	29	32	35	38	41	44	47	50	53
42	60	63	66	69	72	75	78	2	5	8	11	14	17
43	44	47	50	53	56	59	62	65	68	71	74	77	80
44	77	80	4	7	10	13	16	19	22	25	28	31	34
45	35	38	41	44	47	50	53	56	59	62	65	68	71
46	73	76	79	3	6	9	12	15	18	21	24	27	30
47	12	15	18	21	24	27	30	33	36	39	42	45	48
48	61	64	67	70	73	76	79	3	6	9	12	15	18
49	2	5	8	11	14	17	20	23	26	29	32	35	38
50	48	51	54	57	60	63	66	69	72	75	78	2	5
51	23	26	29	32	35	38	41	44	47	50	53	56	59
52	76	79	3	6	9	12	15	18	21	24	27	30	33
53	67	70	73	76	79	3	6	9	12	15	18	21	24
54	42	45	48	51	54	57	60	63	66	69	72	75	78
55	16	19	22	25	28	31	34	37	40	43	46	49	52
56	36	39	42	45	48	51	54	57	60	63	66	69	72
57	68	71	74	77	80	4	7	10	13	16	19	22	25
58	53	56	59	62	65	68	71	74	77	80	4	7	10
59	59	62	65	68	71	74	77	80	4	7	10	13	16
60	45	48	51	54	57	60	63	66	69	72	75	78	2
61	51	54	57	60	63	66	69	72	75	78	2	5	8
62	18	21	24	27	30	33	36	39	42	45	48	51	54
63	8	11	14	17	20	23	26	29	32	35	38	41	44
64	20	23	26	29	32	35	38	41	44	47	50	53	56
65	9	12	15	18	21	24	27	30	33	36	39	42	45
66	70	73	76	79	3	6	9	12	15	18	21	24	27
67	52	55	58	61	64	67	70	73	76	79	3	6	9
68	43	46	49	52	55	58	61	64	67	70	73	76	79
69	4	7	10	13	16	19	22	25	28	31	34	37	40
70	31	34	37	40	43	46	49	52	55	58	61	64	67
71	58	61	64	67	70	73	76	79	3	6	9	12	15
72	38	41	44	47	50	53	56	59	62	65	68	71	74
73	56	59	62	65	68	71	74	77	80	4	7	10	13
74	27	30	33	36	39	42	45	48	51	54	57	60	63
75	47	50	53	56	59	62	65	68	71	74	77	80	4
76	54	57	60	63	66	69	72	75	78	2	5	8	11
77	41	44	47	50	53	56	59	62	65	68	71	74	77
78	33	36	39	42	45	48	51	54	57	60	63	66	69
79	49	52	55	58	61	64	67	70	73	76	79	3	6

Table B.2—Hopping sequence set 2 (continued)

index	40	43	46	49	52	55	58	61	64	67	70	73	76
1	42	45	48	51	54	57	60	63	66	69	72	75	78
2	65	68	71	74	77	80	4	7	10	13	16	19	22
3	25	28	31	34	37	40	43	46	49	52	55	58	61
4	50	53	56	59	62	65	68	71	74	77	80	4	7
5	6	9	12	15	18	21	24	27	30	33	36	39	42
6	58	61	64	67	70	73	76	79	3	6	9	12	15
7	34	37	40	43	46	49	52	55	58	61	64	67	70
8	10	13	16	19	22	25	28	31	34	37	40	43	46
9	61	64	67	70	73	76	79	3	6	9	12	15	18
10	24	27	30	33	36	39	42	45	48	51	54	57	60
11	39	42	45	48	51	54	57	60	63	66	69	72	75
12	71	74	77	80	4	7	10	13	16	19	22	25	28
13	22	25	28	31	34	37	40	43	46	49	52	55	58
14	64	67	70	73	76	79	3	6	9	12	15	18	21
15	15	18	21	24	27	30	33	36	39	42	45	48	51
16	26	29	32	35	38	41	44	47	50	53	56	59	62
17	68	71	74	77	80	4	7	10	13	16	19	22	25
18	40	43	46	49	52	55	58	61	64	67	70	73	76
19	73	76	79	3	6	9	12	15	18	21	24	27	30
20	44	47	50	53	56	59	62	65	68	71	74	77	80
21	60	63	66	69	72	75	78	2	5	8	11	14	17
22	53	56	59	62	65	68	71	74	77	80	4	7	10
23	78	2	5	8	11	14	17	20	23	26	29	32	35
24	35	38	41	44	47	50	53	56	59	62	65	68	71
25	17	20	23	26	29	32	35	38	41	44	47	50	53
26	32	35	38	41	44	47	50	53	56	59	62	65	68
27	63	66	69	72	75	78	2	5	8	11	14	17	20
28	45	48	51	54	57	60	63	66	69	72	75	78	2
29	79	3	6	9	12	15	18	21	24	27	30	33	36
30	52	55	58	61	64	67	70	73	76	79	3	6	9
31	76	79	3	6	9	12	15	18	21	24	27	30	33
32	29	32	35	38	41	44	47	50	53	56	59	62	65
33	49	52	55	58	61	64	67	70	73	76	79	3	6
34	31	34	37	40	43	46	49	52	55	58	61	64	67
35	38	41	44	47	50	53	56	59	62	65	68	71	74
36	46	49	52	55	58	61	64	67	70	73	76	79	3
37	23	26	29	32	35	38	41	44	47	50	53	56	59
38	69	72	75	78	2	5	8	11	14	17	20	23	26
39	54	57	60	63	66	69	72	75	78	2	5	8	11

Table B.2—Hopping sequence set 2 (continued)

index	40	43	46	49	52	55	58	61	64	67	70	73	76
40	67	70	73	76	79	3	6	9	12	15	18	21	24
41	56	59	62	65	68	71	74	77	80	4	7	10	13
42	20	23	26	29	32	35	38	41	44	47	50	53	56
43	4	7	10	13	16	19	22	25	28	31	34	37	40
44	37	40	43	46	49	52	55	58	61	64	67	70	73
45	74	77	80	4	7	10	13	16	19	22	25	28	31
46	33	36	39	42	45	48	51	54	57	60	63	66	69
47	51	54	57	60	63	66	69	72	75	78	2	5	8
48	21	24	27	30	33	36	39	42	45	48	51	54	57
49	41	44	47	50	53	56	59	62	65	68	71	74	77
50	8	11	14	17	20	23	26	29	32	35	38	41	44
51	62	65	68	71	74	77	80	4	7	10	13	16	19
52	36	39	42	45	48	51	54	57	60	63	66	69	72
53	27	30	33	36	39	42	45	48	51	54	57	60	63
54	2	5	8	11	14	17	20	23	26	29	32	35	38
55	55	58	61	64	67	70	73	76	79	3	6	9	12
56	75	78	2	5	8	11	14	17	20	23	26	29	32
57	28	31	34	37	40	43	46	49	52	55	58	61	64
58	13	16	19	22	25	28	31	34	37	40	43	46	49
59	19	22	25	28	31	34	37	40	43	46	49	52	55
60	5	8	11	14	17	20	23	26	29	32	35	38	41
61	11	14	17	20	23	26	29	32	35	38	41	44	47
62	57	60	63	66	69	72	75	78	2	5	8	11	14
63	47	50	53	56	59	62	65	68	71	74	77	80	4
64	59	62	65	68	71	74	77	80	4	7	10	13	16
65	48	51	54	57	60	63	66	69	72	75	78	2	5
66	30	33	36	39	42	45	48	51	54	57	60	63	66
67	12	15	18	21	24	27	30	33	36	39	42	45	48
68	3	6	9	12	15	18	21	24	27	30	33	36	39
69	43	46	49	52	55	58	61	64	67	70	73	76	79
70	70	73	76	79	3	6	9	12	15	18	21	24	27
71	18	21	24	27	30	33	36	39	42	45	48	51	54
72	77	80	4	7	10	13	16	19	22	25	28	31	34
73	16	19	22	25	28	31	34	37	40	43	46	49	52
74	66	69	72	75	78	2	5	8	11	14	17	20	23
75	7	10	13	16	19	22	25	28	31	34	37	40	43
76	14	17	20	23	26	29	32	35	38	41	44	47	50
77	80	4	7	10	13	16	19	22	25	28	31	34	37
78	72	75	78	2	5	8	11	14	17	20	23	26	29
79	9	12	15	18	21	24	27	30	33	36	39	42	45

Table B.3—Hopping sequence set 3

index	2	5	8	11	14	17	20	23	26	29	32	35	38
1	4	7	10	13	16	19	22	25	28	31	34	37	40
2	27	30	33	36	39	42	45	48	51	54	57	60	63
3	66	69	72	75	78	2	5	8	11	14	17	20	23
4	12	15	18	21	24	27	30	33	36	39	42	45	48
5	47	50	53	56	59	62	65	68	71	74	77	80	4
6	20	23	26	29	32	35	38	41	44	47	50	53	56
7	75	78	2	5	8	11	14	17	20	23	26	29	32
8	51	54	57	60	63	66	69	72	75	78	2	5	8
9	23	26	29	32	35	38	41	44	47	50	53	56	59
10	65	68	71	74	77	80	4	7	10	13	16	19	22
11	80	4	7	10	13	16	19	22	25	28	31	34	37
12	33	36	39	42	45	48	51	54	57	60	63	66	69
13	63	66	69	72	75	78	2	5	8	11	14	17	20
14	26	29	32	35	38	41	44	47	50	53	56	59	62
15	56	59	62	65	68	71	74	77	80	4	7	10	13
16	67	70	73	76	79	3	6	9	12	15	18	21	24
17	30	33	36	39	42	45	48	51	54	57	60	63	66
18	2	5	8	11	14	17	20	23	26	29	32	35	38
19	35	38	41	44	47	50	53	56	59	62	65	68	71
20	6	9	12	15	18	21	24	27	30	33	36	39	42
21	22	25	28	31	34	37	40	43	46	49	52	55	58
22	15	18	21	24	27	30	33	36	39	42	45	48	51
23	40	43	46	49	52	55	58	61	64	67	70	73	76
24	76	79	3	6	9	12	15	18	21	24	27	30	33
25	58	61	64	67	70	73	76	79	3	6	9	12	15
26	73	76	79	3	6	9	12	15	18	21	24	27	30
27	25	28	31	34	37	40	43	46	49	52	55	58	61
28	7	10	13	16	19	22	25	28	31	34	37	40	43
29	41	44	47	50	53	56	59	62	65	68	71	74	77
30	14	17	20	23	26	29	32	35	38	41	44	47	50
31	38	41	44	47	50	53	56	59	62	65	68	71	74
32	70	73	76	79	3	6	9	12	15	18	21	24	27
33	11	14	17	20	23	26	29	32	35	38	41	44	47
34	72	75	78	2	5	8	11	14	17	20	23	26	29
35	79	3	6	9	12	15	18	21	24	27	30	33	36
36	8	11	14	17	20	23	26	29	32	35	38	41	44
37	64	67	70	73	76	79	3	6	9	12	15	18	21
38	31	34	37	40	43	46	49	52	55	58	61	64	67
39	16	19	22	25	28	31	34	37	40	43	46	49	52

Table B.3—Hopping sequence set 3 (continued)

index	2	5	8	11	14	17	20	23	26	29	32	35	38
40	29	32	35	38	41	44	47	50	53	56	59	62	65
41	18	21	24	27	30	33	36	39	42	45	48	51	54
42	61	64	67	70	73	76	79	3	6	9	12	15	18
43	45	48	51	54	57	60	63	66	69	72	75	78	2
44	78	2	5	8	11	14	17	20	23	26	29	32	35
45	36	39	42	45	48	51	54	57	60	63	66	69	72
46	74	77	80	4	7	10	13	16	19	22	25	28	31
47	13	16	19	22	25	28	31	34	37	40	43	46	49
48	62	65	68	71	74	77	80	4	7	10	13	16	19
49	3	6	9	12	15	18	21	24	27	30	33	36	39
50	49	52	55	58	61	64	67	70	73	76	79	3	6
51	24	27	30	33	36	39	42	45	48	51	54	57	60
52	77	80	4	7	10	13	16	19	22	25	28	31	34
53	68	71	74	77	80	4	7	10	13	16	19	22	25
54	43	46	49	52	55	58	61	64	67	70	73	76	79
55	17	20	23	26	29	32	35	38	41	44	47	50	53
56	37	40	43	46	49	52	55	58	61	64	67	70	73
57	69	72	75	78	2	5	8	11	14	17	20	23	26
58	54	57	60	63	66	69	72	75	78	2	5	8	11
59	60	63	66	69	72	75	78	2	5	8	11	14	17
60	46	49	52	55	58	61	64	67	70	73	76	79	3
61	52	55	58	61	64	67	70	73	76	79	3	6	9
62	19	22	25	28	31	34	37	40	43	46	49	52	55
63	9	12	15	18	21	24	27	30	33	36	39	42	45
64	21	24	27	30	33	36	39	42	45	48	51	54	57
65	10	13	16	19	22	25	28	31	34	37	40	43	46
66	71	74	77	80	4	7	10	13	16	19	22	25	28
67	53	56	59	62	65	68	71	74	77	80	4	7	10
68	44	47	50	53	56	59	62	65	68	71	74	77	80
69	5	8	11	14	17	20	23	26	29	32	35	38	41
70	32	35	38	41	44	47	50	53	56	59	62	65	68
71	59	62	65	68	71	74	77	80	4	7	10	13	16
72	39	42	45	48	51	54	57	60	63	66	69	72	75
73	57	60	63	66	69	72	75	78	2	5	8	11	14
74	28	31	34	37	40	43	46	49	52	55	58	61	64
75	48	51	54	57	60	63	66	69	72	75	78	2	5
76	55	58	61	64	67	70	73	76	79	3	6	9	12
77	42	45	48	51	54	57	60	63	66	69	72	75	78
78	34	37	40	43	46	49	52	55	58	61	64	67	70
79	50	53	56	59	62	65	68	71	74	77	80	4	7

Table B.3—Hopping sequence set 3 (continued)

index	41	44	47	50	53	56	59	62	65	68	71	74	77
1	43	46	49	52	55	58	61	64	67	70	73	76	79
2	66	69	72	75	78	2	5	8	11	14	17	20	23
3	26	29	32	35	38	41	44	47	50	53	56	59	62
4	51	54	57	60	63	66	69	72	75	78	2	5	8
5	7	10	13	16	19	22	25	28	31	34	37	40	43
6	59	62	65	68	71	74	77	80	4	7	10	13	16
7	35	38	41	44	47	50	53	56	59	62	65	68	71
8	11	14	17	20	23	26	29	32	35	38	41	44	47
9	62	65	68	71	74	77	80	4	7	10	13	16	19
10	25	28	31	34	37	40	43	46	49	52	55	58	61
11	40	43	46	49	52	55	58	61	64	67	70	73	76
12	72	75	78	2	5	8	11	14	17	20	23	26	29
13	23	26	29	32	35	38	41	44	47	50	53	56	59
14	65	68	71	74	77	80	4	7	10	13	16	19	22
15	16	19	22	25	28	31	34	37	40	43	46	49	52
16	27	30	33	36	39	42	45	48	51	54	57	60	63
17	69	72	75	78	2	5	8	11	14	17	20	23	26
18	41	44	47	50	53	56	59	62	65	68	71	74	77
19	74	77	80	4	7	10	13	16	19	22	25	28	31
20	45	48	51	54	57	60	63	66	69	72	75	78	2
21	61	64	67	70	73	76	79	3	6	9	12	15	18
22	54	57	60	63	66	69	72	75	78	2	5	8	11
23	79	3	6	9	12	15	18	21	24	27	30	33	36
24	36	39	42	45	48	51	54	57	60	63	66	69	72
25	18	21	24	27	30	33	36	39	42	45	48	51	54
26	33	36	39	42	45	48	51	54	57	60	63	66	69
27	64	67	70	73	76	79	3	6	9	12	15	18	21
28	46	49	52	55	58	61	64	67	70	73	76	79	3
29	80	4	7	10	13	16	19	22	25	28	31	34	37
30	53	56	59	62	65	68	71	74	77	80	4	7	10
31	77	80	4	7	10	13	16	19	22	25	28	31	34
32	30	33	36	39	42	45	48	51	54	57	60	63	66
33	50	53	56	59	62	65	68	71	74	77	80	4	7
34	32	35	38	41	44	47	50	53	56	59	62	65	68
35	39	42	45	48	51	54	57	60	63	66	69	72	75
36	47	50	53	56	59	62	65	68	71	74	77	80	4
37	24	27	30	33	36	39	42	45	48	51	54	57	60
38	70	73	76	79	3	6	9	12	15	18	21	24	27
39	55	58	61	64	67	70	73	76	79	3	6	9	12

Table B.3—Hopping sequence set 3 (continued)

index	41	44	47	50	53	56	59	62	65	68	71	74	77
40	68	71	74	77	80	4	7	10	13	16	19	22	25
41	57	60	63	66	69	72	75	78	2	5	8	11	14
42	21	24	27	30	33	36	39	42	45	48	51	54	57
43	5	8	11	14	17	20	23	26	29	32	35	38	41
44	38	41	44	47	50	53	56	59	62	65	68	71	74
45	75	78	2	5	8	11	14	17	20	23	26	29	32
46	34	37	40	43	46	49	52	55	58	61	64	67	70
47	52	55	58	61	64	67	70	73	76	79	3	6	9
48	22	25	28	31	34	37	40	43	46	49	52	55	58
49	42	45	48	51	54	57	60	63	66	69	72	75	78
50	9	12	15	18	21	24	27	30	33	36	39	42	45
51	63	66	69	72	75	78	2	5	8	11	14	17	20
52	37	40	43	46	49	52	55	58	61	64	67	70	73
53	28	31	34	37	40	43	46	49	52	55	58	61	64
54	3	6	9	12	15	18	21	24	27	30	33	36	39
55	56	59	62	65	68	71	74	77	80	4	7	10	13
56	76	79	3	6	9	12	15	18	21	24	27	30	33
57	29	32	35	38	41	44	47	50	53	56	59	62	65
58	14	17	20	23	26	29	32	35	38	41	44	47	50
59	20	23	26	29	32	35	38	41	44	47	50	53	56
60	6	9	12	15	18	21	24	27	30	33	36	39	42
61	12	15	18	21	24	27	30	33	36	39	42	45	48
62	58	61	64	67	70	73	76	79	3	6	9	12	15
63	48	51	54	57	60	63	66	69	72	75	78	2	5
64	60	63	66	69	72	75	78	2	5	8	11	14	17
65	49	52	55	58	61	64	67	70	73	76	79	3	6
66	31	34	37	40	43	46	49	52	55	58	61	64	67
67	13	16	19	22	25	28	31	34	37	40	43	46	49
68	4	7	10	13	16	19	22	25	28	31	34	37	40
69	44	47	50	53	56	59	62	65	68	71	74	77	80
70	71	74	77	80	4	7	10	13	16	19	22	25	28
71	19	22	25	28	31	34	37	40	43	46	49	52	55
72	78	2	5	8	11	14	17	20	23	26	29	32	35
73	17	20	23	26	29	32	35	38	41	44	47	50	53
74	67	70	73	76	79	3	6	9	12	15	18	21	24
75	8	11	14	17	20	23	26	29	32	35	38	41	44
76	15	18	21	24	27	30	33	36	39	42	45	48	51
77	2	5	8	11	14	17	20	23	26	29	32	35	38
78	73	76	79	3	6	9	12	15	18	21	24	27	30
79	10	13	16	19	22	25	28	31	34	37	40	43	46

Annex C

(normative)

Formal description of MAC operation

This annex contains formal descriptions of the behavior of MAC station (STA) and access point (AP) entities. These descriptions also describe the frame formats and the generation and interpretation of information encoded in MAC frames, in the parameters of service primitives supported by the MAC, and in MIB attributes used or generated by the MAC. The MAC is described using the 1992 version of the ITU Specification and Description Language (SDL-92). SDL-92 is defined in ITU-T Recommendation Z.100 (03/93). An update to Z.100 was approved in 1996 (SDL-96), but none of the SDL facilities used in this annex were modified. An introduction to the MAC formal description is provided in Clause C.1. Definitions of the data types and operators used by the MAC state machines are provided in Clause C.2. An SDL system describing MAC operation at an IEEE 802.11 station is contained in Clause C.3. Finally, a subset of an SDL system describing the aspects of MAC operation at an IEEE 802.11 AP that differ from operation at a non-AP station is provided in Clause C.4.

In Annex D, the MAC and PHY management information bases are described in Abstract Syntax Notation One (ASN.1), defined in ISO/IEC 8824: 1990 and ISO/IEC 8825: 1990. ITU-T Recommendation Z.105 (03/95) defines the use of SDL in conjunction with ASN.1, allowing system behavior to be defined using SDL and data types to be defined using ASN.1. Incomplete tool support precluded the use of ITU-T Recommendation Z.105 in this annex. However, within the limits of ITU-T Recommendation Z.100 (referred to subsequently as Z.100), the data types in Clause C.2 are defined in a similar manner to ITU-T Recommendation Z.105 (referred to subsequently as Z.105). Annex E contains a listing of available documentation.

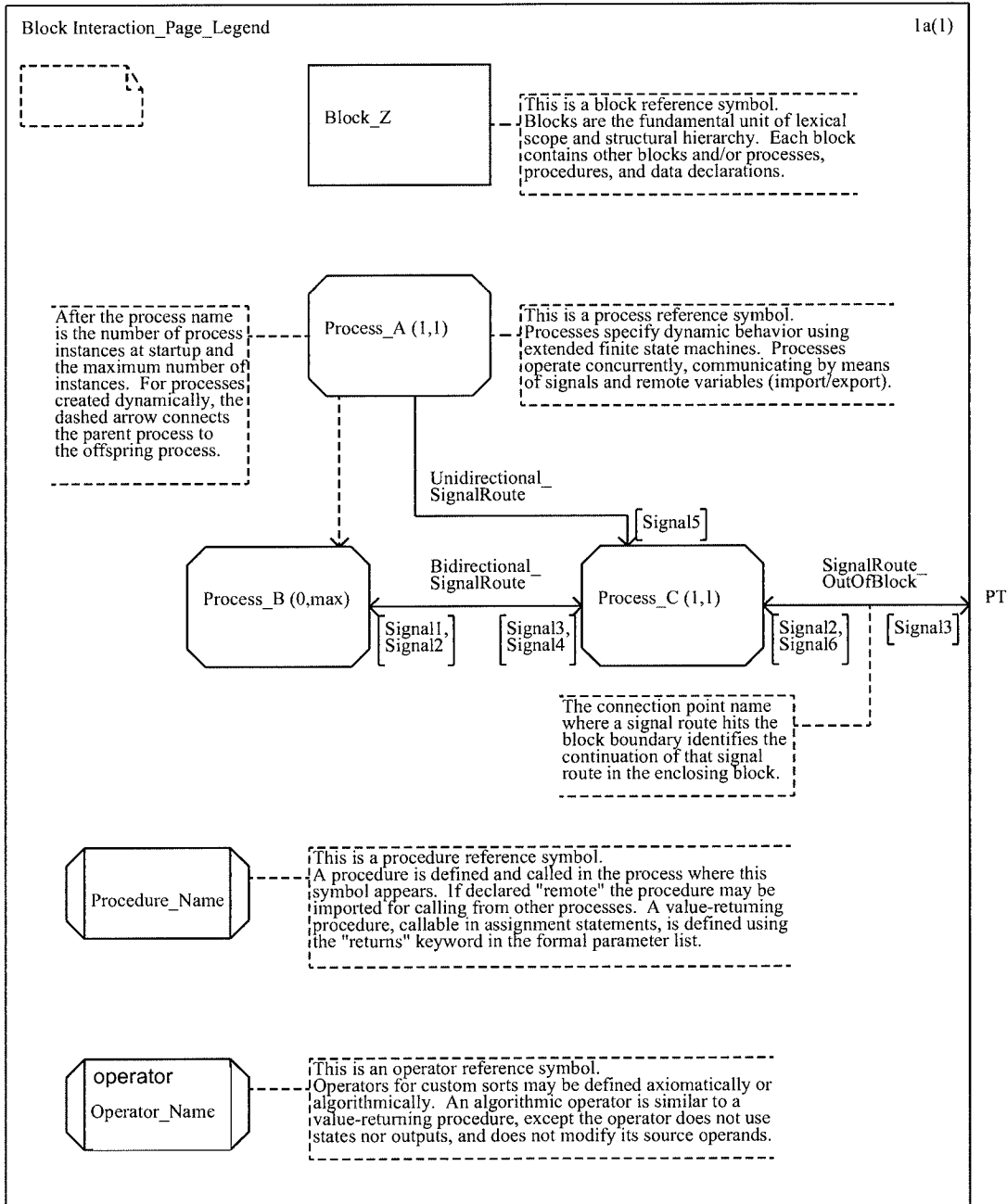
NOTES

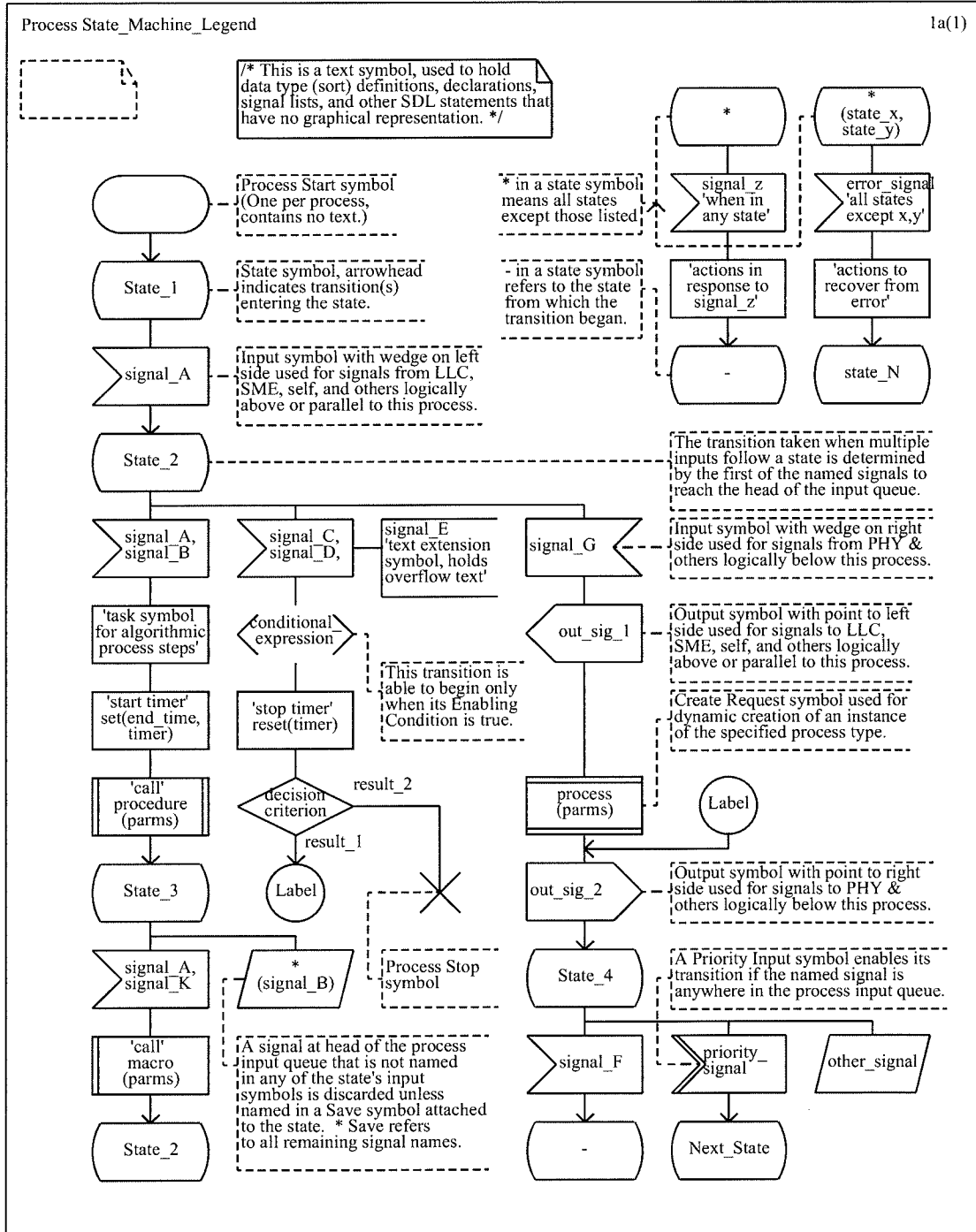
1—The SDL definitions in this annex should be usable with any SDL tool that supports the 1993 version or 1996 update of ITU-T Recommendation Z.100. Software for generating, analyzing, verifying, and simulating SDL system descriptions is available from several sources.

2—The SDL code in this annex was generated using *SDT/PC version 3.02*; from Telelogic AB, Malmo, Sweden (+46-40-174700; internet: telelogic.se); USA office in Princeton, NJ (+1-609-520-1935; internet: telelogic.com). Telelogic offers SDT for several workstation platforms in addition to SDT/PC.

3—The use of Telelogic's product to prepare this annex does not constitute an endorsement of SDT by the IEEE LAN MAN Standards Committee or by the IEEE.

4—The diagrams on the next two pages show most of the symbols of SDL graphical syntax (SDL-GR) used in the MAC formal description. The symbols in these diagrams have labels and comments that explain their meanings. These diagrams are intended to serve as a legend for the SDL-GR symbols that comprise most of the process interaction and state transition diagrams. These diagrams are neither a complete SDL system, nor a complete presentation of SDL-GR symbology. Also, this state machine fragment exists to illustrate the SDL graphical syntax, and does not describe any useful behavior.





C.1 Introduction to the MAC formal description

This formal description defines the behavior of IEEE 802.11 MAC entities. The MAC protocol functional decomposition used herein facilitates explicit description of the reference points and durations of the various timed intervals; the bases for generation and/or validation of header fields, service parameters, and MIB attributes; and the interpretation of each value in cases where enumerated data types are used in service parameters.

C.1.1 Fundamental assumptions

The MAC protocol is described as an SDL system, which is a set of extended finite state machines. Each state machine is a set of independent processes, all of which operate concurrently. All variable data-holding entities and procedures exist solely within the context of a single process. In SDL all interprocess communication is done with signals (there are no global variables). Signals may be sent and received explicitly, using SDL's output and input symbols, or implicitly, using SDL's export/import mechanism (only if the variables or procedures are declared "remote"). By default, signals incur delays when traversing channels between blocks; however, only nondelaying channels and signal routes are used in the MAC state machines, and all remote variables and procedures are declared with the "nodelay" property.

State transitions, procedure calls, and tasks (assignment statements and other algorithmic processing steps) are assumed to require zero time. This permits the time intervals that are part of the normative MAC behavior to be defined explicitly, using SDL timers. One unit of system time (a 1.0 change in the value of "now") is assumed to represent one microsecond of real time. Usec (microsecond) and TU (time unit) data types are defined, with operators to convert Usec and TU values to SDL time or duration when necessary.

The SDL system boundary encloses the MAC entities. The LLC, SME, PHY, and distribution system are part of the environment. SDL generally assumes that entities in the environment operate as specified; however, the MAC state machines that communicate with the various SAPs attempt to validate inputs from the environment, and to handle cases where a pair of communicating entities, one within the system and the other outside the system boundary, have different local views of the medium, station, or service state. All stations in an IEEE 802.11 service set are assumed to exhibit the behaviors described herein. Nevertheless, because of the open nature of the wireless medium, the MAC state machines check for error cases that can arise only when an entity on the wireless medium is transmitting IEEE 802.11 PDUs, but is not obeying the communication protocols specified by this standard.

C.1.2 Notation conventions

When practical, names used in the clauses of this standard are spelled identically in this annex. The principal exceptions are those names that conflict with one of SDL's reserved words (such as power management mode "active," which is renamed "sta_active" in SDL). To help fit the SDL text into the graphic symbols, acronyms with multiple, sequential capital letters are written with only the first letter capitalized (e.g., "MSDU" is written "Msdu" and "MLMEJoin.request" is written "MlmeJoin.request").

SDL reserved words and the names of variables and synonyms (named constants) begin with lowercase letters. The names of sorts (data types), signals, signal routes, channels, blocks, and processes begin with uppercase letters. The names of certain groups of variables and/or synonyms begin with a particular lowercase letter, followed by the remainder of the name, beginning with an uppercase letter. These groups are

"aNameOfAttribute"	PHY operational parameters.
"cNameOfCapability"	Capability bits, also used for internal values exported as MIB counters.
"dNameOfDuration"	Duration (relative time) values, declared as Usec, TU, or Duration.
"dot11NameOfAttribute"	MIB attributes.

“eNameOfElement”	Element ID values.
“mNameOfVariable”	Remote variables used for intra-MAC communication, but not part of the MIB. Most of these variables are exported from the MLME block.
“sNameOfStaticValue”	Synonyms for static data values used within the MAC.
“tNameOfTime”	Time (absolute time) values, declared as Usec, TU, or Time. The names of timers begin with “T.”

C.1.3 Modeling techniques

State machines are grouped according to defined function sets that are visible, directly or indirectly, at an exposed interface. The emphasis in the organization of the state machines is explicitly to show initiation of and response to events at the exposed interfaces, and time-related actions, including those dependent on the absence of external events (e.g., response timeouts) and intervals measured in derived units (e.g., backoff “time” in units of slots during which the wireless medium is idle). The operations associated with the various state transitions emphasize communication functions. Most of the details regarding insertion, extraction, and encoding of information in fields of the PDUs is encapsulated with the definitions of those fields. This approach, which relies heavily on SDL’s abstract data type and inheritance mechanisms, permits the behavior of the data-holding entities to be precisely defined, without obscuring process flow by adding in-line complexity to the individual state transitions.

The modeling of PDUs and SDUs requires sorts such as octet strings, and operators such as bitwise boolean functions, which are not predefined in SDL. These sorts and operators are defined in Package macsorts, which appears in Clause C.2.

Protocol and service data unit sorts are based on the **Bit** sort. Bit is a subtype of SDL’s predefined Boolean sort. As a result, Bit literals “0” and “1” are alternative names for “false” and “true,” and have no numeric significance. To use “0” or “1” as integer values requires a conversion operation. Items of the **Bitstring** sort are 0-origin, variable-length strings of Bits. With Bitstring operands, operators “and,” “or,” “xor,” and “not” operate bitwise, with the length of the result equal to the length of the longest (or only) source string. The **Octet** sort is a subtype of Bitstring that adds conversion operators to and from Integer. Each item of the Octet sort has length=8 {by usage convention in Z.100, enforced in Z.105}. Items of the **Octetstring** sort are 0-origin, variable-length strings of Octets. The **Frame** sort is a subtype of Octetstring that adds operators to extract and to modify all MAC header fields and most other MAC frame fields and elements. Most MAC fields and elements that contain named values with specific value assignments or enumerations are defined as subtypes of Frame, Octetstring, or Bitstring with the names added as literals or synonyms, so that the state machines can refer to the names without introducing ambiguity about the value encodings.

Where communication at a SAP or between processes is strictly first in first out (FIFO), the (implicit) input queue of the SDL processes is used. When more sophisticated queue management is needed, a queue whose entries are instances of one, specified sort is created using the **Queue** generator. Entries on Queue sorts may be added and removed at either the tail or the head, and the number of queue entries may be determined. The contents of a Queue may also be searched to locate entries with particular parameter values.

Clause C.2 contains an SDL-92 Package (a named collection of SDL definitions that can be included by reference into an SDL System specification), which is a formal description of the formats and data encodings used in IEEE 802.11 SDUs, PDUs, and the parameters of the service primitives used at each of the SAPs supported by the IEEE 802.11 MAC. This package also contains definitions for some data structures and operators used internally by one or more of the MAC state machines.

The behaviors of many intra-MAC operators are part of the normative description of the MAC protocol because results of the specified operations are visible, directly or indirectly, at exposed interfaces. For example, custom operators are used to define the generation of the CRC-32 value used in the FCS field (operator `crc32`, page 301), the calculation of frame transmission time used as part of the value in the Duration/ID field

in certain types of frames (operator `calcDur`, page 316), the comparison of the values of particular fields of a received MAC header with cached data values as part of the procedure for detecting duplicate frames (operator `searchTupleCache`, page 289), and numerous other aspects of frame formats and information encoding. On the other hand, data structures used solely for intra-MAC storage or for transferring of information between different state machines of a single station or access point, are only normative to the extent that they define items of internal state and the temporal sequence necessary for proper operation of the MAC protocol. The specific structures and encodings used for internal data storage and communication functions in this formal description do *not* constrain MAC implementations, provided those implementations exhibit the specified behaviors at the defined SAPs and, in conjunction with an appropriate PHY, on the wireless medium.

C.2 Data type and operator definitions for the MAC state machines

This clause is in SDL/PR (phrase notation), with the exception of procedural operators, which are defined in SDL/GR (graphic notation). Package `macsorts` contains the definitions of the sorts (data types with associated operators and literals) and synonyms (named constants) used by the MAC state machines. Package `macmib` defines data types for attributes in the MAC MIB, and portions of the PHY MIB, accessed by the MAC state machines. Package `macmib` exists solely to satisfy SDL's strong type checking in the absence of an SDL tool that fully supports Z.105 (the combined use of SDL with ASN.1).

Package macsorts

3101_dMacEnum(31)

```

/* PACKAGE MACSORTS */
/* This package contains definitions of the custom sorts (data types), operators,
  literals, and synonyms (named constants) used by the MAC state machines. */

```

```

/*****
 * Enumerated types used within the MAC state machines
 *****/
newtype ChangeType /* type of change due at the next boundary */
  literals dwell, /* dwell (only with FH PHY) */
           mocc; /* medium occupancy (only with PCF) */
endnewtype ChangeType;
newtype lmed /* priority for queuing MMPDUs, relative to MSDUs */
  literals head, /* place MMPDU at head of transmit queue */
           norm; /* place MMPDU at tail of transmit queue */
endnewtype lmed;
newtype NavSrc /* source of duration in SetNav & ClearNav signals */
  literals rts, /* RTS frame */
           cfpBss, cfpEndBss, /* start/end of CFP in own BSS */
           cfpOther, cfpEndOther, /* start/end of CFP in other BSS */
           cswitch, /* channel switch */
           misc, /* durlid from other frame types */
           nosrc; /* non-reception events */
endnewtype NavSrc;
newtype PsMode /* power save mode of a station (PsResponse signal) */
  literals sta_active, power_save, unknown; endnewtype PsMode;
newtype PsState /* power save state of this station */
  literals awake, doze; endnewtype PsState;
newtype StateErr /* requests disassoc or deauth (MmIndicate signal) */
  literals noerr, class2, class3; endnewtype StateErr;
newtype StationState /* asoc/auth state of sta (SsResponse signal) */
  literals not_auth, auth_open, auth_key, asoc, dis_asoc;
endnewtype StationState;
newtype TxResult /* transmission attempt status (PduConfirm signal) */
  literals successful, partial, retryLimit, txLifetime,
           atimAck, atimNak; endnewtype TxResult;

```

```

/*****
 * Enumerated types used in PHY service primitives
 *****/
newtype CcaStatus /* <state> parameter of PhyCca.indication */
  literals busy, idle; endnewtype CcaStatus;
newtype PhyRxStat /* <rxerror> parameter of PhyRxEnd.indication */
  literals no_error, fmt_violation, carrier_lost, unsupt_rate;
endnewtype PhyRxStat;

```

```

/*****
 * Placeholders for Mlme/Plme Get/Set Parameter Values
 *****/
/* MibAtrib (placeholder in MlmeGet/Set definitions) */
syntype MibAtrib = Charstring endsyntype MibAtrib;
/* MibValue (placeholder in MlmeGet/Set definitions) */
syntype MibValue = Integer endsyntype MibValue;

```



Package macsorts

3102_dLmeEnum(31)



```

*****
* Enumerated types used in Mac and Mlme service primitives
*****
newtype AuthType /* <authentication type> parm in Mlme primitives */
inherits Octetstring operators all;
adding literals open_system, shared_key;
axioms open_system == mkOS(0, 2); shared_key == mkOS(1, 2);
endnewtype AuthType;
newtype AuthTypeSet powerset( AuthType); endnewtype AuthTypeSet;
newtype BssType /* <BSS type> parameter & BSS description element */
literals infrastructure, independent, any_bss; endnewtype BssType;
newtype BssTypeSet powerset( BssType); endnewtype BssTypeSet;
newtype CfPriority /* <priority> parameter of various requests */
literals contention, contentionFree; endnewtype CfPriority;
newtype MibStatus /* <status> parm of Mlme/Plme Get/Set.confirm */
literals success, invalid, write_only, read_only;
endnewtype MibStatus;
newtype MlmeStatus /* <status> parm of Mlme operation confirm */
literals success, invalid, timeout, refused,
to_many_req, already_bss; endnewtype MlmeStatus;
newtype PwrSave /* <power save mode> parameter of MlmePowerMgt */
literals sta_active, power_save; endnewtype PwrSave;
newtype Routing /* <routing info> parameter for MAC data service */
literals null_rt; endnewtype Routing;
newtype RxStatus /* <reception status> parm of MaUnitdata indication */
literals rx_success, rx_failure; endnewtype RxStatus;
newtype ScanType /* <scan type> parameter of MlmeScan.request */
literals active_scan, passive_scan; endnewtype ScanType;
newtype ServiceClass /* <service class> parameter for MaUnitdata */
literals reorderable, strictlyOrdered; endnewtype ServiceClass;
newtype TxStatus /* <transmission status> parm of MaUnitdataStatus */
literals successful, retryLimit, txLifetime, noBss,
excessiveDataLength, nonNullSourceRouting,
unsupportedPriority, unavailablePriority,
unsupportedServiceClass, unavailableServiceClass,
unavailableKeyMapping; endnewtype TxStatus;

```




Package macsorts

3103_e\IntraMac(31)



```

/*****
 *      Intra-MAC remote variables (names of form mXYZ)
 *****/
remote mActingAsAp Boolean nodelay; /* =true if STA started BSS */
remote mAid AsocId nodelay; /* AID assigned to STA by AP */
remote mAssoc Boolean nodelay; /* =true if STA associated w/BSS */
remote mAtimW Boolean nodelay; /* =true if ATIM window in prog */
remote mBkIP Boolean nodelay; /* =true if backoff in prog */
remote mBrates Ratestring nodelay; /* basic rate set for this sta */
remote mBssId MacAddr nodelay; /* identifier of current (I)BSS */
remote mCap Octetstring nodelay; /* capability info from MlmeJoin */
remote mCfp Boolean nodelay; /* =true if CF period in progress */
remote mDisable Boolean nodelay; /* =true if not in any BSS; then */
/* TX only sends probe_req; RX only accepts beacon, probe_rsp */
remote mDtimCount Integer nodelay; /* =0 at Tbt of Beacon with DTIM */
remote mFxlP Boolean nodelay; /* =true during frame exchange seq */
remote mlbss Boolean nodelay; /* =true if STA is member of IBSS */
remote mListenInt Integer nodelay; /* beacons between wake up @TBTT */
remote mNavEnd Time nodelay; /* NAV end Time, <=now when idle */
remote mNextBdry Time nodelay; /* next boundary Time; =0 if none */
remote mNextTbt Time nodelay; /* Time next beacon due to occur */
remote mPcAvail Boolean nodelay; /* =true if point coord in BSS */
remote mPcDivr Boolean nodelay; /* =true if CF delivery only */
remote mPcPoll Boolean nodelay; /* =true if CF delivery & polling */
remote mPdly Usec nodelay; /* probe delay from start or join */
remote mPss PsState nodelay; /* power save state of STA */
remote mReceiveDTIMs Boolean nodelay; /* =true if DTIMs received */
remote mRxA Boolean nodelay; /* =true if RX indicated by PHY */
remote mSsId Octetstring nodelay; /* name of the current (I)BSS */
remote procedure TSF nodelay; /* read & update 64-bit TSF timer */
fpar Integer, Boolean; returns Integer;

```



Package macrosorts

3104_dStaticData(31)



```

*****
*   Named static data values (names of form sXYZ)
*****/
synonym sMaxMsduLng Integer = 2304; /* max octets in an MSDU */
synonym sMacHdrLng Integer = 24; /* octets in data header, no WEP */
synonym sWepHdrLng Integer = 28; /* octets in data header with WEP */
synonym sWepAddLng Integer = 8; /* octets added for WEP */
synonym sWdsAddLng Integer = 6; /* octets added for WDS (addr4) */
synonym sCrcLng Integer = 4; /* octets for crc32 (FCS, ICV) */
synonym sMaxMpduLng Integer = /* max octets in an MPDU */
(sMaxMsduLng + sMacHdrLng + sWdsAddLng + sWepAddLng + sCrcLng);
syntype FrameIndexRange = Integer /* index range for octets in MPDU */
constants 0 : sMaxMpduLng endsyntype FrameIndexRange;
synonym sTsOctet Integer = 24; /* first octet of Timestamp field */
synonym sMinFragLng Integer = 256; /* min value for aMpduMaxLength */
synonym sMaxFragNum Integer = /* maximum fragment number */
(sMaxMsduLng / (sMinFragLng - sMacHdrLng - sCrcLng));
synonym sAckCtsLng Integer = 112; /* bits in ACK and CTS frames */

```

```

*****
*   Station configuration flags (static, supplementary to MIB)
*****/
synonym sVersion Integer = 0; /* supported Protocol Version */
synonym sCanBeAp Boolean = false; /* =true if STA can operate as AP */
synonym sCanBePc Boolean = false; /* =true if AP can be Point Coord */
synonym sCfPollable Boolean = true; /* =true if responds to CF-polls */

```

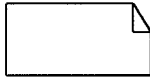
Package macsorts

3105_dUsec_TU(31)

```

/*****
 *   Discrete microsecond and Time Unit sorts
 *****/
/* SDL does not define the relationship between its concept */
/* of Time and physical time in the system being described. */
/* An abstraction is needed to establish this relationship, */
/* because Time in SDL uses the semantics of Real, whereas */
/* time in the MAC protocol is discrete, with the semantics */
/* of Natural and a step size (resolution) of 1 microsecond. */
/* Most MAC times are defined using the subtypes of Integer */
/* Usec and TU. These have operators for explicit conversion */
/* to SDL Time (tUsec, tTU), SDL Duration (dUsec, dTU), and */
/* from SDL Time (uTime, tuTime) as needed to comply with SDL's */
/* strong type checking. Where the MAC state machines need to */
/* access the contents of the TSF timer, SDL's 'now' (current */
/* time) is used. This yields readable time-dependent code, */
/* but the value of 'now' cannot be modified by an SDL program, */
/* so adopting the TSF time from timestamps in received Beacons */
/* or Probe Responses is shown as an informal task symbol. */
/* Microsecond sort -- also has operators tmin and tmax */
newtype Usec inherits Integer operators all;
adding operators
dUsec : Usec -> Duration;
tUsec : Usec -> Time;
uTime : Time -> Usec;
tmax : Usec, Usec -> Usec;
tmin : Usec, Usec -> Usec;
axioms  for all u, w in Usec(
  u >= w ==> tmax(u, w) == u;   u < w ==> tmax(u, w) == w;
  u >= w ==> tmin(u, w) == w;   u < w ==> tmin(u, w) == u;
  for all t in Time( for all r in Real(
    r = float(u) ==> tUsec(u) == Time!(Duration!(r));
    t = Time!(Duration!(r)) and u = fix(r) ==> u == uTime(t));
  for all d in Duration( for all r in Real(
    r = float(u) ==> dUsec(u) == Duration!(r); ));
  constants >= 0 /* constrain value range to be non-negative */
endnewtype Usec;
/* Time Unit sort -- (1 * TU) = (1024 * Usec) */
newtype TU inherits Integer operators all;
adding operators
dTU : TU -> Duration;
tTU : TU -> Time;
tuTime : Time -> TU;
u2TU : Usec -> TU;
tu2U : TU -> Usec;
axioms  for all k in TU( for all t in Time( for all r in Real(
  r = float(k) ==> tTU(k) == Time!(Duration!(1024 * r));
  t = Time!(Duration!(r)) and k = (fix(r) / 1024) ==> k == tuTime(t));
  for all d in Duration( for all r in Real(
    r = float(k) ==> dTU(k) == Duration!(1024 * r); ));
  for all u in Usec( u2TU(u) == u / 1024; tu2U(k) == k * 1024; );
  constants >= 0 /* constrain value range to be non-negative */
endnewtype TU;

```



Package macsorts

3106_dString0(31)

```

*****
* Generator for 0-origin String sorts (adapted from Z.105, Annex A)
*****
/* String0(sort, nullSymbol) can define strings of any sort. */
/* These strings are indexed starting from 0 rather than 1. */
/* Sorts defined by String0 have the normal String operators, plus */
/* Tail (all but first item), Head (all but last item), and */
/* aggregators S2, S3, S4, S6, S8 (make fixed length strings). */
generator String0(type Item, literal Emptystring)
  literals Emptystring;
  operators
    MkString : Item -> String0; /* make a string from an item */
    Length : String0 -> Integer; /* length of string */
    First : String0 -> Item; /* first item in string */
    Tail : String0 -> String0; /* all but first item in string */
    Last : String0 -> Item; /* last item in string */
    head : String0 -> String0; /* all but last item in string */
    "/" : String0, String0 -> String0; /* concatenation */
    Extract! : String0, Integer -> Item; /* get item from string */
    Modify! : String0, Integer, Item -> String0; /* modify string */
    SubStr : String0, Integer, Integer -> String0;
      /* SubStr(s,i,j) is string0 of length j starting at string0(i) */
    S2 : Item, Item -> String0; S3 : Item, Item, Item -> String0;
    S4 : Item, Item, Item, Item -> String0;
    S6 : Item, Item, Item, Item, Item, Item -> String0;
    S8 : Item, Item, Item, Item, Item, Item, Item, Item -> String0;
  /* axioms continued on next page... */

endgenerator String0;

```



Package macsorts

3107_aString0(31)



```

/* String0 axioms */
/* for all item0,item1,item2,item3,item4,item5,item6,item7 in Item(
for all s, s1, S2, S3 in String0( for all i, j in Integer(
constructors are Emptystring, MkString, and "/";
equalities between constructor terms
s // Emptystring == s;      Emptystring // s == s;
(s1 // S2) // S3 == s1 // (S2 // S3);
definition of Length by applying it to all constructors
type String Length(Emptystring) == 0;
type String Length(MkString(item0)) == 1;
type String Length(s1 // S2) == Length(s1) + Length(S2);
definition of Extract! by applying it to all constructors,
Extract!(MkString(item0), 0) == item0;
i < Length(s1) ==> Extract!(s1 // S2, i) == Extract!(s1, i);
i >= Length(s1) ==> Extract!(s1 // S2, i) == Extract!(S2, i - Length(s1));
i < 0 or i >= Length(s) ==> Extract!(s, i) == error!;
definition of First and Last by other operations
First(s) == Extract!(s, 0);
Last(s) == Extract!(s, Length(s) - 1);
definition of substr(s,i,j) by induction on j,
i >= 0 and i <= Length(s) ==> SubStr(s, i, 0) == Emptystring;
i >= 0 and j > 0 and i + j <= Length(s) ==> SubStr(s, i, j) ==
SubStr(s, i, j - 1) // MkString(Extract!(s, i + j - 1));
i < 0 or j < 0 or i + j > Length(s) ==> SubStr(s, i, j) == error!;
definition of Modify!, Head, Tail, Sx by other operations
Modify!(s, i, item0) == SubStr(s, 0, i) // MkString(item0) //
SubStr(s, i + 1, Length(s) - i - 1);
head(s) == SubStr(s, 0, Length(s) - 1);
Tail(s) == SubStr(s, 1, Length(s) - 1);
S2(item0, item1) == MkString(item0) // MkString(item1);
S3(item0, item1, item2) ==
MkString(item0) // MkString(item1) // MkString(item2);
S4(item0, item1, item2, item3) ==
MkString(item0) // MkString(item1) // MkString(item2) //
MkString(item3);
S6(item0, item1, item2, item3, item4, item5) ==
MkString(item0) // MkString(item1) // MkString(item2) //
MkString(item3) // MkString(item4) // MkString(item5);
S8(item0, item1, item2, item3, item4, item5, item6, item7) ==
MkString(item0) // MkString(item1) // MkString(item2) //
MkString(item3) // MkString(item4) // MkString(item5) //
MkString(item6) // MkString(item7); ));
*/

```



Package macrosorts

3108_dBitstring(31)

```

*****
*   ASN.1-style BIT sort (from Z.105, Annex A)
*****
/* Bit is a subtype of Boolean -- bit values 0 and 1 are
/* not numerals and cannot be used with Integer operators */
newtype Bit inherits Boolean
  literals 0 = false, 1 = true; operators all; endnewtype Bit;

```

```

*****
*   ASN.1-style BIT STRING sort (adapted from Z.105, Annex A)
*****
/* Bitstrings are 0-origin strings of Bit. Z.105 uses ASN.1-style */
/* literals in binary ('1011'B) or hexadecimal ('D3'H), but this */
/* syntax is not accepted for Z.100 string literals. Therefore, */
/* this version provides only hexadecimal literals 0x00-0xFF. */
/* Bitstring operators '='>, 'not', 'and', 'or', and 'xor' act */
/* bitwise, with the length of the result string equal to the */
/* length of the longest (or only) source string. */
newtype Bitstring String0(Bit, "")
  adding literals macro Hex_Literals;
  operators
    "not" : Bitstring -> Bitstring;
    "and" : Bitstring, Bitstring -> Bitstring;
    "or"  : Bitstring, Bitstring -> Bitstring;
    "xor" : Bitstring, Bitstring -> Bitstring;
    ">"   : Bitstring, Bitstring -> Bitstring;  noequality;
  axioms macro Hex_Axioms;
  for all s, s1, S2, S3 in Bitstring(
    s = s == true;    s1 = S2 == S2 = s1;
    s1 /= S2 == not (s1 = S2);    s1 = S2 == true ==> s1 == S2;
    ((s1 = S2) and (S2 = S3)) ==> s1 = S3 == true;
    ((s1 = S2) and (S2 /= S3)) ==> s1 = S3 == false;
  for all b, b1, b2 in Bit(
    not (") == ";
    not (MkString(b) // s) == MkString(not (b)) // not (s);
    " and " == ";
    Length(s) > 0 ==> " and s == MkString(0) and s;
    Length(s) > 0 ==> s and " == s and MkString(0);
    (MkString(b1) // s1) and (MkString(b2) // S2) ==
      MkString(b1 and b2) // (s1 and S2);
    s1 or S2 == not (not s1 and not S2);
    s1 xor S2 == (s1 or S2) and not (s1 and S2);
    s1 => S2 == not (not s1 and S2););
  map for all b1, b2 in Bitstring literals(
    for all bs1, bs2 in Charstring literals(
/* connection to the String generator */
    for all b in Bit literals(
      spelling(b1) = "" // bs1 // bs2 // "",
      spelling(b2) = "" // bs2 // "", spelling(b) = bs1
      ==> b1 == MkString(b) // b2; ));
endnewtype Bitstring;

```



Package macsorts

3109_d\Octetstring(31)

```

*****
*   OCTET sort (influenced by Z.105, Annex A)
*****
/* Octet is a subtype of Bitstring where length always =8. */
/* Z.105 adds a "size" keyword to SDL and defines Octet with */
/* "... constants size (8) ..." to impose this length constraint. */
/* Here Octet relies on proper use maintain lengths as multiples */
/* of 8. Proper length strings are created by the hexadecimal */
/* Bitstring literals (e.g. 0xD5) and operator mkOctet: */
/* o:= mkOctet(i) converts a non-negative Integer (mod 256) */
/* to an Octet (exactly 8 bits) */
/* i:= octetVal(o) converts an Octet to an Integer (0:255) */
/* o:= flip(o) reverses bit order of the Octet */
/* (0<-->7, 1<-->6, 2<-->5, 3<-->4) */
newtype Octet inherits Bitstring operators all;
adding operators
mkOctet : Integer -> Octet;
octetVal : Octet -> Integer;
flip : Octet -> Octet;
axioms
for all i in Integer( for all z in Octet(
i = 0 ==> mkOctet(i) == S8(0, 0, 0, 0, 0, 0, 0, 0);
i = 1 ==> mkOctet(i) == S8(1, 0, 0, 0, 0, 0, 0, 0);
i > 1 and i <= 255 ==> mkOctet(i) ==
SubStr((First(mkOctet(i mod 2)) // mkOctet(i / 2)), 0, 8);
i > 255 ==> mkOctet(i) == mkOctet(i mod 256);
i < 0 ==> mkOctet(i) == error!;
z = MkString(0) ==> octetVal(z) == 0;
z = MkString(1) ==> octetVal(z) == 1;
Length(z) > 1 and Length(z) <= 8 ==>
octetVal(z) == octetVal(First(z)) +
(2 * (octetVal(SubStr(z, 1, Length(z) - 1))));
Length(z) > 8 ==> octetVal(z) == error!;
flip(z) == S8(z(7),z(6),z(5),z(4),z(3),z(2),z(1),z(0)); ));
endnewtype Octet;

```

Package macsorts
3109.1_a\Octetstring(31)

```

/*****
*   OCTET STRING sort (somewhat influenced by Z.105, Annex A)
*****/
/* Octetstrings are 0-ORIGIN strings of Octet, NOT 1-ORIGIN */
/* strings like Octet_String in Z.105 (hence the name change). */
/* Octetstring has conversion operators to and from Bitstring, */
/* and integer to Octetstring. Octetstring literals are "null" */
/* and 1-4, 6, 8 item 0x00 strings O1, O2, O3, O4, O6, O8. */
newtype Octetstring String0(Octet, null)
adding literals O1, O2, O3, O4, O6, O8;
operators
  B_S : Octetstring -> Bitstring; /* name changed from Z.105 */
  O_S : Bitstring -> Octetstring; /* name changed from Z.105 */
  mkOS : Integer,Integer -> Octetstring; /* mkOS(i1,i2) returns */
      /* mkstring(mkOctet(i1)) padded (0x00) to length i2 */
  mk2octets : Integer -> Octetstring; /* 16-bit int to 2-octets */
axioms
for all b, b1, b2 in Bitstring(
  for all s in Octetstring(
    for all o in Octet(
      B_S(null) == null;    O_S(null) == null;
      B_S(MkString(o) // s) == o // B_S(s);
      Length(b1) > 0, Length(b1) < 8 ==>
        O_S(b1) == MkString(b1 or 0x00); /* expand b1 to 8 bits */
      b == b1 // b2, Length(b1) = 8 ==>
        O_S(b) == MkString(b1) // O_S(b2);
    for all i, k in Integer(
      k = 1 ==> mkOS(i, k) == MkString(mkOctet(i));
      k > 1 ==> mkOS(i, k) == mkOS(i, k - 1) // MkString(0x00);
      k <= 0 ==> error!;
      mk2octets(i) == MkString(mkOctet(i mod 256)) //
        MkString(mkOctet(i / 256)); );
    O1 == MkString(0x00);    O2 == O1 // O1;
    O3 == O2 // O1;        O4 == O2 // O2;
    O6 == O4 // O2;        O8 == O4 // O4; ));
map for all O1, O2 in Octetstring literals(
  for all b1, b2 in Bitstring literals(
    spelling(O1) = spelling(b1), spelling(O2) = spelling(b2)
    ==> O1 = O2 == b1 = b2; ));
endnewtype Octetstring;

```




Package macsorts

3110_dMacAddr(31)

```

/*****
 *   MAC Address sorts
 *****/
/* MacAddr is a subtype of Octetstring with added operators: */
/* isGroup(m) =true if given a group address */
/* isBcst(m) =true if given the broadcast address */
/* isLocal(m) =true if given a locally-administered address */
/* adrOs(m) converts MacAddr to Octetstring */
/* MAC addresses must be defined to be exactly 6 octets long, */
/* typically using the S6 operator or nullAddr synonym. */
newtype MacAddr inherits Octetstring operators all;
adding operators
isGroup : MacAddr -> Boolean;
isBcst  : MacAddr -> Boolean;
isLocal : MacAddr -> Boolean;
adrOs   : MacAddr -> Octetstring;
axioms
for all m in MacAddr(
  (Length(m) = 6) and ((Extract!(m,0) and 0x01) = 0x01) ==> isGroup(m) == true;
  (Length(m) = 6) and ((Extract!(m,0) and 0x01) = 0x00) ==> isGroup(m) == false;
  (Length(m) = 6) and (m = S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF)) ==> isBcst == true;
  (Length(m) = 6) and (m /= S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF)) ==> isBcst == false;
  (Length(m) = 6) and ((Extract!(m,0) and 0x02) = 0x02) ==> isLocal == true;
  (Length(m) = 6) and ((Extract!(m,0) and 0x02) = 0x00) ==> isLocal == false;
  Length(m) /= 6 ==> error! /* common error! term */;
  for all o in Octetstring(m = MacAddr!(o) == adrOs(m) = o; ));
endnewtype MacAddr;
newtype MacAddrSet powerset( MacAddr) endnewtype MacAddrSet;
synonym bcstAddr MacAddr = /* Broadcast Address */
  <<type MacAddr>> S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF);
synonym nullAddr MacAddr = /* Null Address */
  <<type MacAddr>> S6(0x00,0x00,0x00,0x00,0x00,0x00);

```

```

/*****
 *   BSS description sorts
 *****/
/* BssDscr is used with MlmeScan.confirm and MlmeJoin.request */
newtype BssDscr struct
bdBssId MacAddr;
bdSsid  Octetstring; /* 1 <= length <= 32 */
bdType  BssType;
bdBcnPer TU; /* beacon period in Time Units */
bdDtimPer Integer; /* DTIM period in beacon periods */
bdTstamp Octetstring; /* 8 Octets from ProbeRsp/Beacon */
bdStartTs Octetstring; /* 8 Octets TSF when rx Tstamp */
bdPhyParms PhyParms; /* empty if not needed by PHY */
bdCfParms  CfParms; /* empty if not CfPollable/no PCF */
bdIbssParms IbssParms; /* empty if infrastructure BSS */
bdCap Capability; /* capability information */
bdBrates Ratestring; /* BSS basic rate set */
endnewtype BssDscr;
newtype BssDscrSet powerset( BssDscr) endnewtype BssDscrSet;

```



Package macsorts

3111_d\TupleCache(31)



```

*****
* Duplicate filtering support sorts
*****/
syntype FragNum = Integer /* Range of possible fragment numbers */
constants 0:sMaxFragNum endsyntype FragNum;
syntype SeqNum = Integer /* Range of possible sequence numbers */
constants 0:4095 endsyntype SeqNum;
newtype Tuple struct /* for duplicate filtering & defragmentation */
full Boolean; /* =true if Tuple contains valid info */
ta MacAddr; /* transmitting station address (Addr2) */
sn SeqNum; /* Msdu/Mmpdu sequence number */
fn FragNum; /* most recent MpdU fragment number */
tRx Time; /* reception time (endRx of fragment) */
default (, false, nullAddr, 0, 0, 0.);
endnewtype Tuple;
    
```

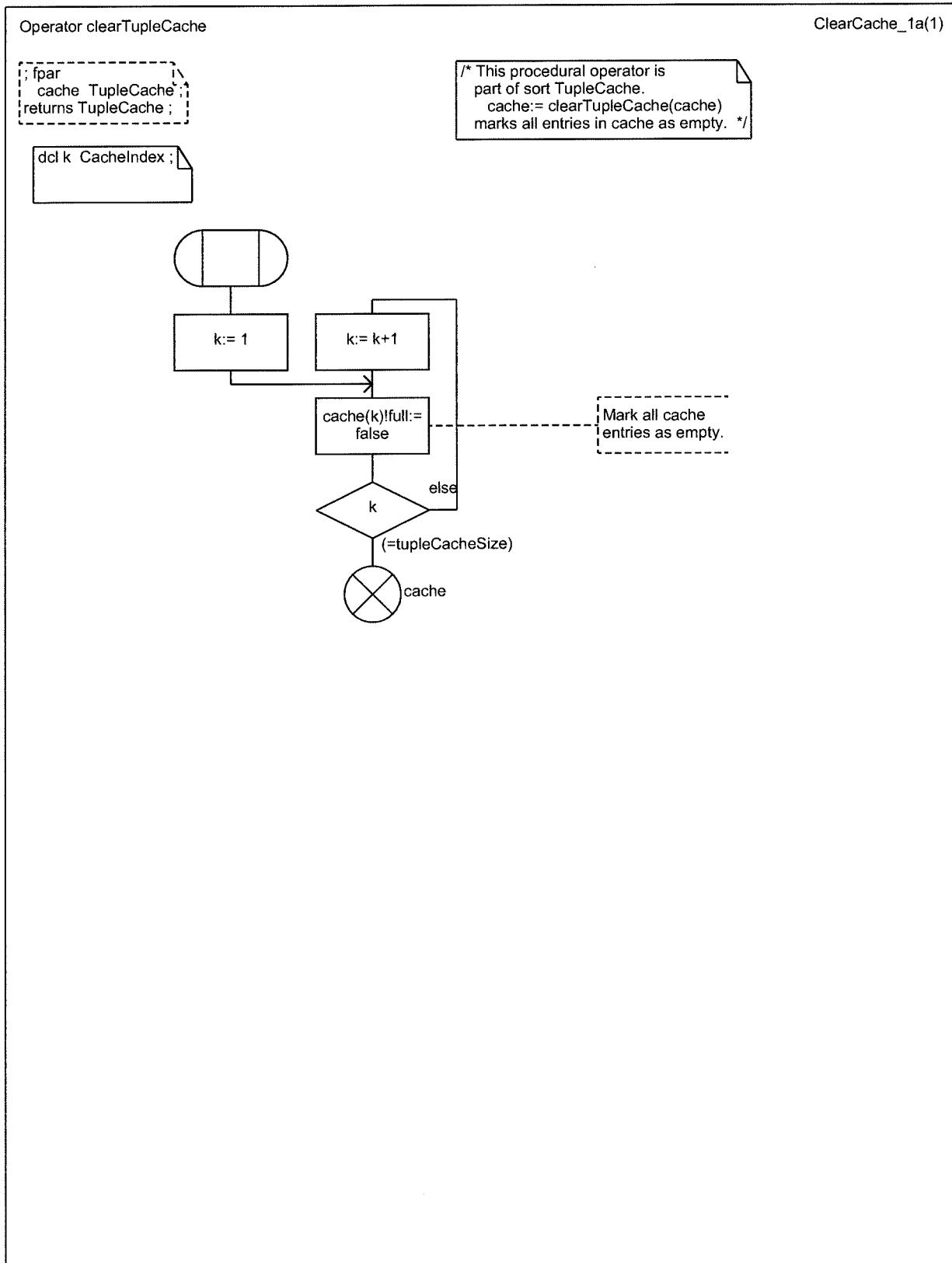
operator
clearTupleCache

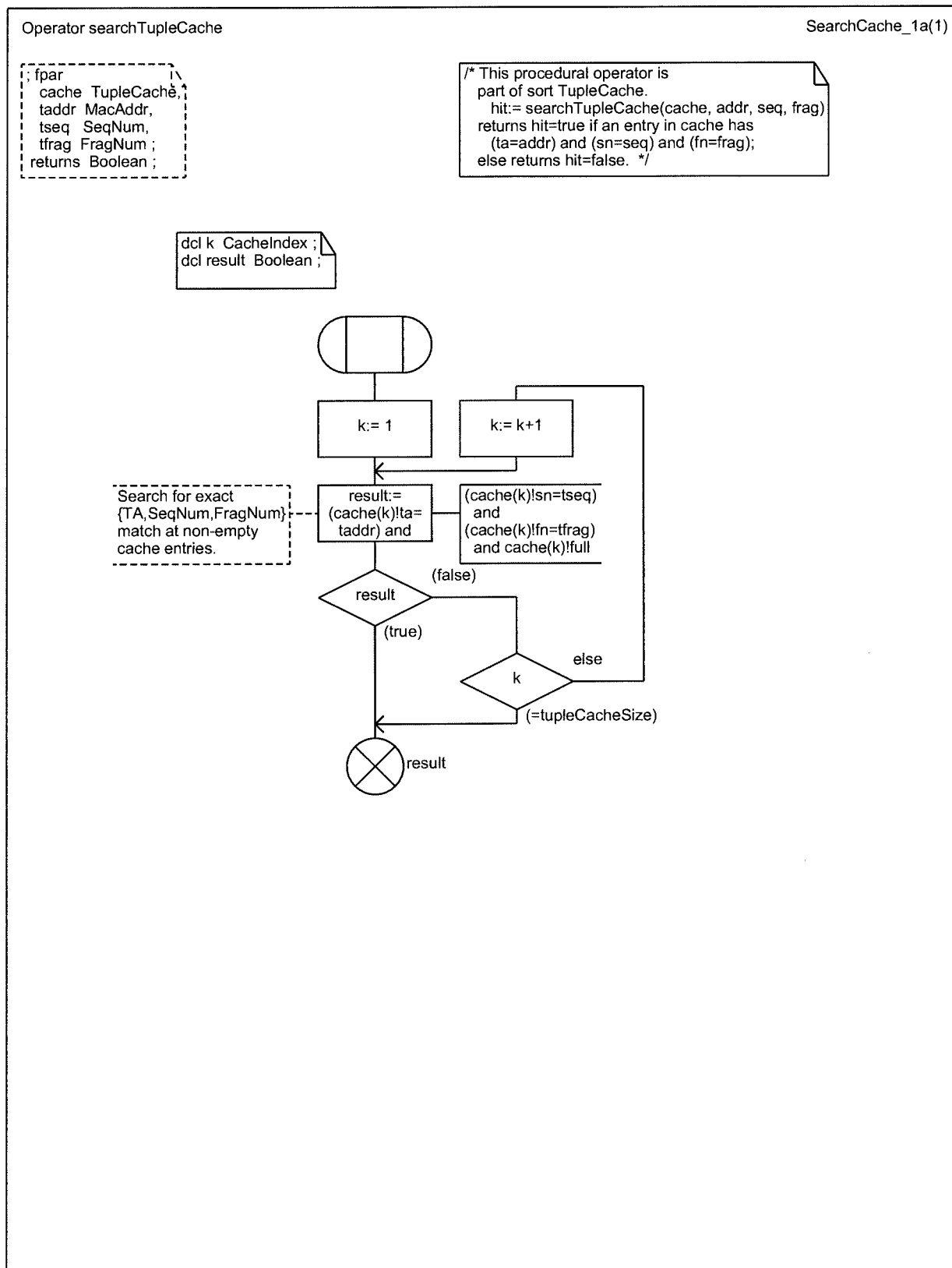
operator
searchTupleCache

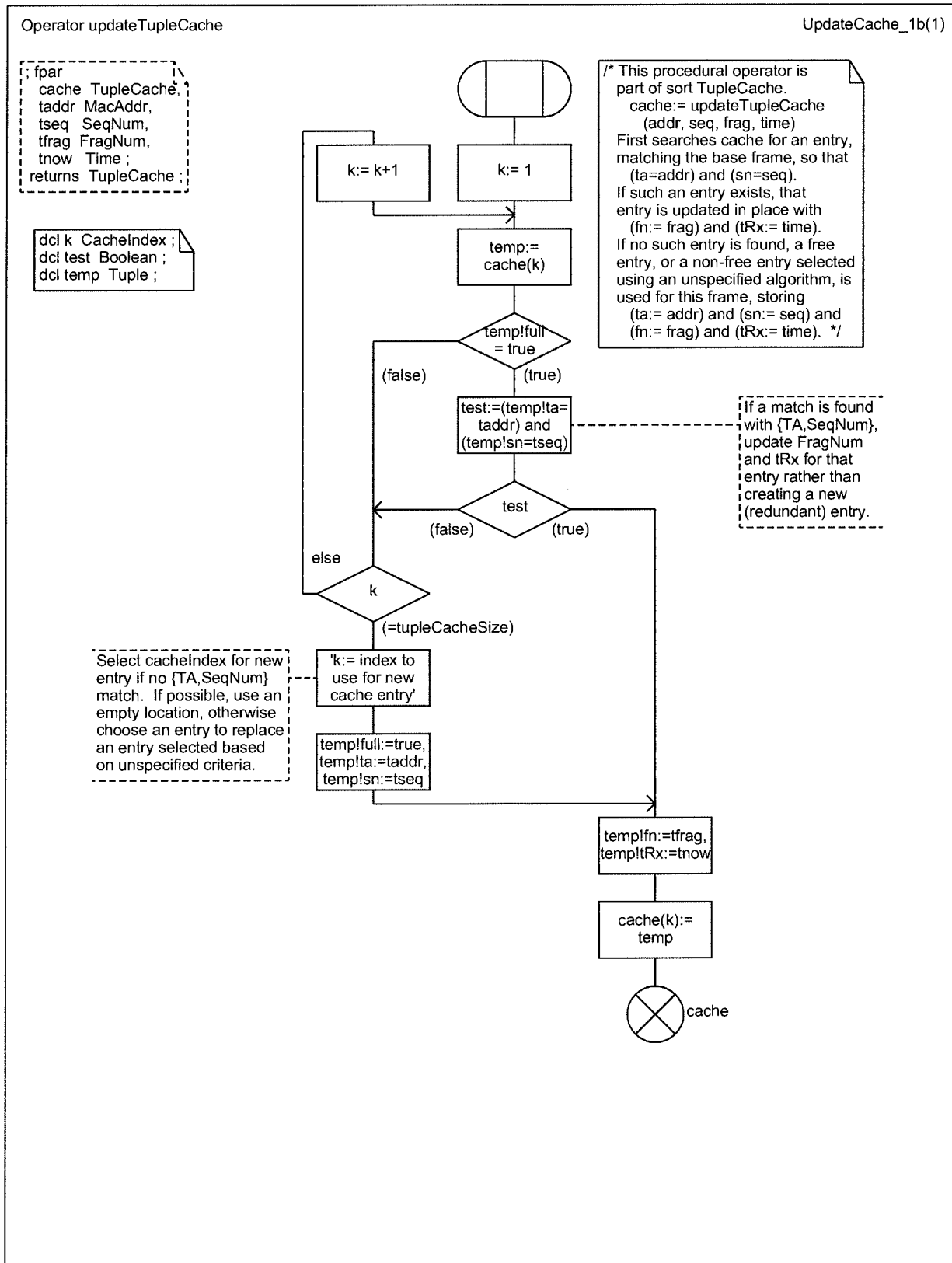
operator
updateTupleCache

```

*****
* TupleCache support sorts
*****/
/* Number of TupleCache entries and associated index range */
synonym tupleCacheSize Integer = 32; /* this value is an example,
TupleCache size is implementation dependent */
syntype CacheIndex = Integer constants 1:tupleCacheSize
endsyntype CacheIndex;
/* TupleCache array */
/* cache:= ClearTupleCache(cache) to initialize cache */
/* cache:= UpdateTupleCache(cache, addr, seq, frag, endRx) */
/* if <addr,seq> is already cached, updates frag */
/* if <addr,seq> not cached, fills an empty entry */
/* or replaces an entry using an unspecified algorithm */
/* SearchTupleCache(cache, addr, seq, frag) */
/* returns true if specified <addr,seq,frag> in cache */
newtype TupleCache Array( CacheIndex, Tuple);
adding operators
ClearTupleCache : TupleCache -> TupleCache;
SearchTupleCache : TupleCache, MacAddr, SeqNum, FragNum -> Boolean;
UpdateTupleCache : TupleCache, MacAddr, SeqNum, FragNum, Time ->
TupleCache;
operator ClearTupleCache;
fpar cache TupleCache; returns TupleCache; referenced;
operator SearchTupleCache;
fpar cache TupleCache, taddr MacAddr, tseq SeqNum, tfrag FragNum;
returns Boolean; referenced;
operator UpdateTupleCache;
fpar cache TupleCache, taddr MacAddr, tseq SeqNum, tfrag FragNum,
tnow Time; returns TupleCache; referenced;
endnewtype TupleCache;
    
```









Package macsorts

3112_dCounter(31)

```

/*****
 * 32-bit Counter sort and Integer string sort
 *****/
/* This sort used for MIB counters, needed because SDL Integers */
/* have no specified maximum value. inc(counter) increments the */
/* counter value by 1, with wraparound from (2^32)-1 to 0. */

newtype Counter32 inherits Integer operators all;
adding operators
  inc : Counter32 -> Counter32;
axioms
  for all c in Counter32 (
    c < 4294967295 ==> inc(c) == c + 1;
    c >= 4294967295 ==> inc(c) == 0; );
endnewtype Counter32;
/* String (1-origin) of Integer */
newtype Intstring String( Integer, nolnt); endnewtype Intstring;

```

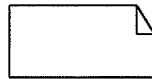
Package macsorts

3113_d\Queue(31)

```

*****
* Generator for Queue sorts
*****
/* The Queue generator is derived from the String0 generator */
/* to create Queues of any sort. Queues operators are: */
/* Qfirst(queue,item) adds item as the first queue element */
/* Qlast(queue,item) adds item as the last queue element */
/* and the String0 operators Length, //, First, Last, Head, Tail */
/* Because operators can only return a single value, removing an */
/* element from a queue is a 2-step process: */
/* dequeue first: item:=First(queue); queue:=Tail(queue); */
/* dequeue last: item:=Last(queue); queue:=Head(queue); */
generator Queue(type Item, literal Emptyqueue)
literals Emptyqueue;
operators
MkQ : Item -> Queue; /* make a queue from an item */
Length : Queue -> Integer; /* number of items on queue */
First : Queue -> Item; /* first item on queue */
Qfirst : Queue,Item -> Queue; /* add item as first on queue */
Tail : Queue -> Queue; /* all but first item on queue */
Last : Queue -> Item; /* last item on queue */
Qlast : Queue,Item -> Queue; /* add item as last on queue */
head : Queue -> Queue; /* all but last item on queue */
"//" : Queue, Queue -> Queue; /* concatenation */
Extract! : Queue,Integer -> Item; /* copy item from queue */
Modify! : Queue,Integer,Item -> Queue; /* modify item in queue */
SubQ : Queue,Integer,Integer -> Queue;
/* SubQ(q,i,j) queue of length j starting from queue(i) */
axioms
for all item0 in Item( for all q, q1, q2, q3 in Queue(
for all i, j in Integer(
/* constructors are Emptyqueue, MkQueue, and "//"; */
/* equalities between constructor terms */
q // Emptyqueue == q; Emptyqueue // q == q;
(q1 // q2) // q3 == q1 // (q2 // q3);
/* definition of Length by applying it to all constructors */
type Queue Length(Emptyqueue) == 0;
type Queue Length(MkQueue(item0)) == 1;
type Queue Length(q1 // q2) == Length(q1) + Length(q2);
/* definition of Extract! by applying it to all constructors, */
Extract!(MkQueue(item0), 0) == item0;
i < Length(q1) ==> Extract!(q1 // q2, i) == Extract!(q1, i);
i >= Length(q1) ==> Extract!(q1 // q2, i) == Extract!(q2, i - Length(q1));
i < 0 or i >= Length(q) ==> Extract!(q, i) == error!;
/* definition of First and Last by other operations */
First(q) == Extract!(q, 0); Last(q) == Extract!(q, Length(q) - 1);
/* definition of SubQ(q,i,j) by induction on j, */
i >= 0 and i <= Length(q) ==> SubQ(q, i, 0) == Emptyqueue;
i >= 0 and j > 0 and i + j <= Length(q) ==> SubQ(q, i, j) ==
SubQ(q, i, j - 1) // MkQueue(Extract!(q, i + j - 1));
i < 0 or j < 0 or i + j > Length(q) ==> SubQ(q,i,j) == error!;
/* define Modify!, Head, Tail, Qfirst, Qlast by other ops */
Modify!(q, i, item0) == SubQ(q, 0, i) //
MkQueue(item0) // SubQ(q, i + 1, Length(q) - i - 1);
head(q) == SubQ(q, 0, Length(q) - 1);
Tail(q) == SubQ(q, 1, Length(q) - 1);
Qfirst(q, item0) == MkQueue(item0) // q;
Qlast(q, item0) == q // MkQueue(item0); )))
endgenerator Queue;

```



Package macsorts

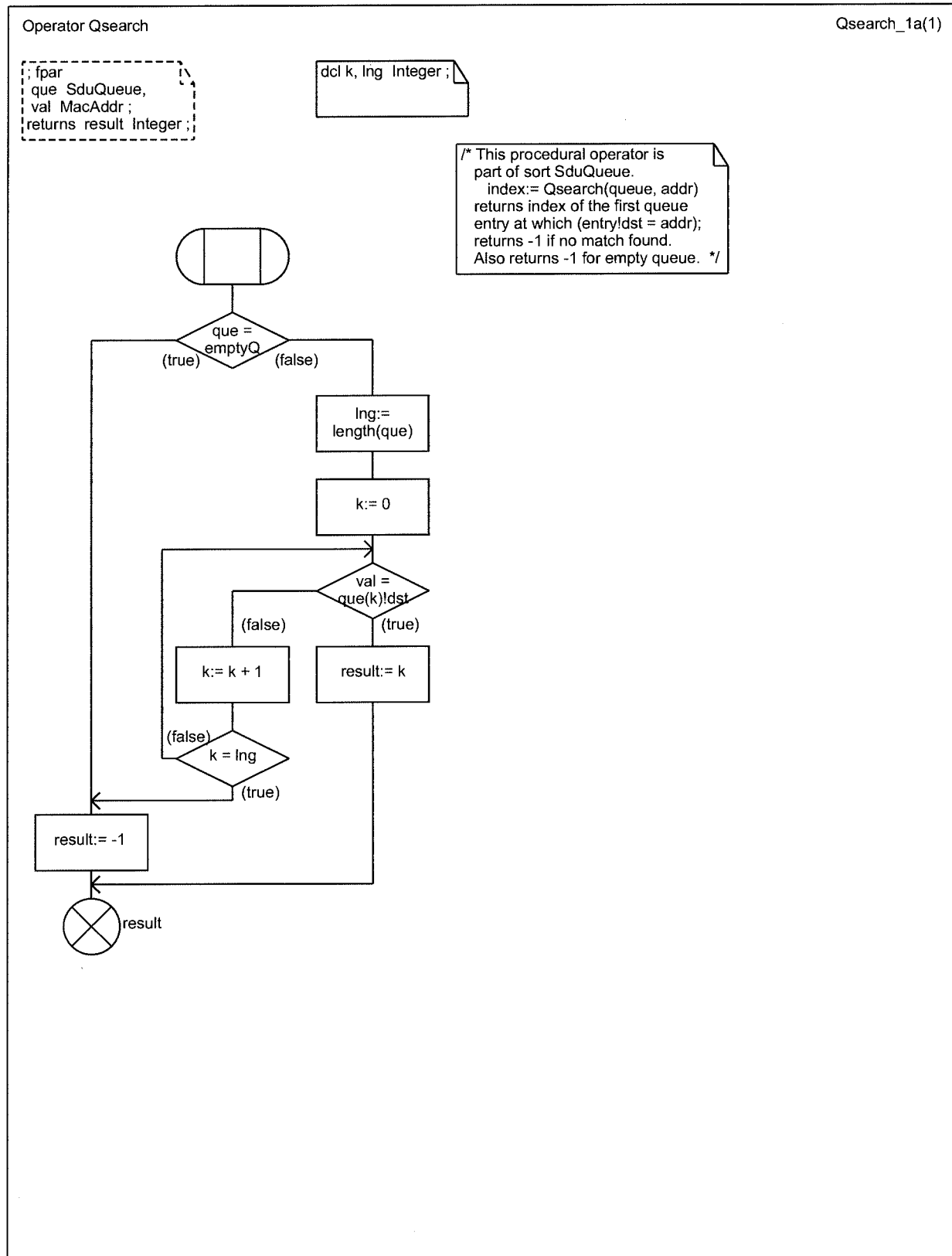
3114_d\Fragment(31)

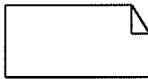
operator
qSearch

```

/*****
*   Fragmentation support sorts
*****/
/* Array to hold up to FragNum fragments of an MsdU/Mmpdu */
newtype FragArray Array(FragNum, Frame); endnewtype FragArray;
/* FragSdu structure is for OUTGOING MSDUs/MMPDUs (called SDUs) */
/* Each SDU, even if not fragmented, is held in an instance of */
/* this structure awaiting its (re)transmission attempt(s). */
/* Transmit queue(s) are ordered lists of FragSdu instances. */
newtype FragSdu struct
  fTot  FragNum; /* number of fragments in pdu FragArray */
  fCur  FragNum; /* next fragment number to send */
  fAnc  FragNum; /* next fragment to announce in ATIM or TIM
                 when fAnc > fCur, pdu(fCur)+ may be sent */
  eol   Time; /* set to (now + dUsec(aMaxTxMsdULifetime))
              when the entry is created */
  sqf   SeqNum; /* SDU sequence number, set at 1st Tx attempt */
  src   Integer; /* short retry counter for this SDU */
  lrc   Integer; /* long retry counter for this SDU */
  dst   MacAddr; /* destination address */
  grpa  Boolean; /* =true if RA (not DA) is a group address */
  psm   Boolean; /* =true if RA (not DA) may be in pwr_save */
  resume Boolean; /* =true if fragment burst being resumed */
  cnfTo Pld; /* address to which confirmation is sent */
  txrate Rate; /* data rate used for initial fragment */
  cf    CfPriority; /* requested priority (from LLC) */
  pdu   FragArray; /* array of Frame to hold fragments */
endnewtype FragSdu;
/* Queue of FragSdu */
/* for power save buffers, etc., searchable with Qsearch operator: */
/* index:= Qsearch(queue, addr) where queue is an SduQueue, */
/* index identifies the first queue entry at which */
/* entry!dst = addr; or as -1 if no match (or queue empty). */
newtype SduQueue Queue(FragSdu, emptyQ);
adding operators
  qSearch : SduQueue, MacAddr -> Integer;
operator qSearch;
  fpar que SduQueue, val MacAddr; returns Integer; referenced;
endnewtype SduQueue;

```



Package macsorts

3115_dDefragment(31)



operator
ArAge

operator
ArFree

operator
ArSearch

```

*****
*   Defragmentation support sorts
*****
/* The PartialSdu structure is for INCOMPLETE MSDUs/MMPDUs */
/* (generically SDUs) for which at least 1 fragment has been */
/* received. Unfragmented SDUs are reported upward immediately, */
/* and are never stored in instances of this structure. */
newtype PartialSdu struct
  inUse Boolean; /* =true if this instance holds any fragments */
  rta MacAddr; /* transmitting station (Addr2) */
  rsn SeqNum; /* SDU sequence number */
  rCur FragNum; /* fragment number of most recent Mpdu */
  reol Time; /* (now+dUsec(aMaxReceiveLifetime) @ 1st Mpdu */
  rsdu Frame; /* buffer where Mpdus are concatenated */
  default (. false, nullAddr, 0, 0, 0, null .);
endnewtype PartialSdu;
newtype PartialSduKeys struct /* if aPrivacyOptionImplemented=true */
  wDefKeys KeyVector; /* default keys when 1st frag received */
  wKeyMap KeyMapArray; /* key mappings when 1st frag received */
  wExclude Boolean; /* aExcludeUnencrypted @ 1st frag rx */
endnewtype PartialSduKeys;
/* Number of entries in defragmentation array at this station. */
/* The value is implementation dependent (min=3, see 9.5). */
synonym defragSize Integer = 6;
syntype defragIndex = Integer constants 1:defragSize
endsyntype defragIndex;
/* Array of PartialSdu for use defragmenting Msdus and Mmpdus. */
/* Searchable using the ArSearch operator */
/* index:= ArSearch(array, addr, seq, frag) */
/* where index is returned to identify the first element for which */
/* ((inUse = true) and (entry!rta = addr) and (entry!rsn = seq) */
/* and (entry!rCur = (frag-1))); or as =1 if no match found. */
/* index:= ArFree(array) returns the index of a free entry, */
/* or -1 if no entries free. May free an entry, selected using */
/* an unspecified algorithm, to avoid returning -1. */
/* array:= ArAge(array, age) */
/* frees where (entry!eol < age), also used to clear array. */
newtype DefragArray Array( defragIndex, PartialSdu);
adding operators
  ArSearch : DefragArray, MacAddr, SeqNum, FragNum -> Integer;
  ArFree : DefragArray -> Integer;
  ArAge : DefragArray, Time -> DefragArray;
operator ArSearch;
  fpar ar DefragArray, adr MacAddr, seq SeqNum, frg FragNum;
  returns Integer; referenced;
operator ArFree; fpar ar DefragArray; returns Integer; referenced;
operator ArAge; fpar ar DefragArray, age Time;
  returns DefragArray; referenced;
endnewtype DefragArray;
newtype DefragKeysArray Array( defragIndex, PartialSduKeys);
endnewtype DefragKeysArray;

```

Operator ArAge

ArAge_1a(1)

```

; fpar
ar DefragArray,
age Time;
returns DefragArray;
    
```

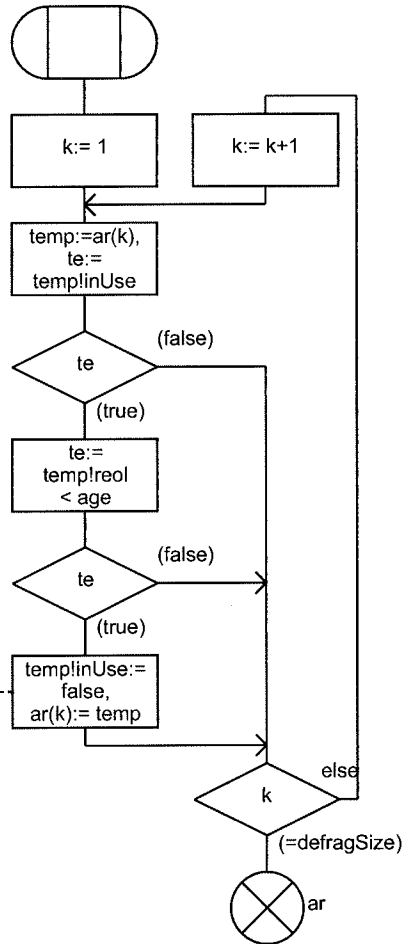
```

/* This procedural operator
is part of sort DefragArray.
array:= ArAge(array, age)
frees entryleol < age. This is
used both for the aging function
and to clear the DefragArray. */
    
```

```

dcl k DefragIndex;
dcl te Boolean;
dcl temp PartialSdu;
    
```

Mark all entries with end-of-life (reol) earlier than specified as not in use.



Operator ArFree

ArFree_1b(1)

```

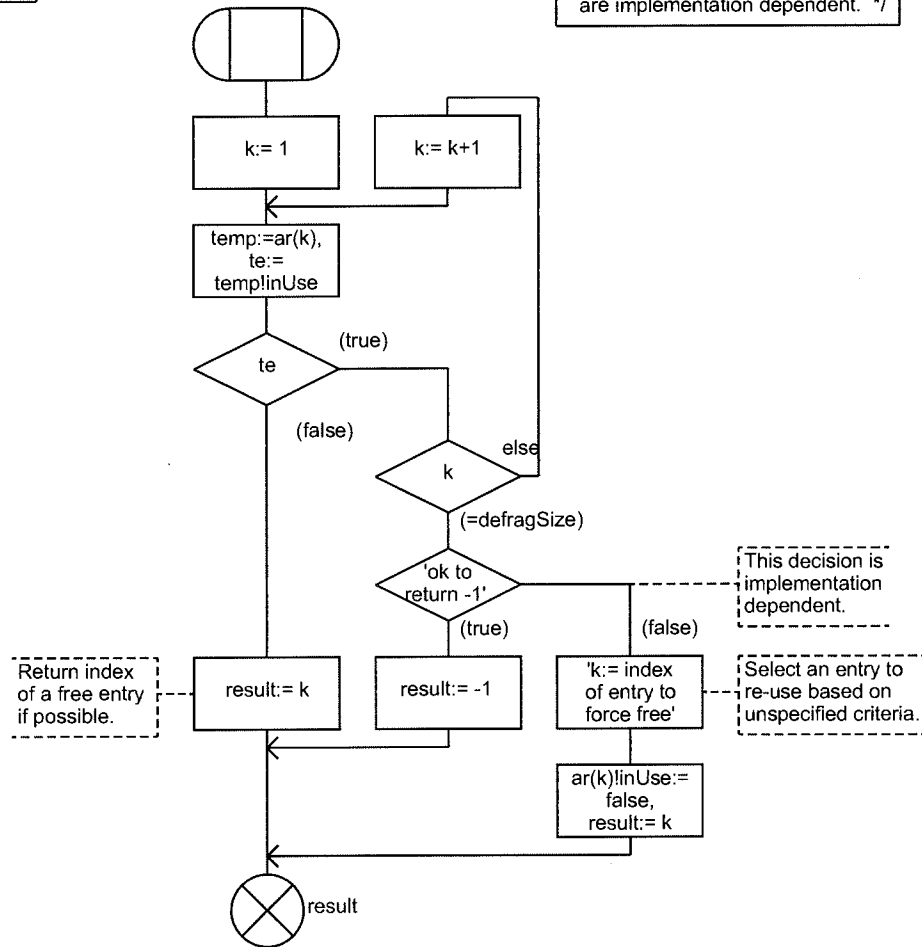
; fpar
ar DefragArray ;
returns Integer ;
    
```

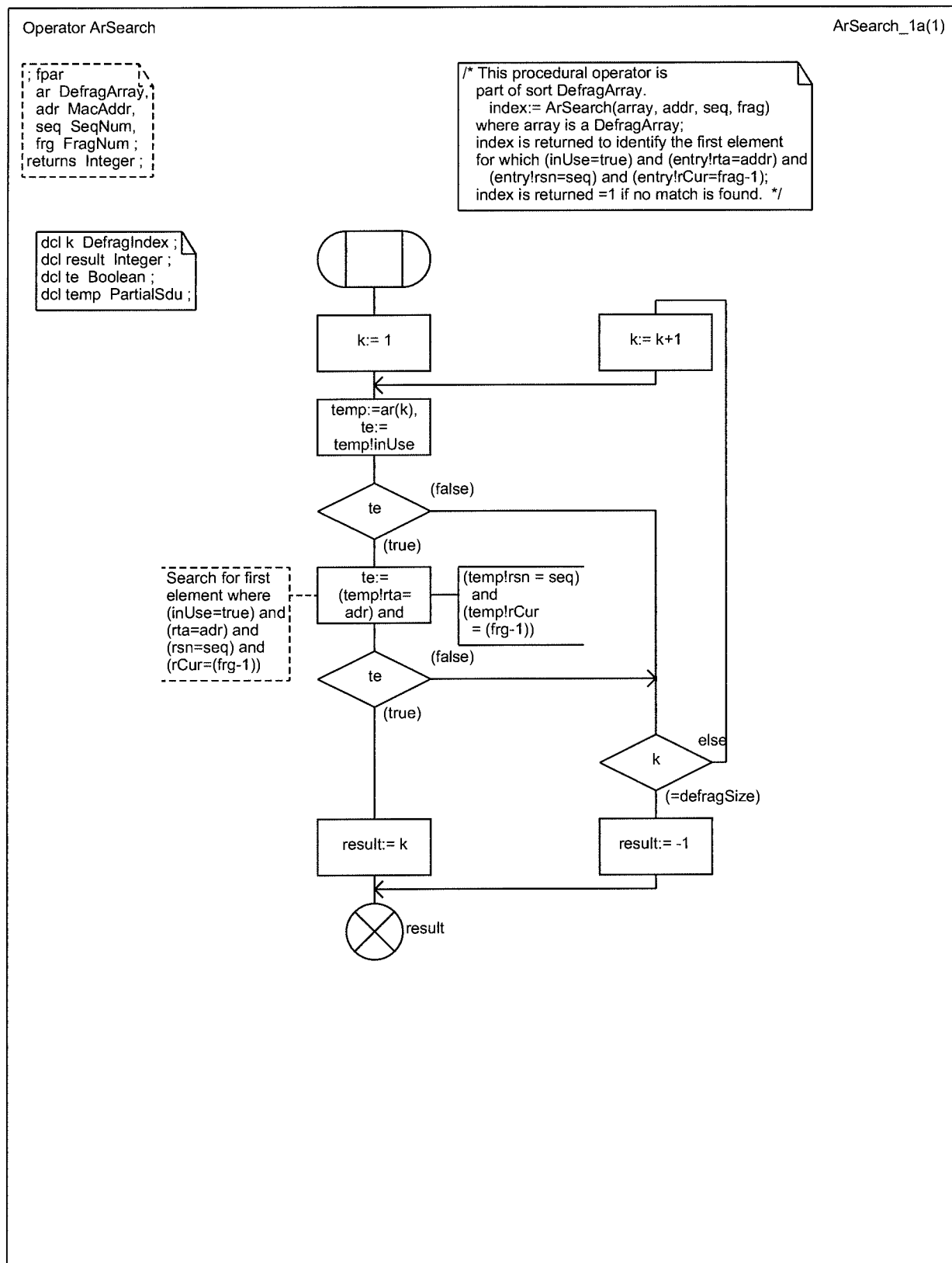
```

dcl k DefragIndex ;
dcl result Integer ;
dcl te Boolean ;
dcl temp PartialSdu ;
    
```

```

/* This procedural operator is
part of the sort DefragArray.
index:= ArFree(array)
returns index of an unused entry
in the array. If all entries are used,
either returns -1, or selects an
arbitrary entry to free in order to
return a usable index. Decision
criteria for case of no free entries
are implementation dependent. */
    
```







Package macsorts

3116_d\Crc_Wep(31)



operator
crc32

```

*****
* CRC-32 sorts (for FCS and ICV)
*****
/* Crc is a subtype of Octetstring with added operators: */
/* crc:= Crc32(crc,octet) */
/* updates the crc value to include the new octet, and */
/* Mirror(crc), which returns a Crc value with the order */
/* of the octets, and of the bits in each octet, reversed for */
/* MSb-first transmission (see 7.1.1). Crc variables must have */
/* exactly 4 octets, which is done using initCrc or S4. */
newtype Crc inherits Octetstring operators all;
adding operators
  Crc32 : Crc, Octet -> Crc;
  mirror : Crc -> Octetstring;
operator Crc32; fpar crcin Crc, val Octet; returns Crc; referenced;
axioms for all c in Crc(
  mirror(c) == S4(flip(c(3)),flip(c(2)),flip(c(1)), flip(c(0))); );
endnewtype Crc;
synonym initCrc Crc = /* Initial Crc value (all 1s) */
  << type Crc>> S4(0xFF,0xFF,0xFF,0xFF);
synonym goodCrc Crc = /* Unique remainder for valid CRC-32 */
  << type Crc>> S4(0x7B,0xDD,0x04,0xC7);

```

operator
keyLookup

```

*****
* WEP support sorts
*****
syntype KeyIndex = Integer constants 0:3 endsyntype KeyIndex;
newtype PrngKey inherits Octetstring operators all;
adding literals nullKey; /* nullKey is not any of 2^40 key values */
axioms nullKey == null; default nullKey; endnewtype PrngKey;
newtype KeyVector /* vector of default WEP keys */
  Array( KeyIndex, PrngKey); endnewtype KeyVector;
/* Number of entries in aWepKeyMappings array at this station.
/* implementation dependent value, minimum=10 (see 8.3.2). */
synonym sWepKeyMappingLength Integer = 10;
syntype KeyMappingRange = Integer
  constants 1:sWepKeyMappingLength endsyntype KeyMappingRange;
newtype KeyMap struct /* structure used for entries in KeyMapArray */
  mappedAddr MacAddr;
  wepOn Boolean;
  wepKey PrngKey;
endnewtype KeyMap;
/* KeyMapArray -- used for aWepKeyMapping table; */
/* an array of KeyMap indexed by KeyMappingRange, with operator */
/* KeyMap := keyLookup(addr, keyMapArray, keyMapArrayLength) */
/* returns the KeyMap entry for the specified addr, or */
/* (. nullAddr, false, nullKey .) if no mapping for addr. */
newtype KeyMapArray Array( KeyMappingRange, KeyMap);
adding operators
  keyLookup : MacAddr, KeyMapArray, Integer -> KeyMap;
operator keyLookup;
fpar luadr MacAddr, kma KeyMapArray, kml Integer;
returns KeyMap; referenced;
endnewtype KeyMapArray;

```

Operator Crc32

crc32_1a(1)

```

; fpar
  crcin Crc,
  val Octet ;
returns Crc ;
    
```

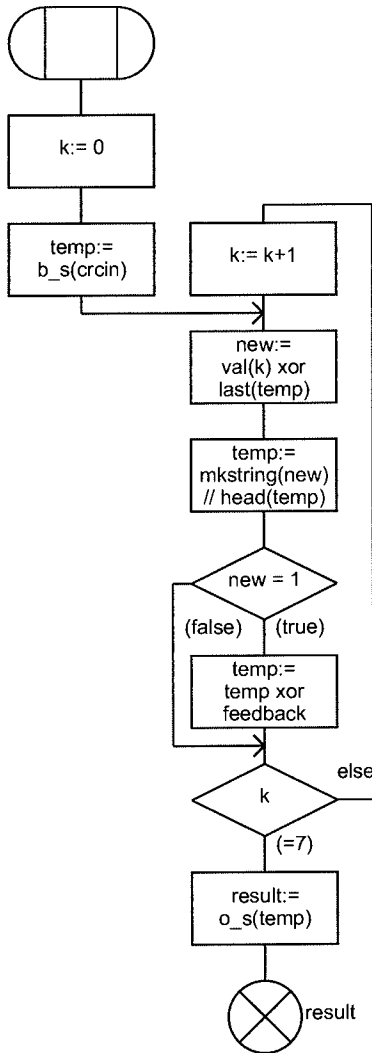
```

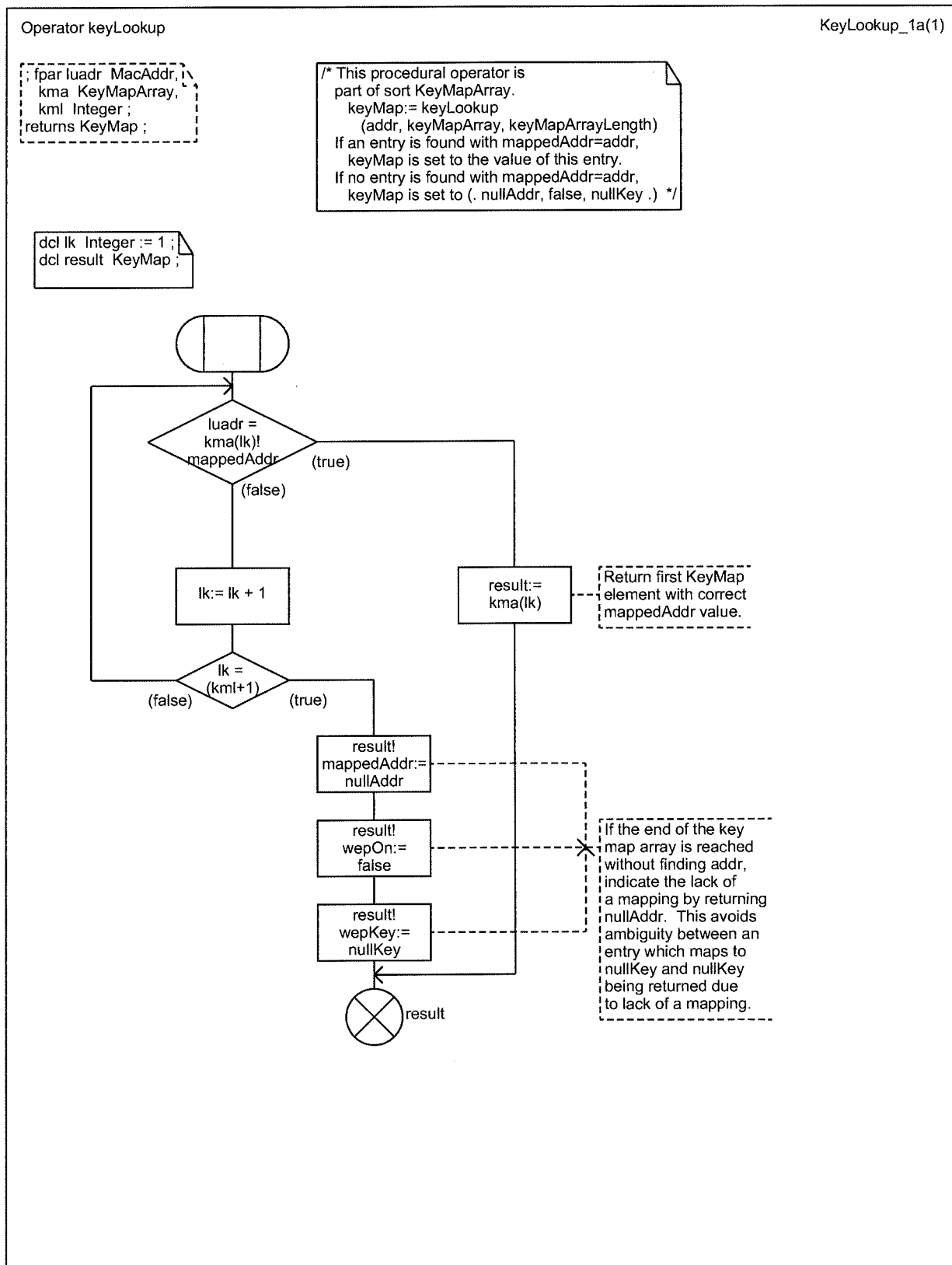
/* This procedural operator is
part of sort Crc.
  crc:= Crc32(crc, octet)
generates CRC-32 polynomial,
LSb-first, for the 8 bits of
octet into accumulator crc. */
    
```

```

dcl k Integer ;
dcl new Bit ;
dcl result Crc ;
dcl temp Bitstring ;

/* Bitstring with 1s at bit
positions with feedback
terms in CRC-32 polynomial */
synonym feedback Bitstring =
  S8(0,1,1,0,1,1,0,1) //
  S8(1,0,1,1,1,0,0,0) //
  S8(1,0,0,0,0,0,1,1) //
  S8(0,0,1,0,0,0,0,0) ;
    
```







Package macsorts

3117_d\Frame_1(31)



```

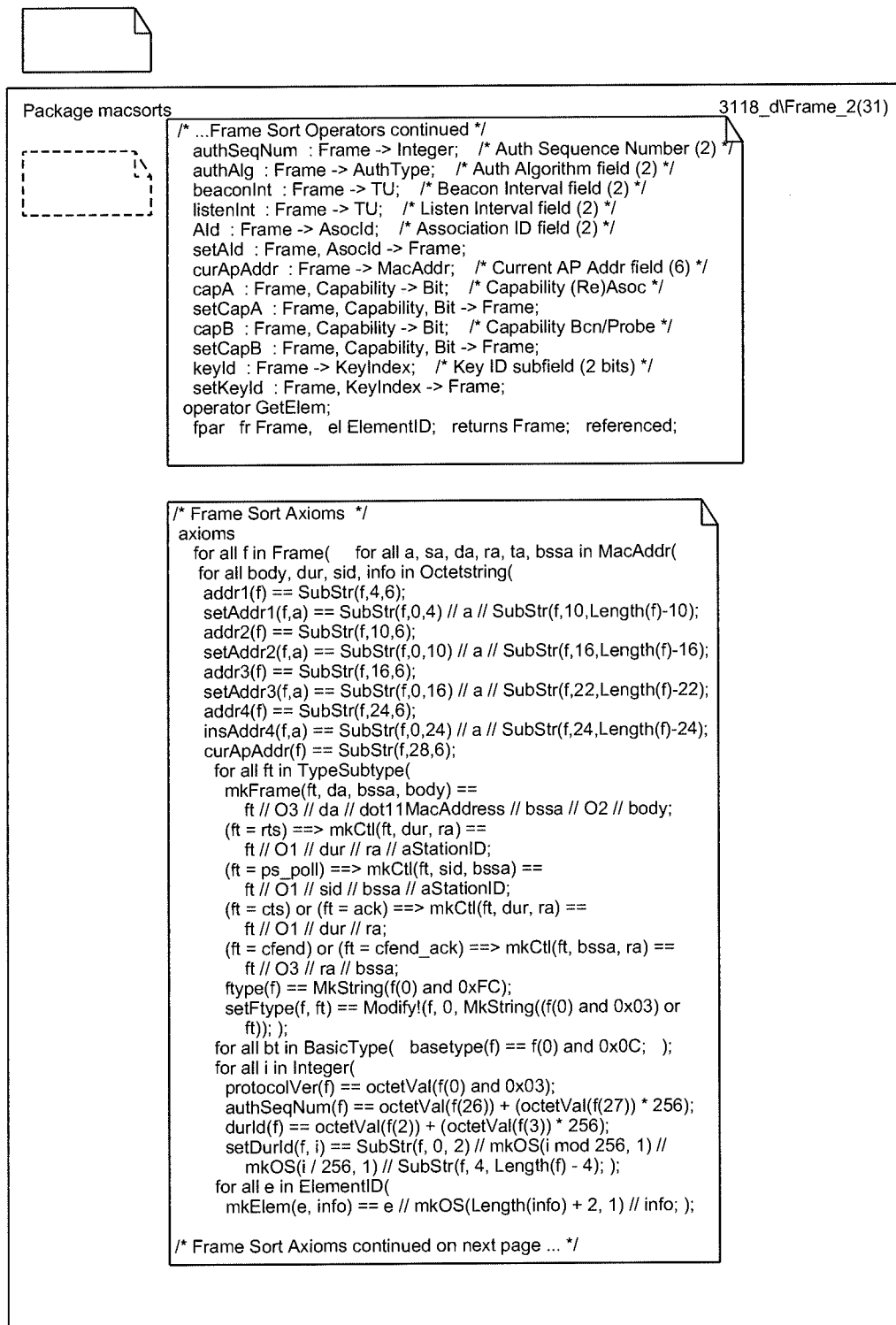
/*****
 *   FRAME sort (the basic definition of fields in MAC frames)
 *****/
/* Frame is a subtype of Octetstring with operators for creating
/* MAC headers, extracting each of the header fields and some
/* management frame fields, and modifying most of these fields.
/* There are operators to create and extract management frame
/* elements, but no operators for the frame body, IV, ICV, and FCS
/* fields, which are handled directly as Octetstrings. */
newtype Frame inherits Octetstring operators all;
adding operators
mkFrame : TypeSubtype, MacAddr, MacAddr, Octetstring -> Frame;
mkCtl : TypeSubtype, Octetstring, MacAddr -> Frame;
protocolVer : Frame -> Integer; /* Protocol version (2 bits) */
basetype : Frame -> BasicType; /* Type field (2 bits) */
ftype : Frame -> TypeSubtype; /* Type & Subtype (6 bits) */
setFtype : Frame, TypeSubtype -> Frame;
toDs : Frame -> Bit; /* To DS bit (1 bit) */
setToDs : Frame, Bit -> Frame;
frDs : Frame -> Bit; /* From DS bit (1 bit) */
setFrDs : Frame, Bit -> Frame;
moreFrag : Frame -> Bit; /* More Fragments bit (1 bit) */
setMoreFrag : Frame, Bit -> Frame;
retryBit : Frame -> Bit; /* Retry bit (1 bit) */
setRetryBit : Frame, Bit -> Frame;
pwrMgt : Frame -> Bit; /* Power Management bit (1 bit) */
setPwrMgt : Frame, Bit -> Frame;
moreData : Frame -> Bit; /* More Data bit (1 bit) */
setMoreData : Frame, Bit -> Frame;
wepBit : Frame -> Bit; /* WEP bit (1 bit) */
setWepBit : Frame, Bit -> Frame;
orderBit : Frame -> Bit; /* {strictly}Order(ed) (1 bit) */
setOrderBit : Frame, Bit -> Frame;
durl : Frame -> Integer; /* Duration/ID field (2) */
setDurl : Frame, Integer -> Frame;
addr1 : Frame -> MacAddr; /* Address 1 [DA/RA] field (6) */
setAddr1 : Frame, MacAddr -> Frame;
addr2 : Frame -> MacAddr; /* Address 2 [SA/TA] field (6) */
setAddr2 : Frame, MacAddr -> Frame;
addr3 : Frame -> MacAddr; /* Address 3 [Bss/DA/SA] field */
setAddr3 : Frame, MacAddr -> Frame;
addr4 : Frame -> MacAddr; /* Address 4 [WDS-SA] field (6) */
insAddr4 : Frame, MacAddr -> Frame;
seq : Frame -> SeqNum; /* Sequence Number (12 bits) */
setSeq : Frame, SeqNum -> Frame;
frag : Frame -> FragNum; /* Fragment Number (4 bits) */
setFrag : Frame, FragNum -> Frame;
ts : Frame -> Time; /* Timestamp field (8) */
setTs : Frame, Time -> Frame;
mkElem : ElementID, Octetstring -> Frame; /* make element */
GetElem : Frame, ElementID -> Frame; /* get element if aval */
status : Frame -> StatusCode; /* Status Code field (2) */
setStatus : Frame, StatusCode -> Frame;
authStat : Frame -> StatusCode; /* Status Code in Auth frame */
reason : Frame -> ReasonCode; /* Reason Code field (2) */

/* Frame operators continued on next page ...*/

```

operator
getElem

Gets element
from body of
Management
frame. If the
target element
is not present
an Octetstring
of length zero
is returned.





Package macsorts

3119_d\Frame_3(31)

```

/* ... Frame Sort Axioms continued */
for all b in Bit(
  toDs(f) == if (f(1) and 0x01) then 1 else 0 fi;
  setToDs(f, b) ==
    Modify!(f, 1, (f(1) and 0xFE) or S8(0,0,0,0,0,0,b));
  frDs(f) == if (f(1) and 0x02) then 1 else 0 fi;
  setFrDs(f, b) ==
    Modify!(f, 1, (f(1) and 0xFD) or S8(0,0,0,0,0,0,b));
  moreFrag(f) == if (f(1) and 0x04) then 1 else 0 fi;
  setMoreFrag(f, b) ==
    Modify!(f, 1, (f(1) and 0xFB) or S8(0,0,0,0,0,0,b));
  retryBit(f) == if (f(1) and 0x08) then 1 else 0 fi;
  setRetryBit(f, b) ==
    Modify!(f, 1, (f(1) and 0xF7) or S8(0,0,0,0,0,0,b));
  pwrMgt(f) == if (f(1) and 0x10) then 1 else 0 fi;
  setPwrMgt(f, b) ==
    Modify!(f, 1, (f(1) and 0xFB) or S8(0,0,0,b,0,0,0));
  moreData(f) == if (f(1) and 0x20) then 1 else 0 fi;
  setMoreData(f, b) ==
    Modify!(f, 1, (f(1) and 0xFB) or S8(0,0,b,0,0,0,0));
  wepBit(f) == if (f(1) and 0x40) then 1 else 0 fi;
  setWepBit(f, b) ==
    Modify!(f, 1, (f(1) and 0xFB) or S8(0,b,0,0,0,0,0));
  orderBit(f) == if (f(1) and 0x80) then 1 else 0 fi;
  setOrderBit(f, b) ==
    Modify!(f, 1, (f(1) and 0xFB) or S8(b,0,0,0,0,0,0));
  for all c in Capability(
    capA(f,c) == if (B_S(SubStr(f,24,2)) and c) then 1 else 0 fi;
    setCapA(f,c,b) == SubStr(f,0,24) // (B_S(SubStr(f,24,2) and
      (not c)) or (if b then c else 02 fi)) //
      SubStr(f,26,Length(f) - 26);
    capB(f,c) == if (B_S(SubStr(f,34,2)) and c) then 1 else 0 fi;
    setCapB(f,c,b) == SubStr(f,0,34) // (B_S(SubStr(f,34,2) and
      (not c)) or (if b then c else 02 fi)) //
      SubStr(f,36,Length(f) - 36); );
  for all sq in SeqNum(
    seq(f) == (octetVal(f(22) and 0xF0)/16)+(octetVal(f(23)*16));
    setSeq(f, sq) == SubStr(f, 0, 22) // MkString((f(22) and 0x0F)
      or mkOctet((sq mod 16) * 16)) // mkOS(sq / 16, 1) //
      SubStr(f, 24, Length(f) - 24); );
  for all fr in FragNum(
    frag(f) == octetVal(f(22) and 0x0F);
    setFrag(f, fr) ==
      SubStr(f, 0, 22) // MkString((f(22) and 0xF0) or
      mkOctet(fr)) // SubStr(f, 23, Length(f) - 23); );
  for all tm in Time(
    ts(f) == tUsec( Usec!(octetVal(f(24)) +
      (256 * (octetVal(f(25)) +
      (256 * (octetVal(f(26)) +
      (256 * (octetVal(f(27)) +
      (256 * (octetVal(f(28)) +
      (256 * (octetVal(f(29)) +
      (256 * (octetVal(f(30)) +
      (256 * octetVal(f(31)))))))))))))))); );
/* Frame Sort Axioms continued on next page ... */

```



Package macsorts

3120_d\Frame_4(31)

```

/* ... Frame Sort Axioms continued */
setTs(f, tm) == SubStr(f, 0, 24) // mkOS(fix(tm), 1) //
mkOS((fix(tm) / 256), 1) // mkOS((fix(tm) / 65536), 1) //
mkOS((fix(tm) / 16777216), 1) //
mkOS((fix(tm) / 4294967296), 1) //
mkOS(((fix(tm) / 4294967296) / 256), 1) //
mkOS(((fix(tm) / 4294967296) / 65536), 1) //
mkOS(((fix(tm) / 4294967296) / 16777216), 1) //
SubStr(f, 32, Length(f) - 32);
for all stat in StatusCode(
  status(f) == SubStr(f, 26, 2);
  setStatus(f, stat) ==
    SubStr(f, 0, 26) // stat // SubStr(f, 28, Length(f) - 28);
  authStat(f) == SubStr(f, 28, 2); );
for all rea in ReasonCode( reason(f) == SubStr(f, 24, 2); );
for all alg in AuthType( AuthType(f) == SubStr(f, 24, 2); );
for all u in TU(
  beaconInt(f) == octetVal(f(32)) + (octetVal(f(33)) * 256);
  listenInt(f) == octetVal(f(26)) + (octetVal(f(27)) * 256); );
for all sta in AssocId(
  Aid(f) == octetVal(f(28)) + (octetVal(f(29)) * 256);
  setAid(f, sta) == SubStr(f, 0, 28) // mkOS(sta mod 256, 1) //
  mkOS(sta / 256, 1) // SubStr(f, 30, Length(f) - 30); );
for all kid in KeyIndexRange(
  keyId(f) == octetVal(f(27)) / 64;
  setKeyId(f, kid) == Modify!(f, 27, mkOS(kid * 64)); ); );
endnewtype Frame;

```

```

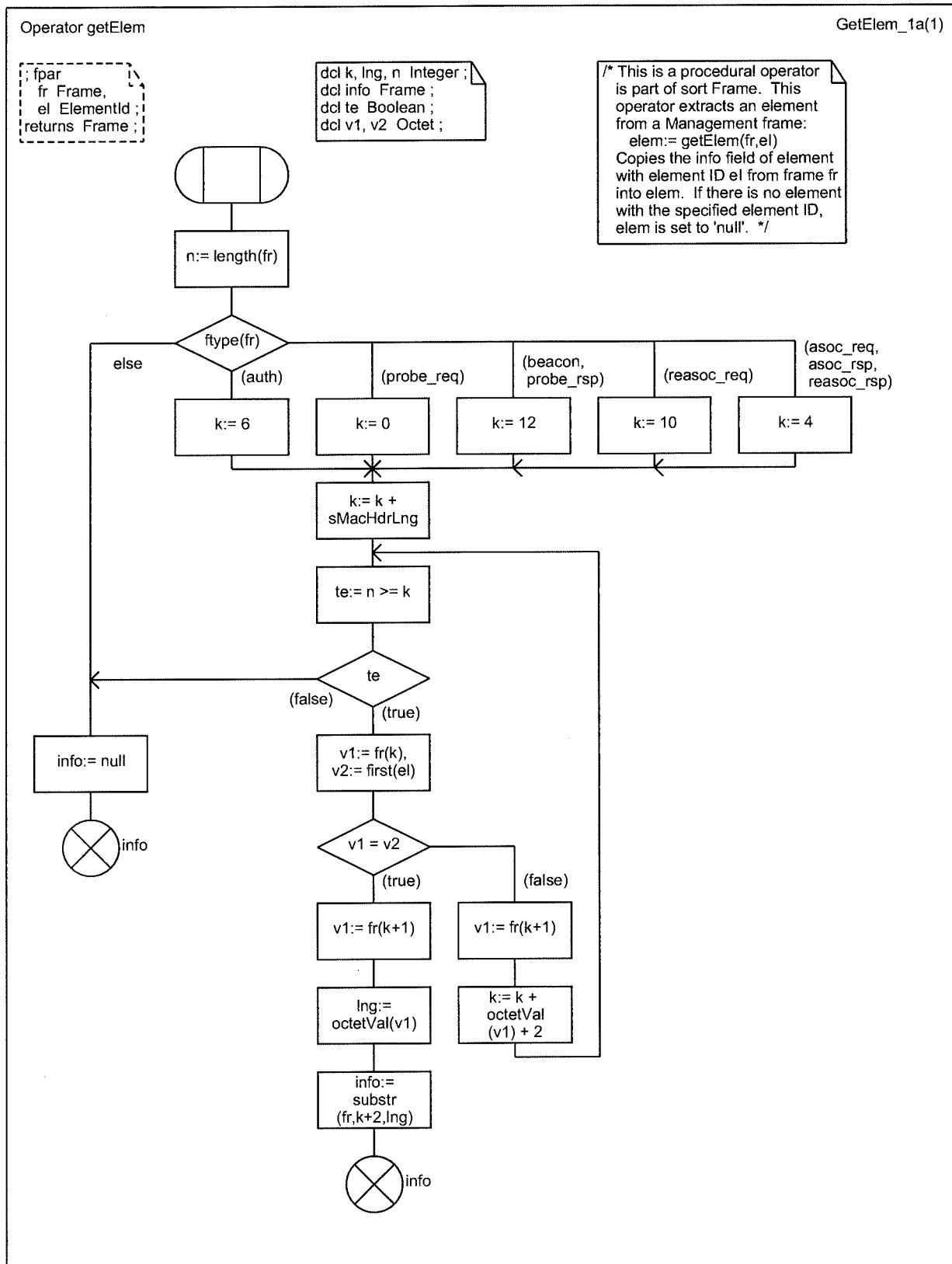
*****
* ReasonCode sort
*****
newtype ReasonCode inherits Octetstring operators all;
adding literals unspec_reason, auth_not_valid, deauth_lv_ss,
inactivity, ap_overload, class2_err, class3_err,
disas_lv_ss, asoc_not_auth;
axioms
unspec_reason == mkOS(1, 2); auth_not_valid == mkOS(2, 2);
deauth_lv_ss == mkOS(3, 2); inactivity == mkOS(4, 2);
ap_overload == mkOS(5, 2); class2_err == mkOS(6, 2);
class3_err == mkOS(7, 2); disas_lv_ss == mkOS(8, 2);
asoc_not_auth == mkOS(9, 2);
endnewtype ReasonCode;

```

```

*****
* StatusCode sort
*****
newtype StatusCode inherits Octetstring operators all;
adding literals successful, unspec_fail, unsup_cap,
reasoc_no_asoc, fail_other, unsupt_alg, auth_seq_fail,
chlng_fail, auth_timeout, ap_full, unsup_rate;
axioms
successful == mkOS(0, 2); unspec_failure == mkOS(1, 2);
unsup_cap == mkOS(10, 2); reasoc_no_asoc == mkOS(11, 2);
fail_other == mkOS(12, 2); unsupt_alg == mkOS(13, 2);
auth_seq_fail == mkOS(14, 2); chlng_fail == mkOS(15, 2);
auth_timeout == mkOS(16, 2); ap_full == mkOS(17, 2);
unsup_rate == mkOS(18, 2);
endnewtype StatusCode;

```





Package macsorts

3121_dFrameType(31)

```

*****
*   Frame Type sorts
*****
/* TypeSubtype defines the full, 6-bit frame type identifiers. */
/* These values are useful with ftype operator of Frame sort. */
newtype TypeSubtype inherits Octetstring operators all;
adding literals asoc_req, asoc_rsp, reasoc_req, reasoc_rsp,
  probe_req, probe_rsp, beacon, atim, disasoc, auth, deauth,
  ps_poll, rts, cts, ack, cfend, cfend_ack, data, data_ack,
  data_poll, data_poll_ack, null_frame, cfack, cfpoll, cfpoll_ack;
axioms
asoc_req == MkString(S8(0,0,0,0,0,0,0,0));
asoc_rsp == MkString(S8(0,0,0,0,1,0,0,0));
reasoc_req == MkString(S8(0,0,0,0,0,1,0,0));
reasoc_rsp == MkString(S8(0,0,0,0,1,1,0,0));
probe_req == MkString(S8(0,0,0,0,0,0,1,0));
probe_rsp == MkString(S8(0,0,0,0,1,0,1,0));
beacon == MkString(S8(0,0,0,0,0,0,0,1));
atim == MkString(S8(0,0,0,0,1,0,0,1));
disasoc == MkString(S8(0,0,0,0,0,1,0,1));
auth == MkString(S8(0,0,0,0,1,1,0,1));
deauth == MkString(S8(0,0,0,0,0,0,1,1));
ps_poll == MkString(S8(0,0,1,0,0,1,0,1));
rts == MkString(S8(0,0,1,0,1,1,0,1));
cts == MkString(S8(0,0,1,0,0,0,1,1));
ack == MkString(S8(0,0,1,0,1,0,1,1));
cfend == MkString(S8(0,0,1,0,0,1,1,1));
cfend_ack == MkString(S8(0,0,1,0,1,1,1,1));
data == MkString(S8(0,0,0,1,0,0,0,0));
data_ack == MkString(S8(0,0,0,1,1,0,0,0));
data_poll == MkString(S8(0,0,0,1,0,1,0,0));
data_poll_ack == MkString(S8(0,0,0,1,1,1,0,0));
null_frame == MkString(S8(0,0,0,1,0,0,1,0));
cfack == MkString(S8(0,0,0,1,1,0,1,0));
cfpoll == MkString(S8(0,0,0,1,0,1,1,0));
cfpoll_ack == MkString(S8(0,0,0,1,1,1,1,0));
endnewtype TypeSubtype;
/* BasicTypes defines the 2-bit frame type groups */
newtype BasicType inherits Bitstring operators all;
adding literals control, data, management, reserved;
axioms
control == S8(0,0,1,0,0,0,0,0); data == S8(0,0,0,1,0,0,0,0);
management == S8(0,0,0,0,0,0,0,0); reserved == S8(0,0,1,1,0,0,0,0);
endnewtype BasicType;

```



Package macsorts

3122_dMgmtFields(31)

```

*****
*   ElementID sort
*****/
newtype ElementID inherits Octetstring operators all;
adding literals eSsid, eSupRates, eFhParms, eDsParms,
eCfParms, eTim, elbParms, eCtext;
axioms
eSsid == mkOS(0, 1); /* service set identifier (0:32) */
eSupRates == mkOS(1, 1); /* supported rates (1:8) */
eFhParms == mkOS(2, 1); /* FH parameter set (5) */
eDsParms == mkOS(3, 1); /* DS parameter set (1) */
eCfParms == mkOS(4, 1); /* CF parameter set (6) */
eTim == mkOS(5, 1); /* Traffic Information Map (4:254) */
elbParms == mkOS(6, 1); /* IBSS parameter set (2) */
eCtext == mkOS(16, 1); /* challenge text (128, see 8.1.2.2) */
endnewtype ElementID;

```

```

*****
*   Capability field bit assignments sort
*****/
newtype Capability inherits Bitstring operators all;
adding literals cEss, cbss, cPollable, cPollReq, cPrivacy;
axioms
cEss == S8(1,0,0,0,0,0,0,0) // 0x00; /* ESS capability */
cbss == S8(0,1,0,0,0,0,0,0) // 0x00; /* IBSS capability */
cPollable == S8(0,0,1,0,0,0,0,0) // 0x00; /* CF-pollable (sta), PC present (ap) */
cPollReq == S8(0,0,0,1,0,0,0,0) // 0x00; /* not CF poll req (sta), PC polls (ap) */
cPrivacy == S8(0,0,0,0,1,0,0,0) // 0x00; /* WEP required */
endnewtype Capability;

```

```

*****
*   IBSS parameter set sort
*****/
newtype lbssParms inherits Octetstring operators all;
adding operators
atimWin : lbssParms -> TU;
setAtimWin : lbssParms, TU -> lbssParms;
axioms
for all ib in lbssParms( for all u in TU(
  atimWin(ib) == octetVal(ib(0)) + (octetVal(ib(1)) * 256);
  setAtimWin(ib, u) == mkOS(u mod 256, 1) // mkOS(u / 256, 1); );
endnewtype lbssParms;

```

Package macsorts

3123_d\CF_And_AsocParams(31)

```

*****
*   CF parameter set sort
*****/
newtype CfParms inherits Octetstring operators all;
adding operators
  cfpCount : CfParms -> Integer; /* CfpCount field (1) */
  setCfpCount : CfParms, Integer -> CfParms;
  cfpPeriod : CfParms -> Integer; /* CfpPeriod field (1) */
  setCfpPeriod : CfParms, Integer -> CfParms;
  cfpMaxDur : CfParms -> TU; /* CfpMaxDuration field (2) */
  setCfpMaxDur : CfParms, TU -> CfParms;
  cfpDurRem : CfParms -> TU; /* CfpDurRemaining field (2) */
  setCfpDurRem : CfParms, TU -> CfParms;
axioms for all cf in CfParms( for all i in Integer( for all u in TU(
  cfpCount(cf) == octetVal(cf(0));
  setCfpCount(cf, i) == mkOS(i, 1) // Tail(cf);
  cfpPeriod(cf) == octetVal(cf(1));
  setCfpPeriod(cf, i) == cf(0) // mkOS(i, 1) // SubStr(cf,2,4);
  cfpMaxDur(cf) == octetVal(cf(2)) + (octetVal(cf(3)) * 256);
  setCfpMaxDur(cf, u) == SubStr(cf, 0, 2) // mkOS(u mod 256, 1)
    // mkOS(u / 256, 1) // SubStr(cf, 4, 2);
  cfpDurRem(cf) == octetVal(cf(4)) + (octetVal(cf(5)) * 256);
  setCfpDurRem(cf, u) == SubStr(cf, 0, 4) // mkOS(u mod 256, 1)
    // mkOS(u / 256, 1); ));
endnewtype CfParms;

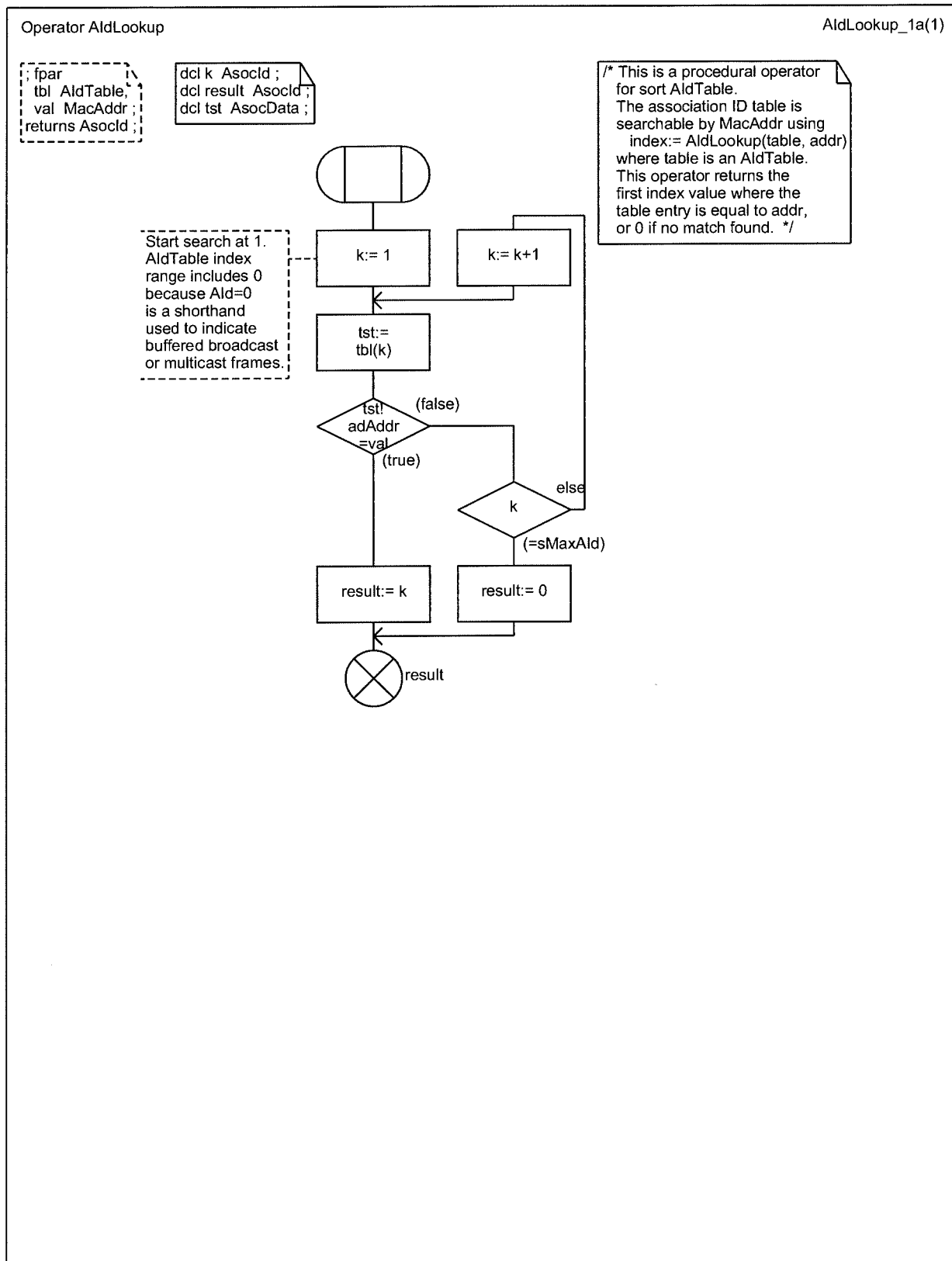
```

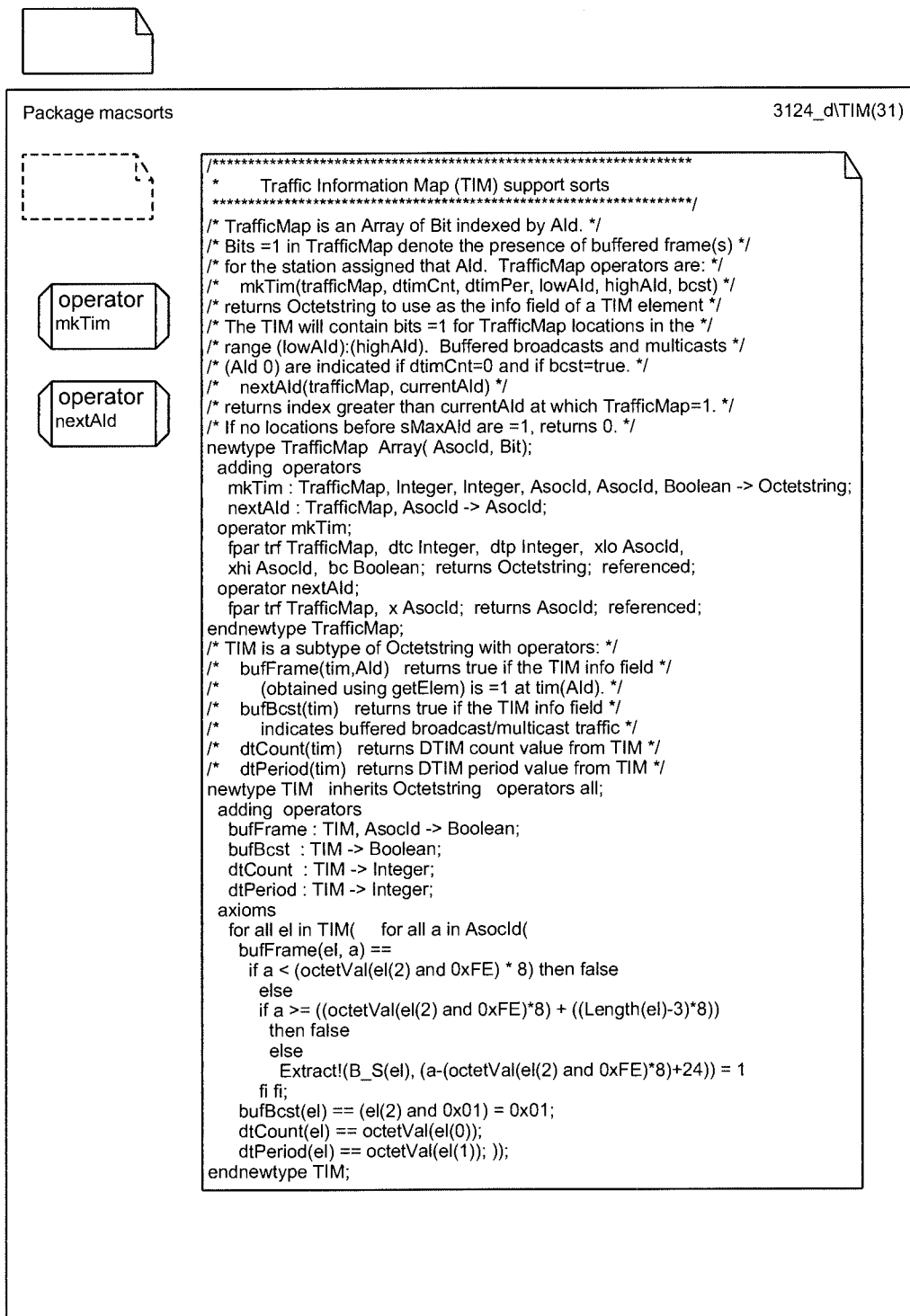
operator
AldLookup

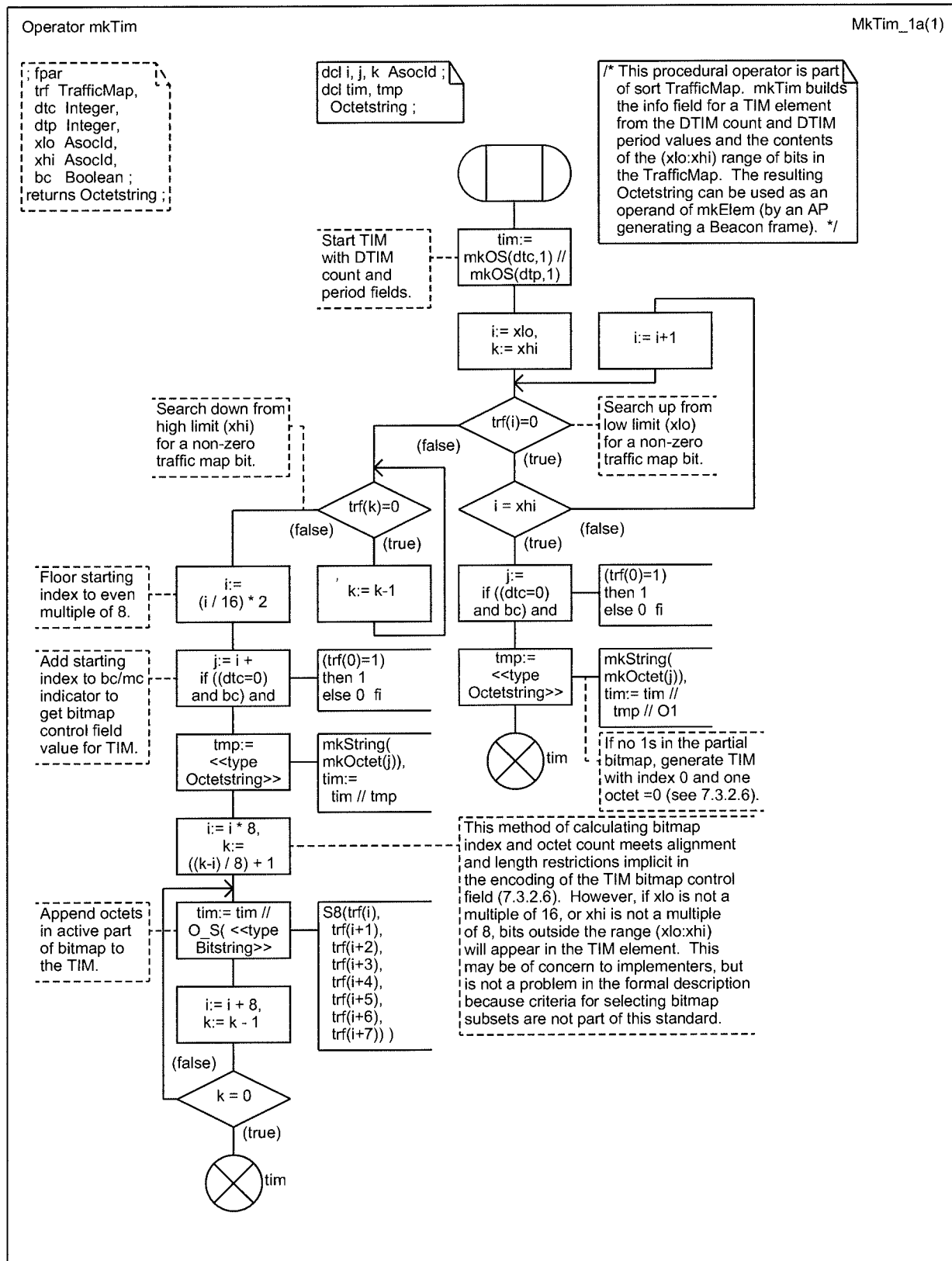
```

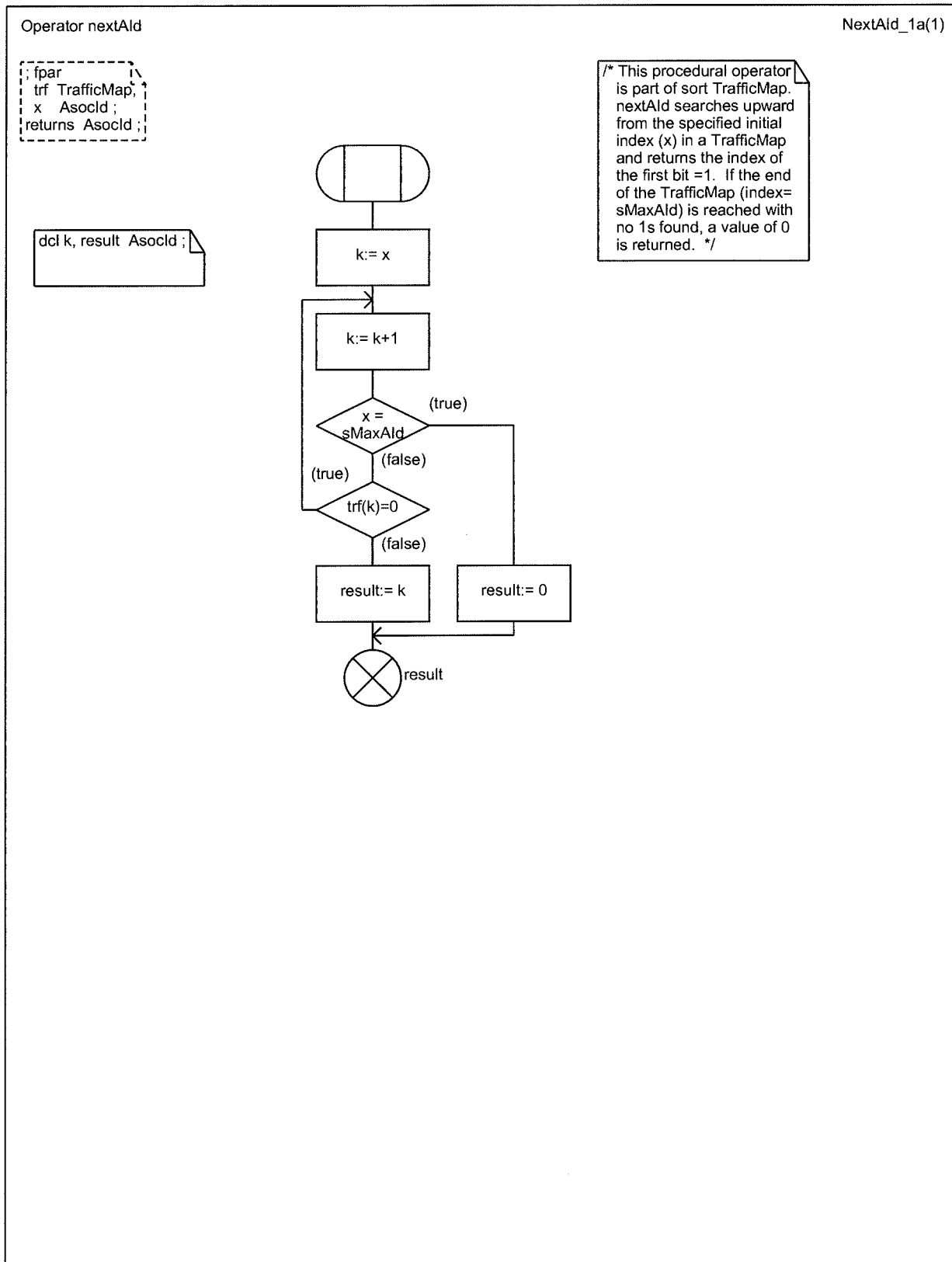
*****
*   Sorts for association management at AP
*****/
synonym sMaxAld Integer = 2007; /* 2007 is largest allowable value */
/* implementation limit may be lower */
syntype AsocId = Integer constants 0:sMaxAld endsyntype AsocId;
/* Station Association Record -- only used at APs */
newtype AsocData struct
  adAddr MacAddr; /* address of associated station */
  adPsm PwrSave; /* power save mode of the station */
  adCfPoll Boolean; /* true if station is CfPollable */
  adPollRq Boolean; /* true if station requested polling */
  adNoPoll Boolean; /* true if station requested no polling */
  adMsduIP Boolean; /* true if partial Msdu outstanding to sta */
  adAuth AuthType; /* authentication type used by station */
  adRates RateSet; /* supported rates from association request */
  adAge Time; /* time of association */
endnewtype AsocData;
/* Association table -- array of AsocData, only used at APs */
/* index:= AldLookup(table, addr) */
/* returns the index of location where table(x).adAddr=addr */
/* or 0 if no such location found. */
newtype AldTable Array(AsocId, AsocData);
adding operators
  AldLookup : AldTable, MacAddr -> AsocId;
operator AldLookup;
fpar tbl AldTable, val MacAddr; returns AsocId; referenced;
endnewtype AldTable;

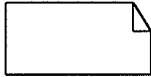
```









Package macsorts

3125_dRateAndDurationSorts(31)



```

*****
* Multi-rate support sorts
*****/
newtype Rate inherits Octet operators all;
adding operators
  calcDur : Rate, Integer -> Integer; /* converts (rate,bitCount) to integer usec */
  rateVal : Rate -> Rate; /* clears high-order bit */
  basicRate : Rate -> Rate; /* sets high-order bit */
  isBasic : Rate -> Boolean; /* true if high-order bit set */
axioms
  for all r in Rate( for all i in Integer( for all b in Boolean(
    calcDur(r, i) == (((10000000 + (octetVal(r and 0x7F) - 1)) /
      (500 * octetVal(r and 0x7F))) * i) + 9999) / 10000;
    rateVal(r) == r and 0x7F; basicRate(r) == r or 0x80;
    isBasic(r) == (r and 0x80) = 0x80; ));
endnewtype Rate;
syntype RateString = Octetstring endsyntype RateString;

```

```

*****
* MPDU duration factor support sort
*****/
/* These operators support the encoding used to allow */
/* an Integer to represent the value of aMpduDurationFactor. */
/* calcDF(PlcpBits, MpduBits) returns an Integer which is */
/* the fractional part of ((PlcpBits/MpduBits)-1)*(1e9). */
/* stuff(durFactor, MpduBits) returns the number of PlcpBits */
/* which result from MpduBits at the specified durFactor. */
newtype DurFactor inherits Integer operators all;
adding operators
  calcDF : Integer, Integer -> DurFactor;
  stuff : DurFactor, Integer -> Integer;
axioms
  for all df in DurFactor( for all mb, pb in Integer(
    calcDF(pb, mb) == ((pb * 1000000000) / mb) - 1000000000;
    stuff(df, mb) == ((mb * df) + (mb - 1)) / 1000000000; );
endnewtype DurFactor;

```



Package macsorts

3126_d\FH_DS_Params(31)

```

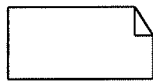
/*****
*   FH parameter set sort
*****/
newtype FhParms inherits Octetstring operators all;
adding operators
  dwellTime : FhParms -> TU; /* Dwell Time field (2) */
  setDwellTime : FhParms, TU -> FhParms;
  hopSet : FhParms -> Integer; /* Hop Set field (1) */
  setHopSet : FhParms, Integer -> FhParms;
  hopPattern : FhParms -> Integer; /* Hop Pattern field (1) */
  setHopPattern : FhParms, Integer -> FhParms;
  hopIndex : FhParms -> Integer; /* Hop Index field (1) */
  setHopIndex : FhParms, Integer -> FhParms;
axioms
  for all fh in FhParms( for all i in Integer( for all u in TU(
    dwellTime(fh) == octetVal(fh(0)) + (octetVal(fh(1)) * 256);
    setDwellTime(fh, u) == mkOS(u mod 256, 1) // mkOS(u / 256, 1) // SubStr(fh, 2, 3);
    hopSet(fh) == octetVal(fh(2));
    setHopSet(fh, i) == SubStr(fh, 0, 2) // mkOS(i, 1) // SubStr(fh, 3, 2);
    hopPattern(fh) == octetVal(fh(3));
    setHopPattern(fh, i) == SubStr(fh, 0, 3) // mkOS(i, 1) // Last(fh);
    hopIndex(fh) == octetVal(fh(4));
    setHopIndex(fh, i) == SubStr(fh, 0, 4) // mkOS(i, 1);));
endnewtype FhParms;

```

```

/*****
*   DS parameter set sort
*****/
newtype DsParms inherits Octetstring operators all;
adding operators
  curChannel : DsParms -> Integer; /* Current Channel (1) */
  setCurChannel : DsParms, Integer -> DsParms;
axioms
  for all ds in DsParms( for all i in Integer(
    curChannel(ds) == octetVal(ds(0));
    setCurChannel(ds, i) == mkOS(i);));
endnewtype DsParms;

```



```

Package macsorts 3127_d\PHY_Params(31)

/*
 * Generic PHY parameter set sort
 */
/* Generic PHY parameter element for signals related to Beacons */
/* and Probe Responses that are PHY-type independent. */
syntype PhyParms = Octetstring endsyntype PhyParms;

NEWTYPED Phychrstcs struct
aSlotTime Usec;
aSifsTime Usec;
aCCATime Usec;
aRxTxTurnaroundTime Usec;
aTxPLCPDelay Usec;
aRxPLCPDelay Usec;
aRxTxSwitchTime Usec;
aTxRampOnTime Usec;
aTxRampOffTime Usec;
aTxRFDelay Usec;
aRxRFDelay Usec;
aAirPropagationTime Usec;
aMACProcessingDelay Usec;
aPreambleLength Usec;
aPLCPHeaderLength Usec;
aMPDUDurationFactor DurFactor;
aMPDUMaxLength Integer;
aCWmin Integer;
aCWmax Integer;
EndNewType Phychrstcs;
    
```

```
use macsorts;
```

Package macmib

3201_d\StationConfig(5)

```
/* This Package contains definitions of the MAC MIB attributes
and the subset of the PHY MIB attributes used by the MAC state
machines. These are needed under Z.100 to permit analysis of
the state machine definitions. In future revisions these may
be replaced with the ASN.1 MIB definition which appears as
as Annex D, for use with a Z.105-compliant SDL tool is available. */
```

```

/*****
 * StationConfig Table
 *****/
remote dot11MediumOccupancyLimit TU nodelay;
synonym dot11CfPollable Boolean = <<package macsorts>> sCfPollable;
remote dot11CfPeriod Integer nodelay;
remote dot11CfMaxDuration TU nodelay;
remote dot11AuthenticationResponseTimeout TU nodelay;
synonym dot11PrivacyOptionImplemented Boolean=true;
remote dot11PowerMangementMode PwrSave nodelay;
remote dot11DesiredSsid OctetString nodelay;
remote dot11DesiredBssType BssType nodelay;
remote dot11OperationalRateSet RateString nodelay;
remote dot11BeaconPeriod TU nodelay;
remote dot11DtimPeriod Integer nodelay;
remote dot11AssociationResponseTimeout TU nodelay;

remote dot11WepUndecryptableCount Counter32 nodelay;
remote dot11ReceiveDTIMs Boolean nodelay;
remote dot11AuthenticationType AuthTypeSet nodelay;

```

```

/*****
 * AuthenticationAlgorithms Table
 *****/
synonym dot11AuthenticationAlgorithms AuthTypeSet =
incl(open_system, incl(shared_key));
/* NOTE: The members of this set are the
dot11AuthenticationAlgorithm values of all
dot11 AuthenticationAlgorithmsEntry instances
for which dot11AuthenticationAlgorithmsEnable=True.
Do not include shared_key in this set
unless dot11PrivacyOptionImplemented=true. */

```

```

/*****
 * WepDefaultKeys Table
 * (if dot11PrivacyOptionImplemented=true)
 *****/
remote dot11WepDefaultKeys KeyVector nodelay;

```

```

/*****
 * WepKeyMappings Table
 * (if dot11PrivacyOptionImplemented=true)
 *****/
remote dot11WepKeyMappings KeyMapArray nodelay;

```



```
use macsorts ;
```

```
Package macmib
```

```
3202_d\PrivOperation(5)
```

```

/*****
 * Privacy Table
 * (only if dot11PrivacyOptionImplemented=true)
 *****/
remote dot11PrivacyInvoked Boolean nodelay;
remote dot11WepDefaultKeyId KeyIndex nodelay;
synonym dot11WepKeyMappingLength Integer =
  <<package macsorts>> sWepKeyMappingLength;
remote dot11ExcludeUnencrypted Boolean nodelay;
remote dot11WepIcvErrorCount Counter32 nodelay;
remote dot11WepExcludedCount Counter32 nodelay;

```

```

/*****
 * Operation Table
 *****/
synonym dot11MacAddress MacAddr =
  <<type MacAddr>> S6(0x00, 0x11, 0x22, 0x33, 0x44, 0x55);
  /* each station has a unique globally administered address */
  /* Value may be overwritten with locally administered address at */
  /* MlmeReset, but is always a static value during MAC operation */
remote dot11RtsThreshold Integer nodelay;
remote dot11ShortRetryLimit Integer nodelay;
remote dot11LongRetryLimit Integer nodelay;
remote dot11FragmentationThreshold Integer nodelay;
remote dot11MaxTransmitMsduLifetime TU nodelay;
remote dot11MaxReceiveLifetime TU nodelay;
synonym dot11ManufacturerId Charstring = 'name of manufacturer';
synonym dot11ProductId Charstring = 'identifier unique to manufacturer';

```

```

/*****
 * GroupAddresses Table
 *****/
remote dot11GroupAddresses MacAddrSet nodelay;

```

use macsorts ;

Package macmib

3203_d\Counters(5)



```

/*****
 *   Counters Table
 *****/
remote dot11TransmittedFragmentCount Counter32 nodelay;
remote dot11MulticastTransmittedFrameCount Counter32 nodelay;
remote dot11FailedCount Counter32 nodelay;
remote dot11RetryCount Counter32 nodelay;
remote dot11MultipleRetryCount Counter32 nodelay;
remote dot11RtsSuccessCount Counter32 nodelay;
remote dot11RtsFailureCount Counter32 nodelay;
remote dot11AckFailureCount Counter32 nodelay;
remote dot11ReceivedFragmentCount Counter32 nodelay;
remote dot11MulticastReceivedFrameCount Counter32 nodelay;
remote dot11FcsErrorCount Counter32 nodelay;
remote dot11FrameDuplicateCount Counter32 nodelay;

```

use macsorts ;

Package macmib

3204_dPhyOperation(5)

```

/*****
 *   PhyOperation Table
 *   (values shown are mostly for FH PHY)
 *****/
synonym FHphy Integer = 01; /* enumerated dot11PHYType value */
synonym DSphy Integer = 02; /* enumerated dot11PHYType value */
synonym IRPhy Integer = 03; /* enumerated dot11PHYType value */
synonym dot11PHYType Integer = FHphy;
remote dot11CurrentRegDomain Integer nodelay;
synonym dot11TempType Integer = 01;

/*****
 *   PhyCharacteristic Parameters (values shown are mostly for FH PHY )
 *****/
/* NOTE: The PhyCharacteristics are defined as synonyms because
 their values are static during MAC operation. It is assumed
 that , during each initialization of MAC operation, current
 values for each of these parameters are obtained from the
 PHY using the PlmeCharacteristics primitive. */
synonym aSlotTime Usec = (aCcaTime + aRxTxTurnaroundTime +
 aAirPropagationTime + aMacProcessingTime);
synonym aCcaTime Usec = 27;
synonym aRxTxTurnaroundTime Usec = (aTxPlcpDelay + aRxTxSwitchTime +
 aTxRampOnTime + aTxRfDelay);
synonym aTxPlcpDelay Usec = 1;
synonym aRxTxSwitchTime Usec = 10;
synonym aTxRampOnTime Usec = 8;
synonym aTxRfDelay Usec = 1;
synonym aSifsTime Usec = (aRxRfDelay + aRxPlcpDelay +
 aMacProcessingTime + aRxTxTurnaroundTime);
synonym aRxRfDelay Usec = 4;
synonym aRxPlcpDelay Usec = 2;
synonym aMacProcessingTime Usec = 2;
synonym aTxRampOffTime Usec = 8;
synonym aPreambleLength Usec = 96;
synonym aPlcpHeaderLength Usec = 32;
synonym aMpduDurationFactor <<package macsorts>> DurFactor = 31250000;
synonym aMpduMaxLength Integer = 4095;
synonym aAirPropagationTime Usec = 1;
synonym aCWmax Integer = 1023;
synonym aCWmin Integer = 15;

```

use macsorts ;

Package macmib

3205_dPhyRateFhss(5)



```

*****
* SupportedDataRatesTx Table (values shown are for FH PHY)
*****/
synonym aSupportedRatesTx Octetstring = S8(0x82, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00);

*****
* SupportedDataRatesRx Table (values shown are for FH PHY)
*****/
synonym aSupportedRatesRx Octetstring = S8(0x82, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00);

synonym aPrefMaxMpduFragmentLength Integer = aMpduMaxLength;
    
```

```

*****
* PhyFHSS Table
* (only used with FH PHY)
*****/
synonym dot11HopTime Usec = 224;
remote dot11CurrentChannelNumber Integer nodelay;
synonym dot11MaxDwellTime TU = 390;
remote dot11CurrentSet Integer nodelay;
remote dot11CurrentPattern Integer nodelay;
remote dot11CurrentIndex Integer nodelay;
    
```

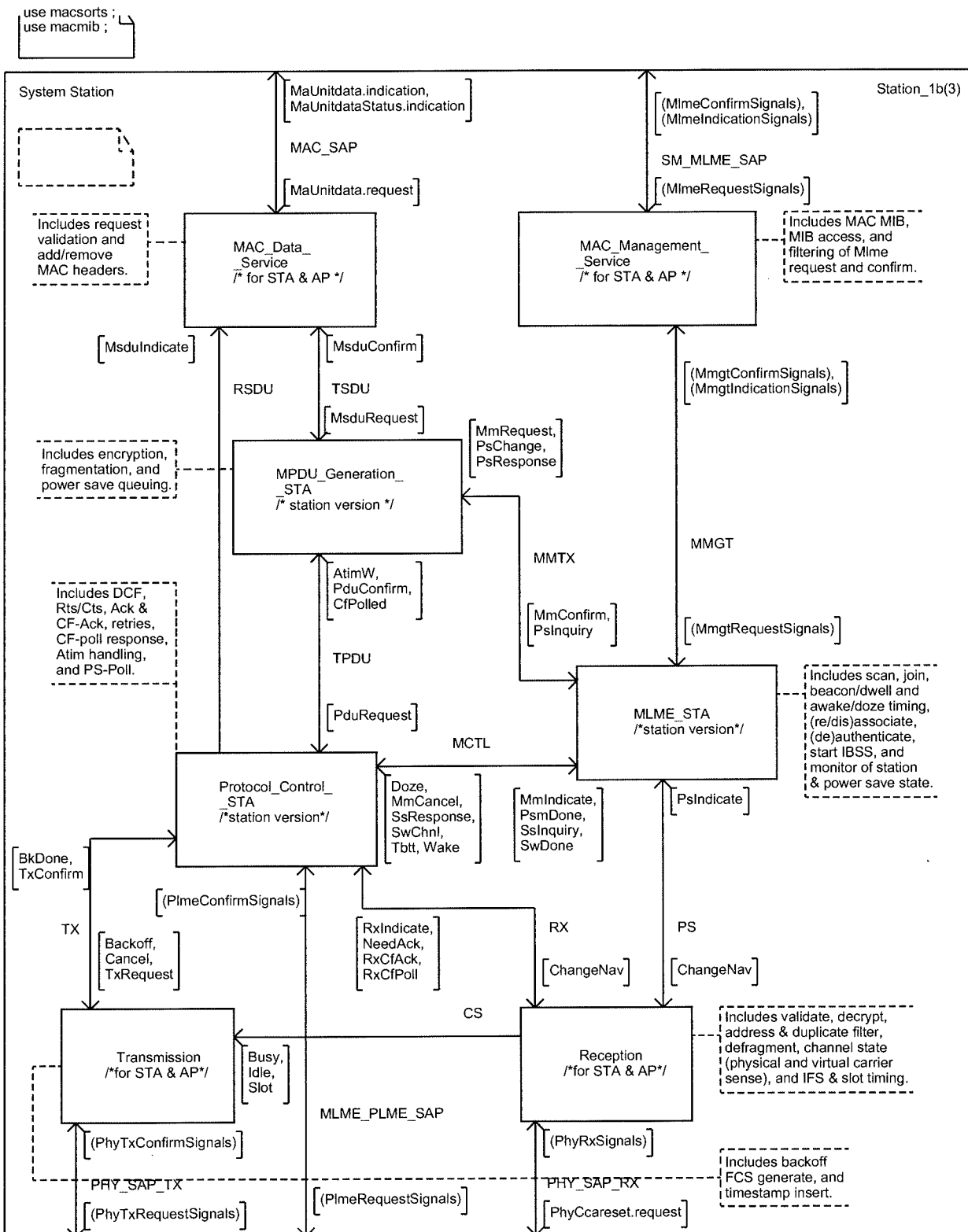
```

*****
/* The MAC state machines currently do not reference any attributes in:
   PhyAntenna Table, PhyTxPower Table, PhyDsss Table, PhyIrr Table,
   RegDomainsSupported Table, AntennasList Table. */
/*endpackage;*/
*****/
    
```

C.3 State machines for MAC stations

The following SDL-92 system specification defines operation of the MAC protocol at an IEEE 802.11 STA. Many aspects of STA operation also apply to AP operation. These are defined in blocks and processes referenced from both the STA and AP system specifications. Blocks and processes used in both STA and AP are identifiable by the SDL comment `/* for STA & AP */` below the block or process name. Blocks and processes specific to STA operation are identifiable by the SDL comment `/* station version */` below the block or process name. The definitions of all blocks and processes referenced in the station system specification appear in Clause C.3.

The remainder of Clause C.3 is the formal description, in SDL/GR, of an IEEE 802.11 STA.



```
use macsorts ;
use macmib ;
```

System Station

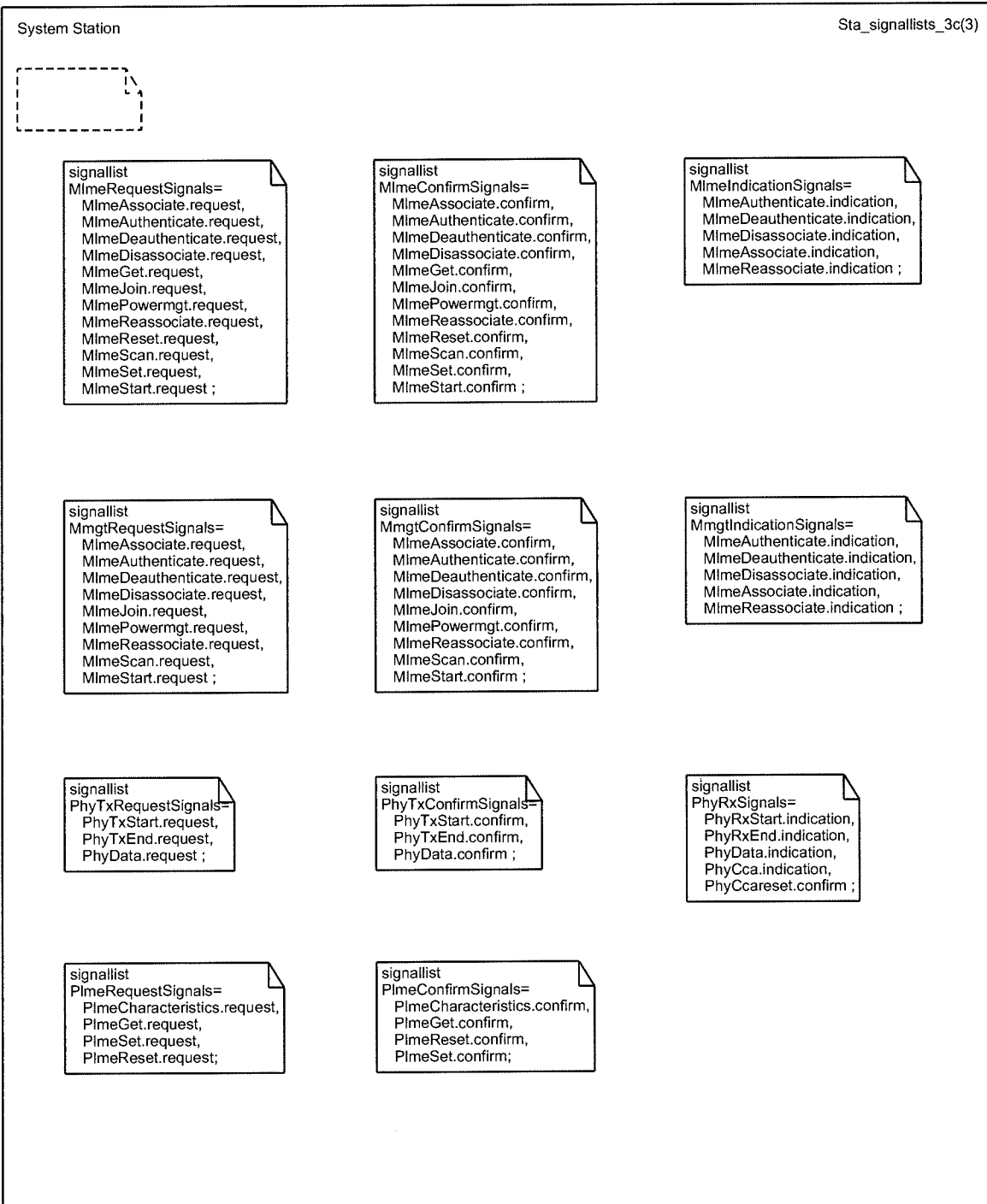
Sta_signals_2d(3)

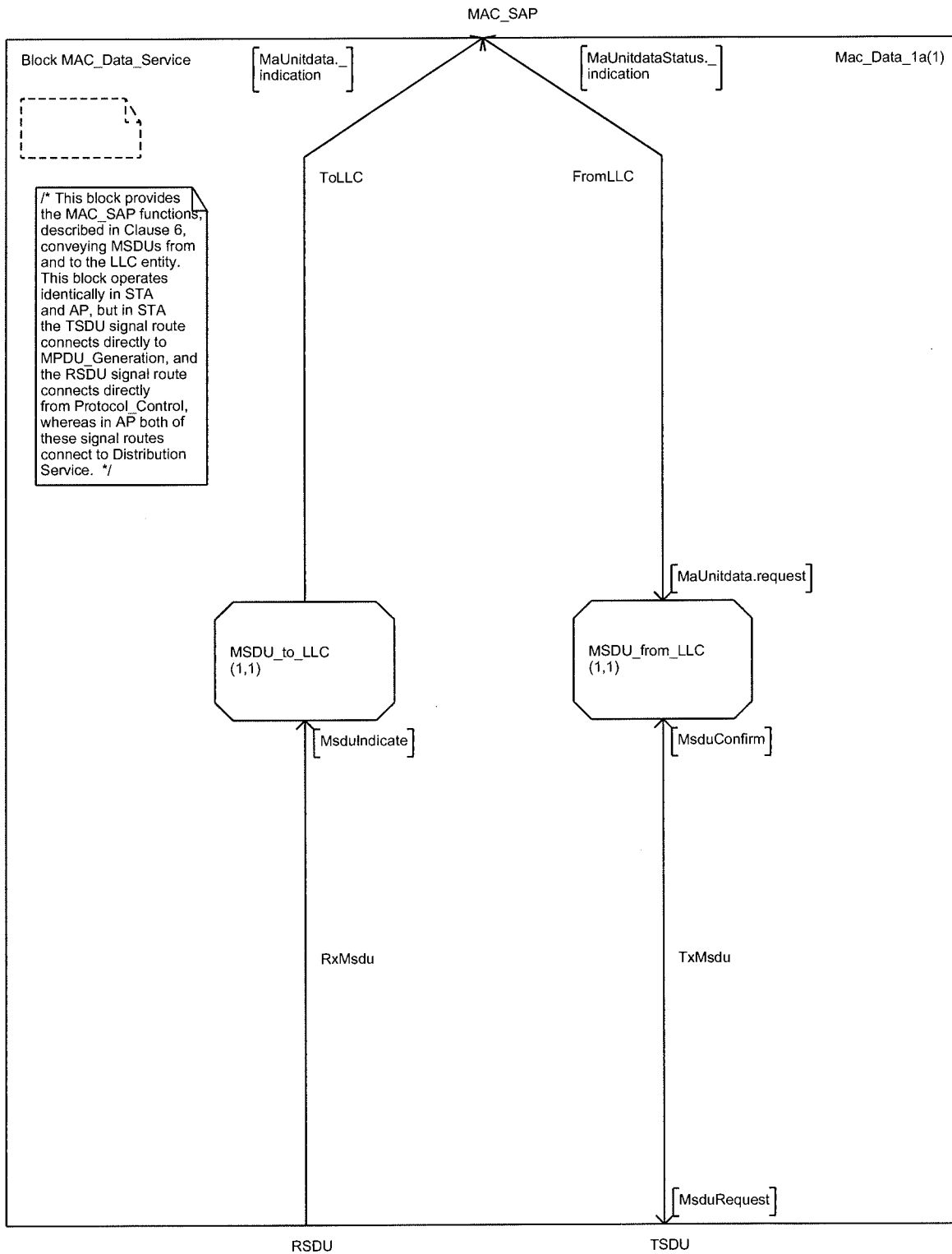


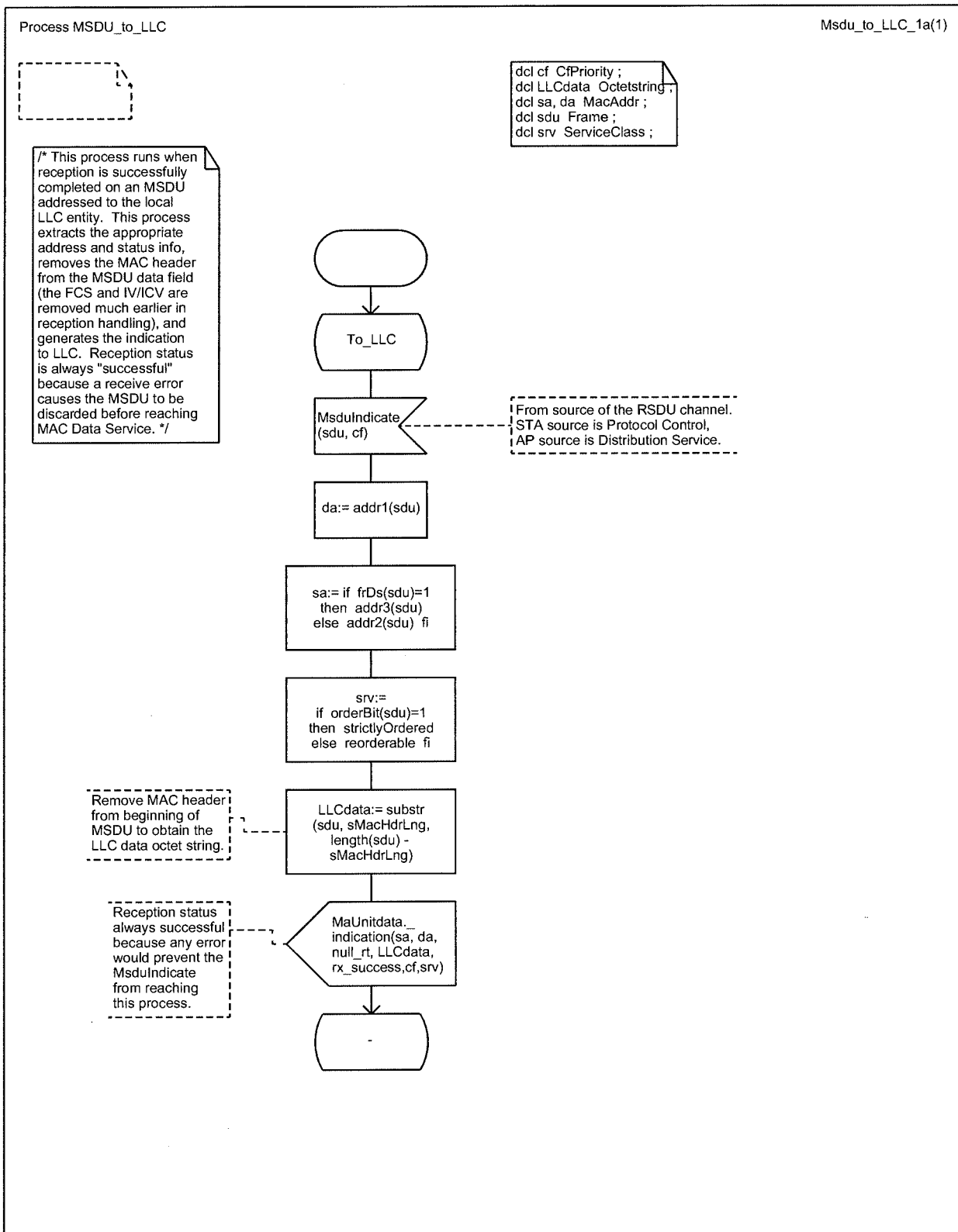
```
signal
AtimW,
Backoff(Integer,Integer),
BkDone(Integer),
Busy,
Cancel,
CfPolled,
ChangeNav(Time,Duration,NavSrc),
Doze,
Idle,
MaUnitdata.indication(MacAddr,MacAddr,
Routing,Octetstring,RxStatus,
CfPriority,ServiceClass),
MaUnitdata.request(MacAddr,MacAddr,
Routing,Octetstring,CfPriority,ServiceClass),
MaUnitdataStatus.indication(MacAddr,
MacAddr,TxStatus,CfPriority,ServiceClass),
MlmeAssociate.confirm(MlmeStatus),
MlmeAssociate.indication(MacAddr),
MlmeAssociate.request(MacAddr,Kusec,Capability,Integer),
MlmeAuthenticate.confirm
(MacAddr,AuthType,MlmeStatus),
MlmeAuthenticate.indication(MacAddr,AuthType),
MlmeAuthenticate.request(MacAddr,AuthType,Kusec),
MlmeDeauthenticate.confirm(MacAddr,MlmeStatus),
MlmeDeauthenticate.indication(MacAddr,ReasonCode),
MlmeDeauthenticate.request(MacAddr,ReasonCode),
MlmeDisassociate.confirm(MlmeStatus),
MlmeDisassociate.indication(MacAddr,ReasonCode),
MlmeDisassociate.request(MacAddr,ReasonCode),
MlmeGet.confirm(MibStatus,MibAtrib,MibValue),
MlmeGet.request(MibAtrib),
MlmeJoin.confirm(MlmeStatus),
MlmeJoin.request(BssDscr,Integer,Usec,Ratestring),
MlmePowermgt.confirm(MlmeStatus),
MlmePowermgt.request(PwrSave,Boolean,Boolean),
MlmeReassociate.confirm(MlmeStatus),
MlmeReassociate.indication(MacAddr),
MlmeReassociate.request(MacAddr,Kusec,Capability,Integer),
MlmeReset.confirm(MlmeStatus),
MlmeReset.request(MacAddr,Boolean),
MlmeScan.confirm(BssDscrSet,MlmeStatus),
MlmeScan.request(BssTypeSet,MacAddr,Octetstring,
ScanType,Usec,Intstring,Kusec,Kusec),
MlmeSet.confirm(MibStatus,MibAtrib),
MlmeSet.request(MibAtrib,MibValue),
MlmeStart.confirm(MlmeStatus),
MlmeStart.request(Octetstring,BssType,Kusec,
Integer,CfParms,PhyParms,IbssParms,Usec,
Capability,Ratestring,Ratestring) ;
```

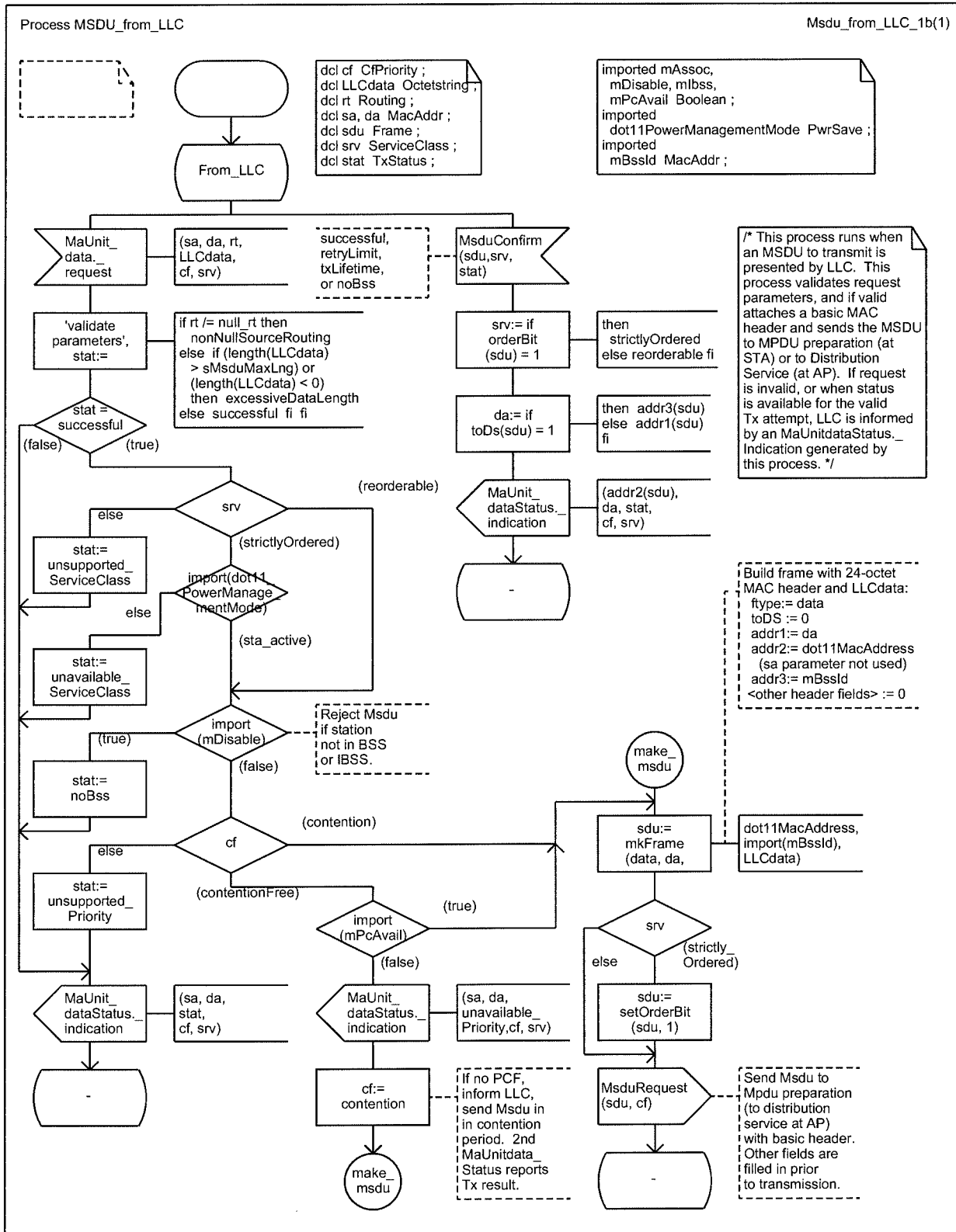
```
signal
MmCancel,
MmConfirm(Frame,TxStatus),
MmIndicate(Frame,Time,Time,StateErr),
MmRequest(Frame,Imed,Rate),
MsduConfirm(Frame,CfPriority,TxStatus),
MsduIndicate(Frame,CfPriority),
MsduRequest(Frame,CfPriority),
NeedAck(MacAddr,Time,Duration,Rate),
PduConfirm(FragSdu,TxResult),
PduRequest(FragSdu),
PhyCca.indication(Ccastatus),
PhyCcarst.confirm,
PhyCcarst.request,
PhyData.confirm,
PhyData.indication(Octet),
PhyData.request(Octet),
PhyRxEnd.indication(PhyRxStat),
PhyRxStart.indication(Integer,Rate),
PhyTxEnd.confirm,
PhyTxEnd.request,
PhyTxStart.confirm,
PhyTxStart.request(Integer,Rate),
PlmeCharacteristics.confirm(PhyChrctcs),
PlmeCharacteristics.request,
PlmeGet.confirm(MibStatus,
MibAtrib,MibValue),
PlmeGet.request(MibAtrib),
PlmeReset.confirm(Boolean),
PlmeReset.request,
PlmeSet.confirm(MibStatus,MibAtrib),
PlmeSet.request(MibAtrib,MibValue),
PsmDone,
PsChange(MacAddr,PsMode),
PsIndicate(MacAddr,PsMode),
PsInquiry(MacAddr),
PsResponse(MacAddr,PsMode),
ResetMAC,
RxCfAck(MacAddr),
RxIndicate(Frame,Time,Time,Rate),
Slot,
SsInquiry(MacAddr),
SsResponse(MacAddr,
StationState,StationState),
SwChnl(Integer,Boolean),
SwDone,
TBTT,
TxConfirm,
TxRequest(Frame,Rate),
Wake ;
```

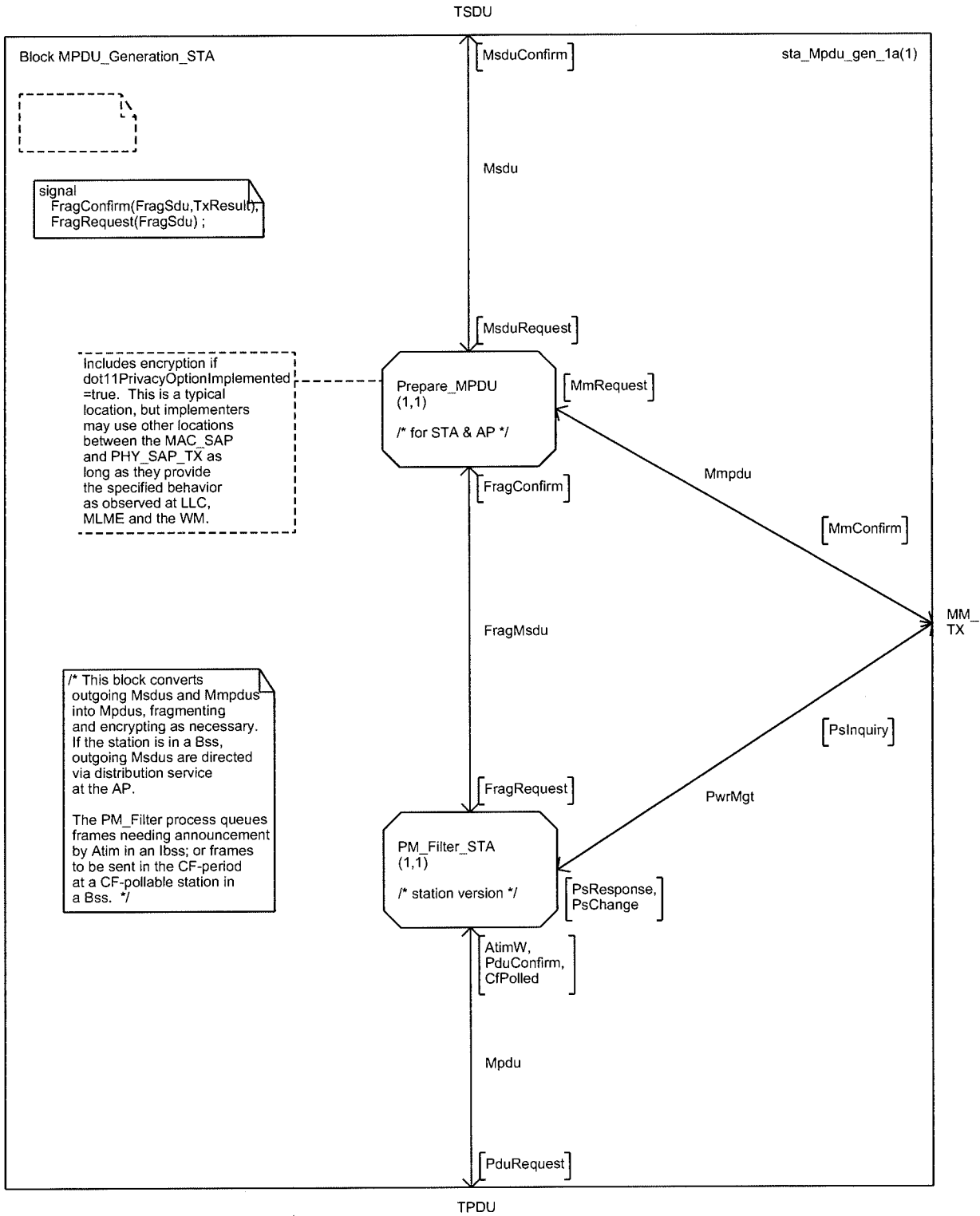
```
use macsorts ;
use macmib ;
```

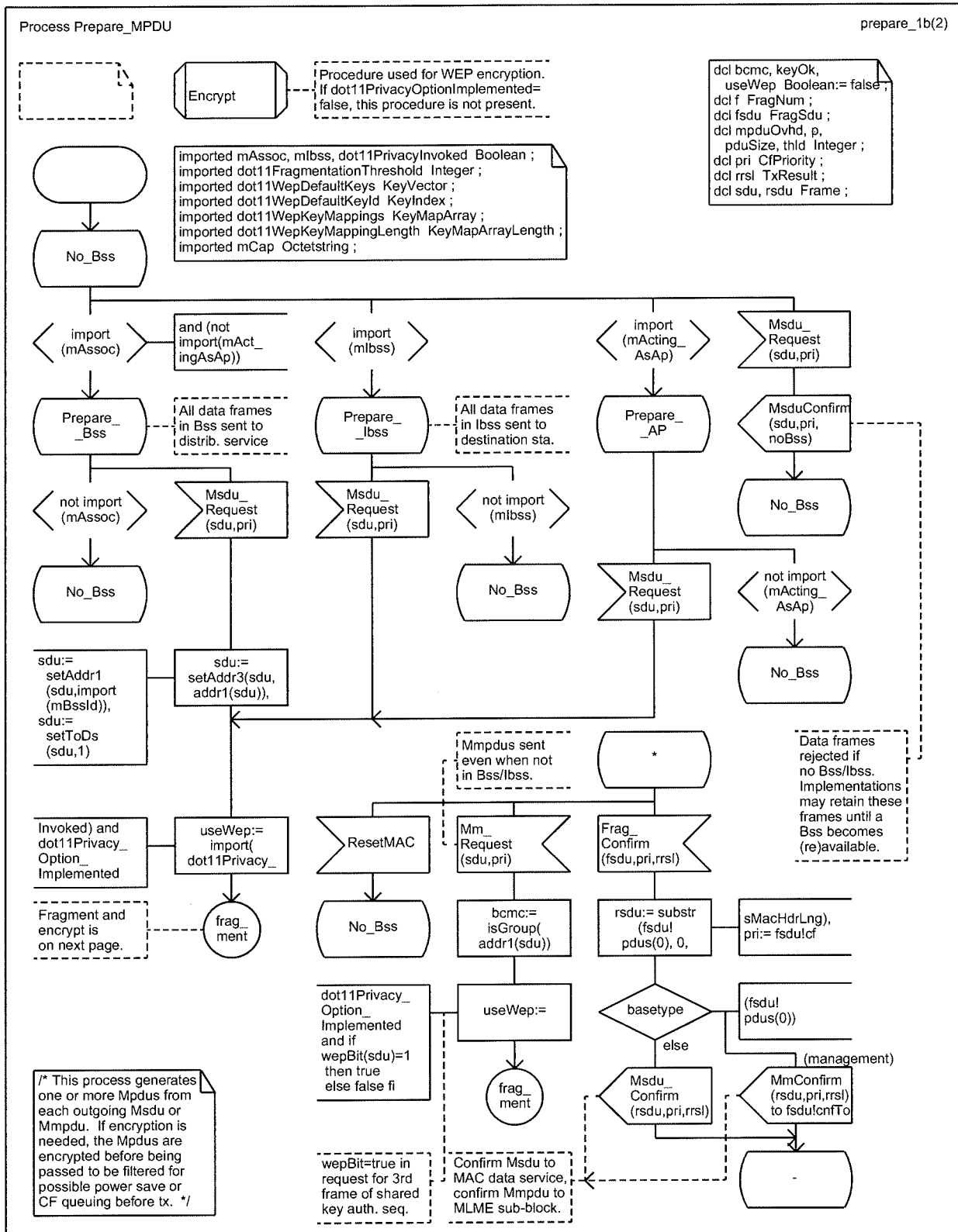


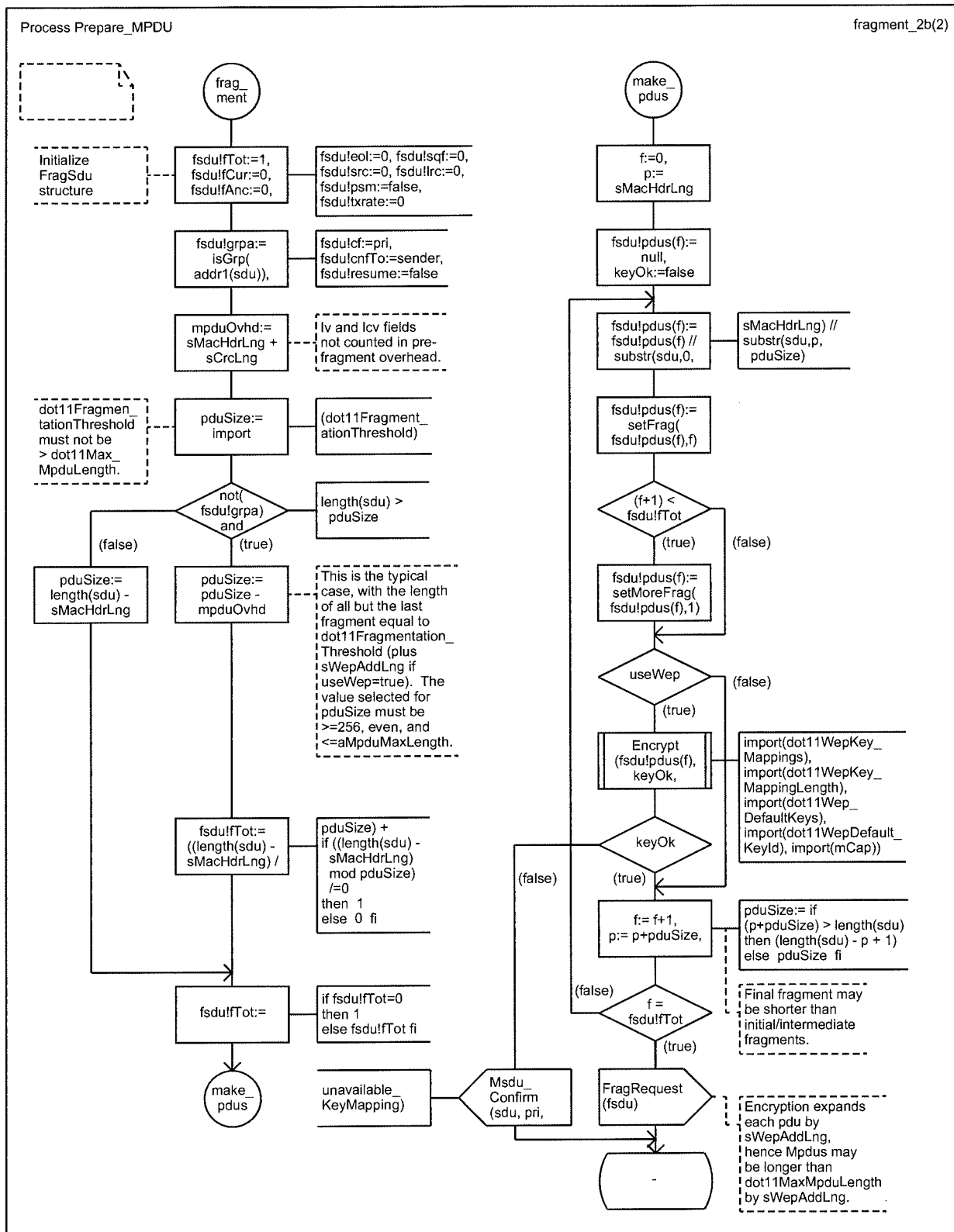


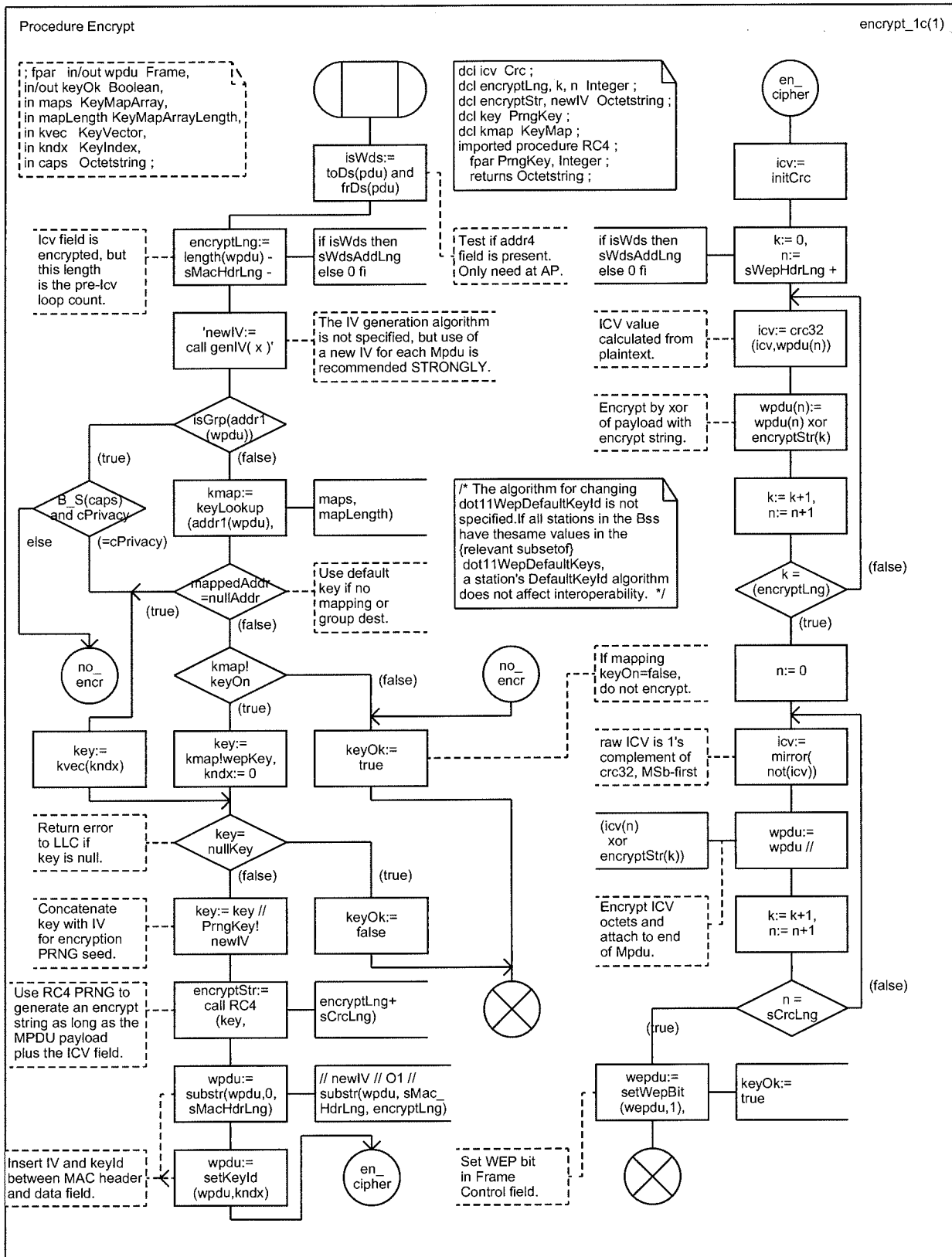


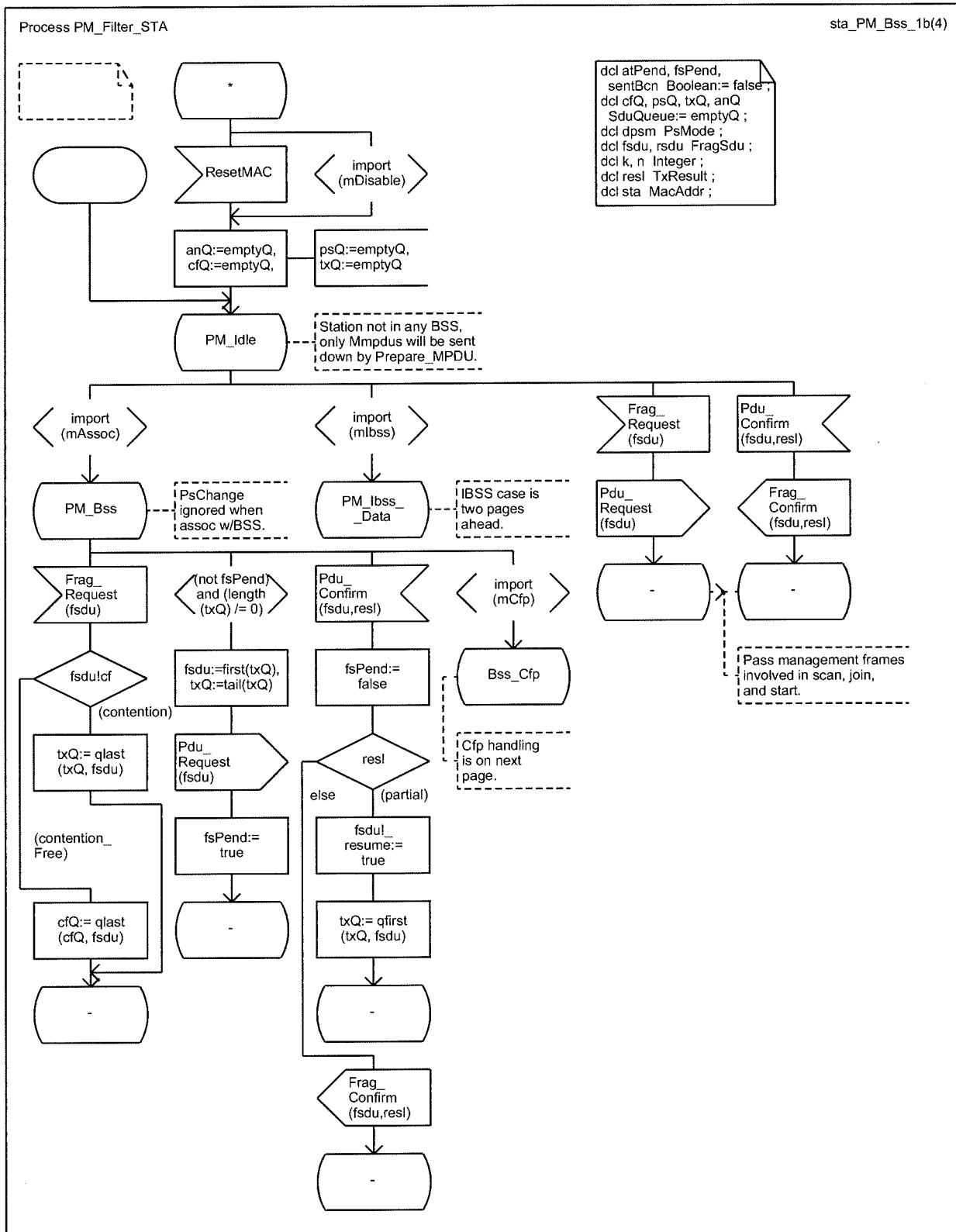


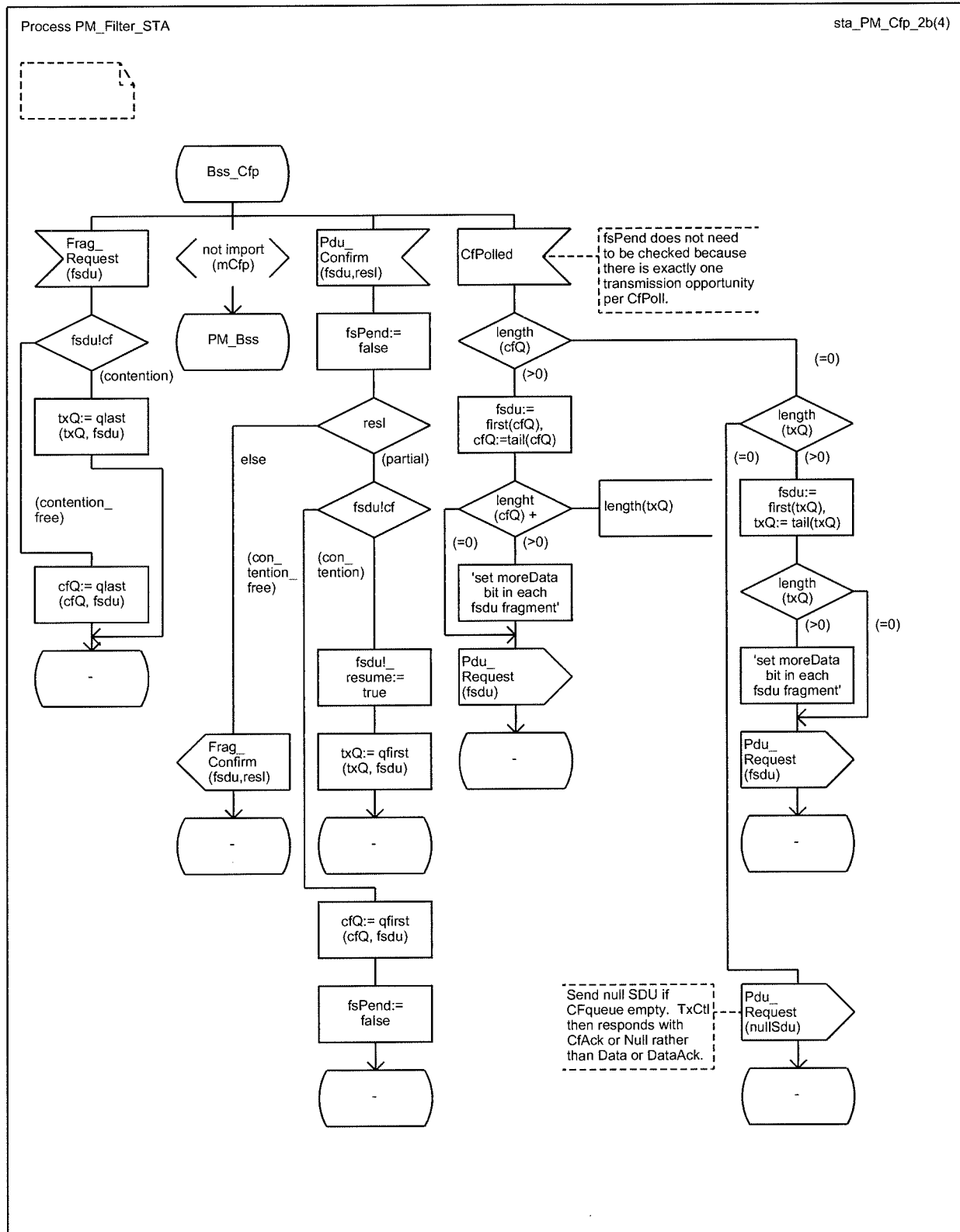


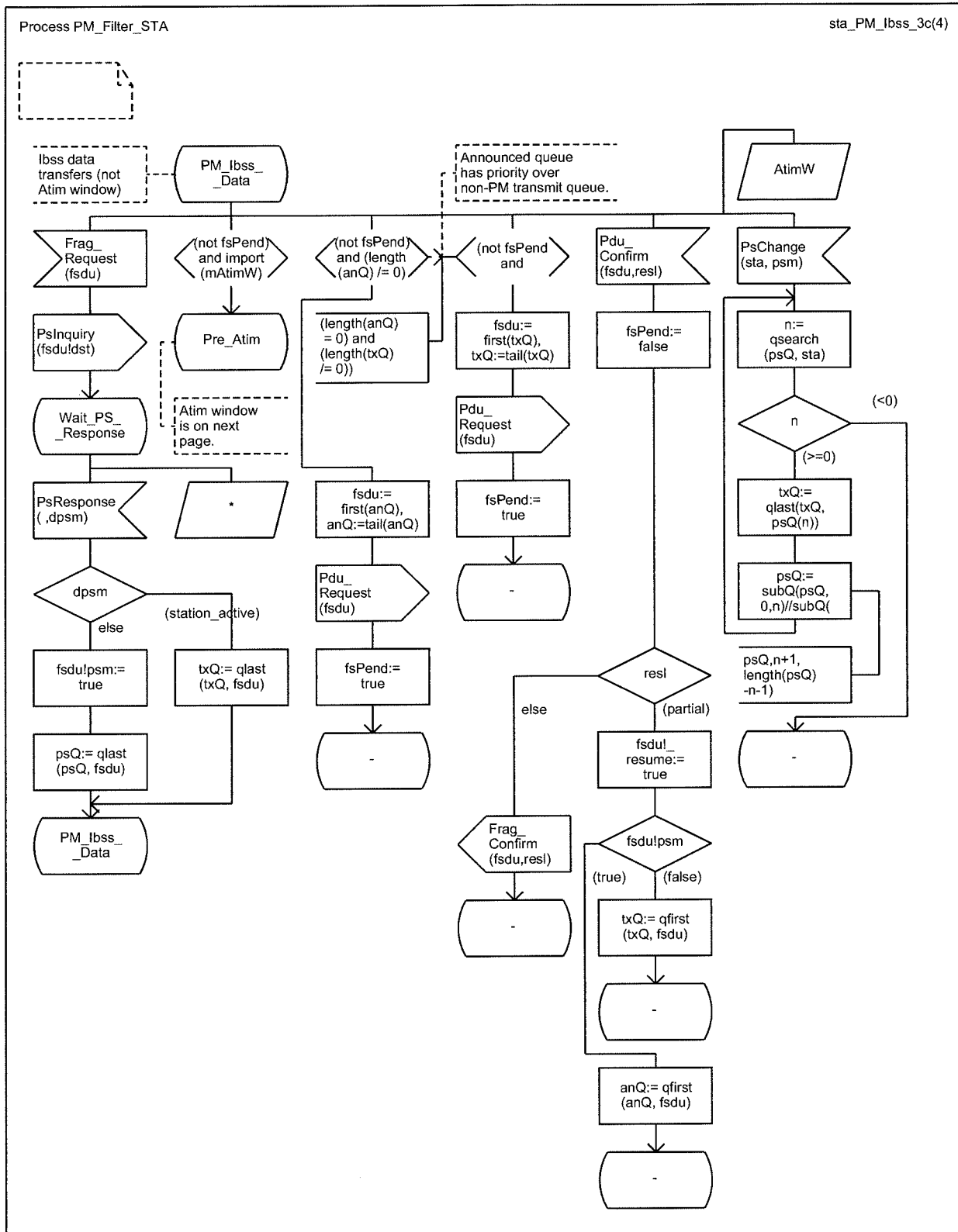


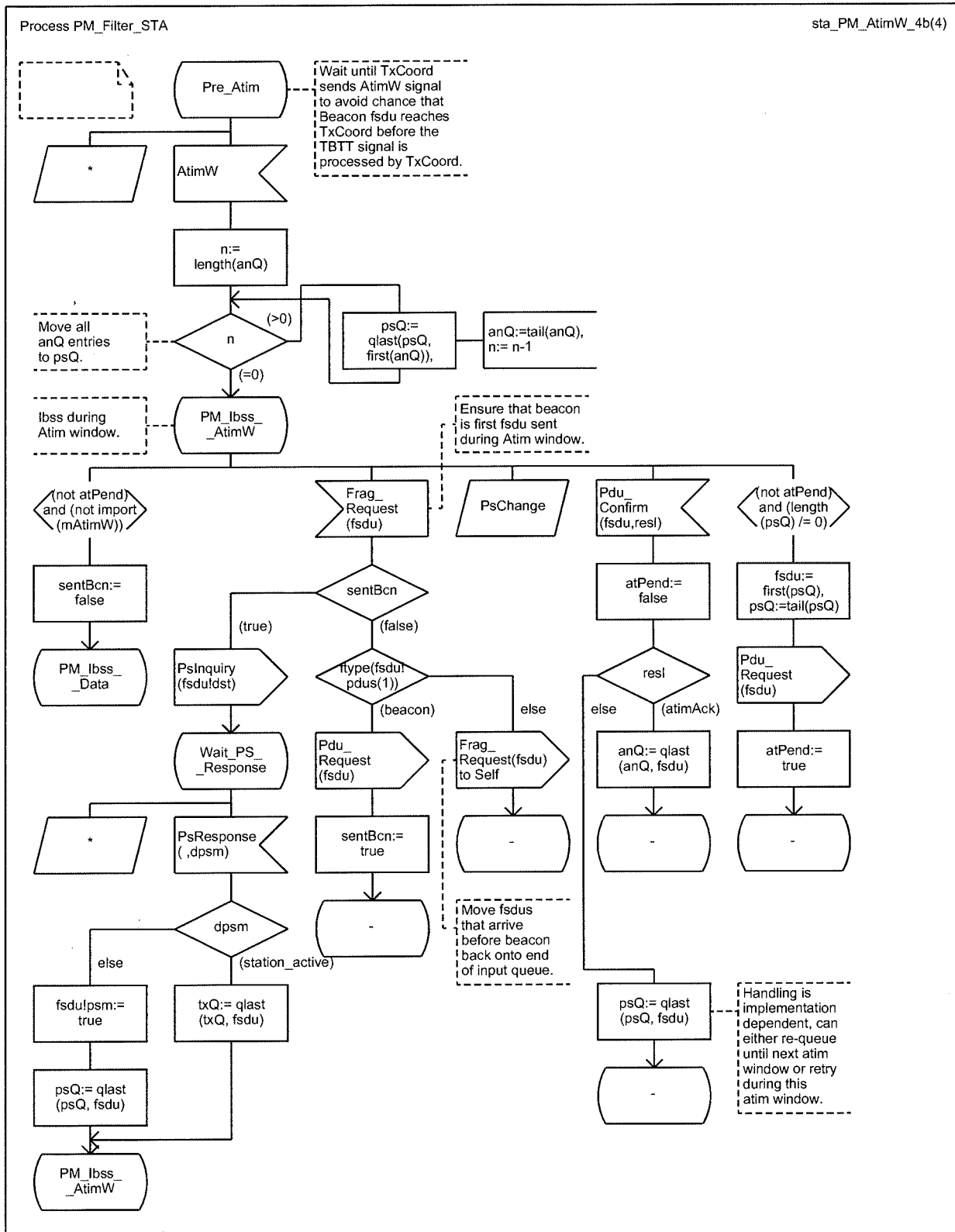


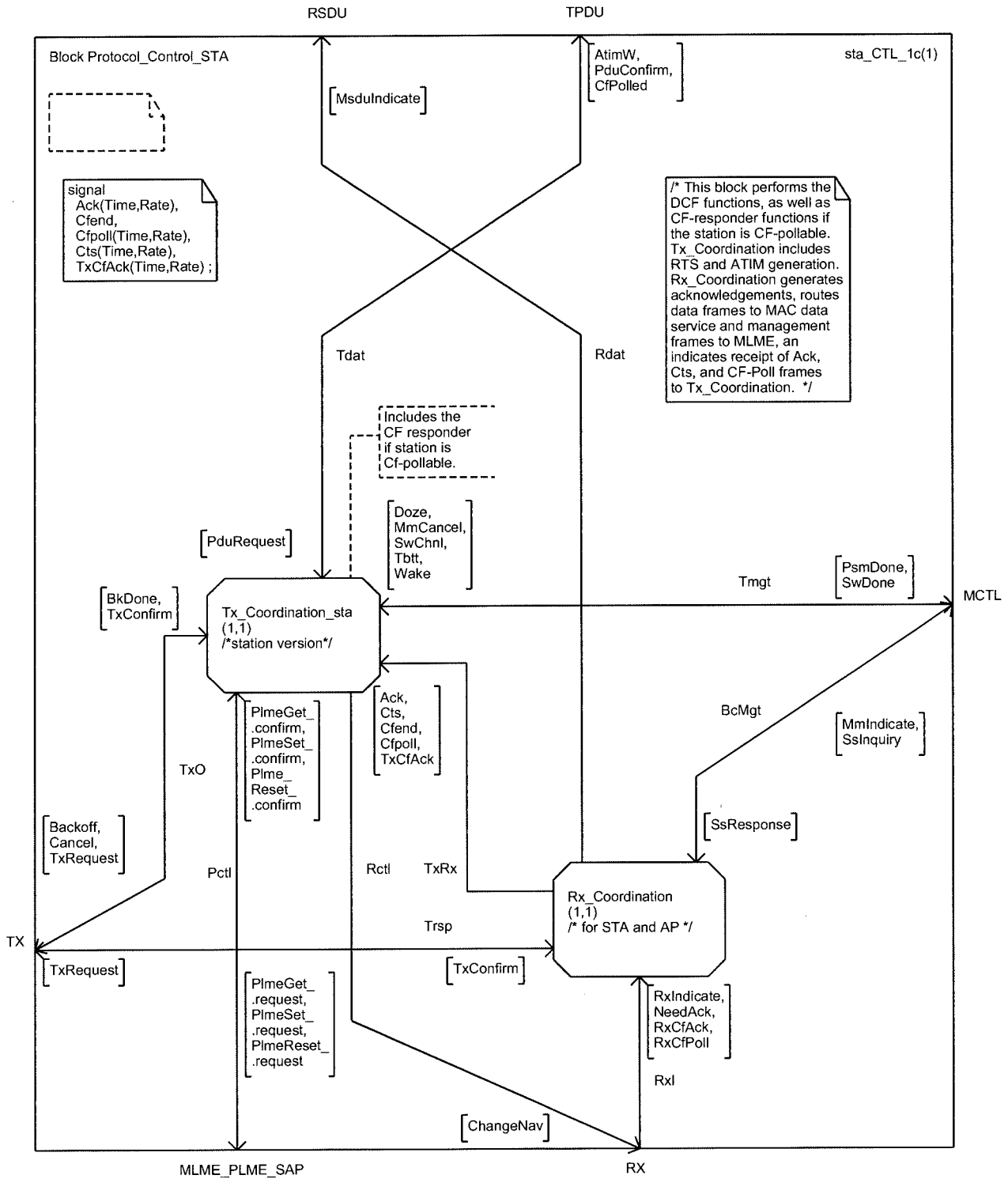


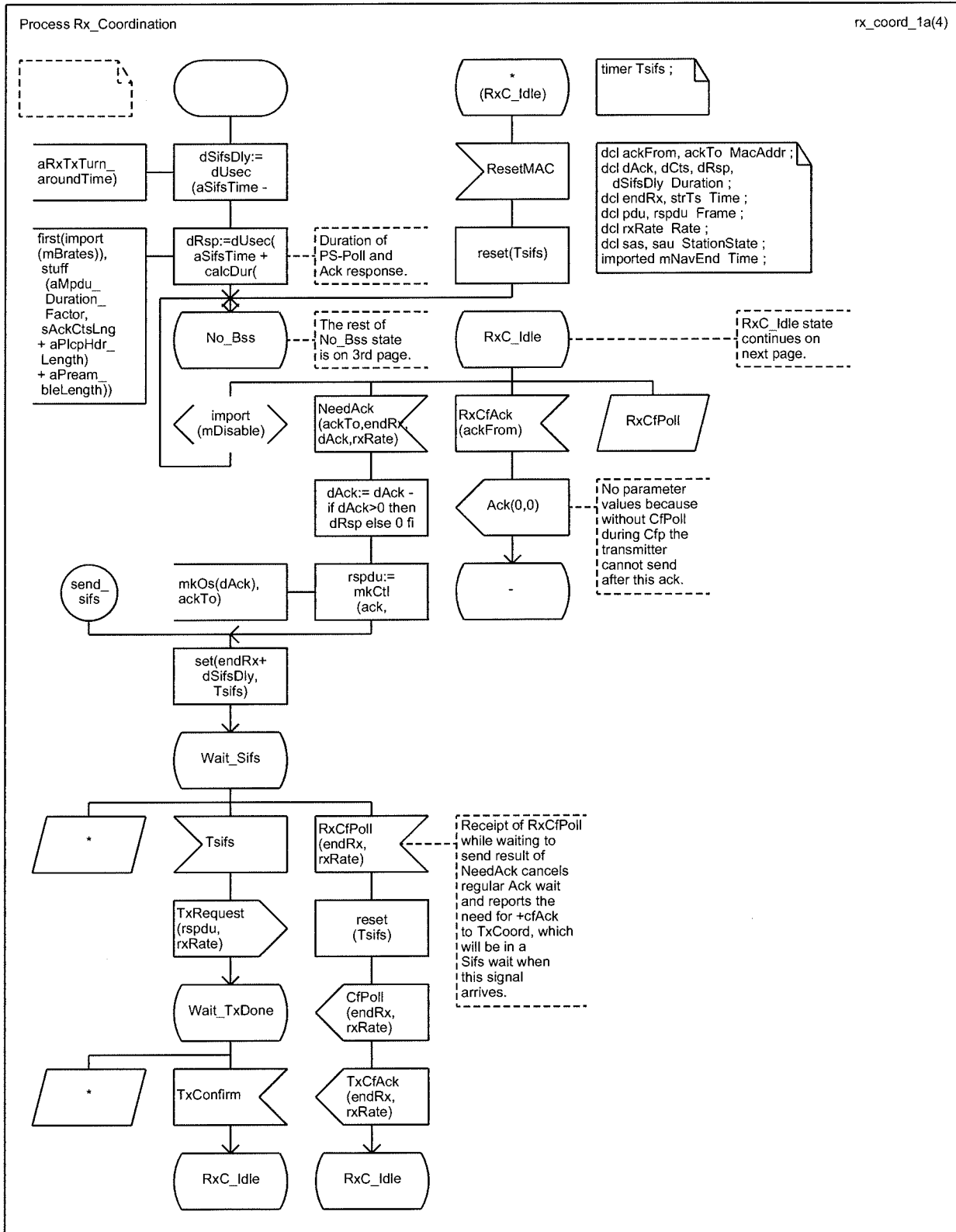


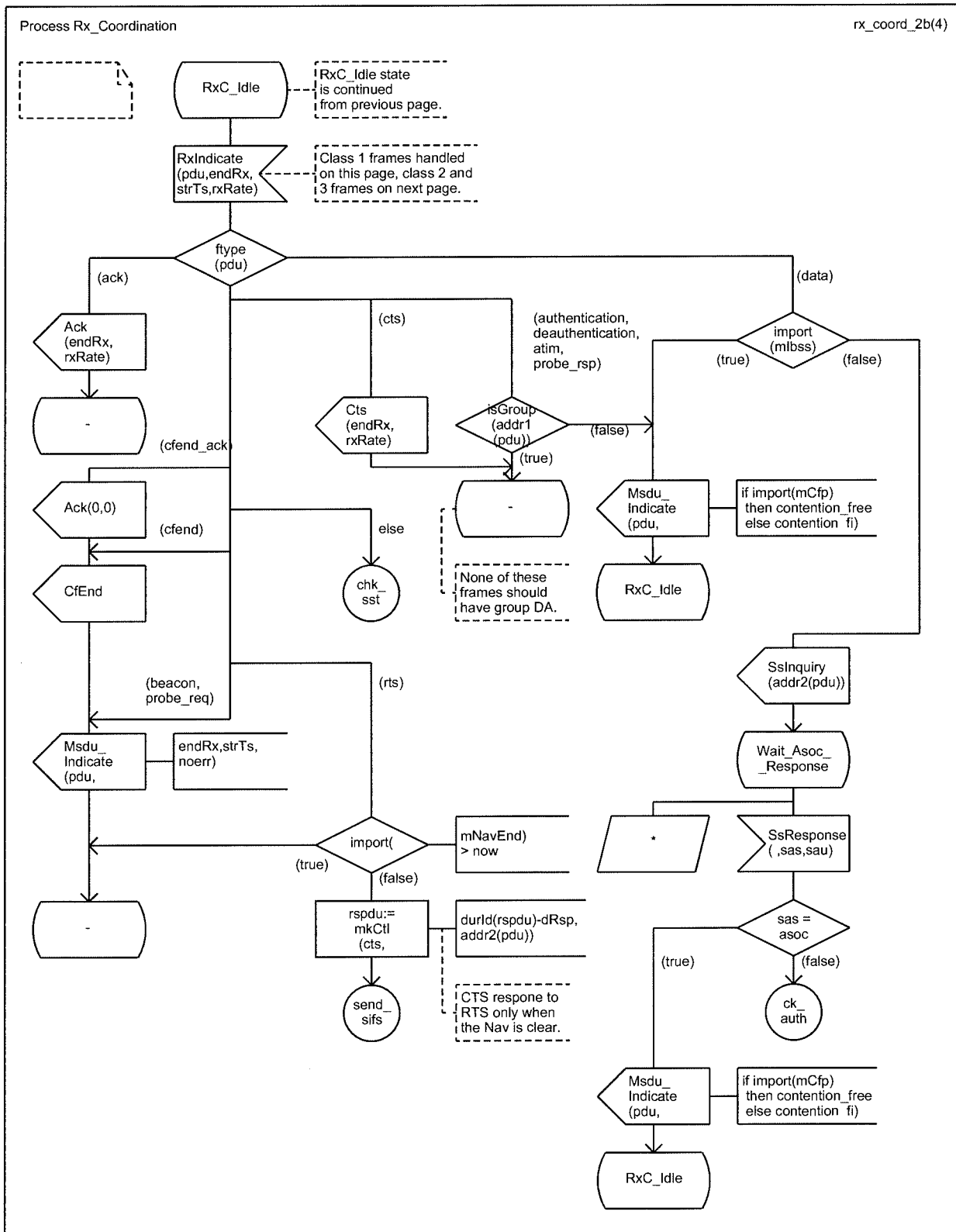


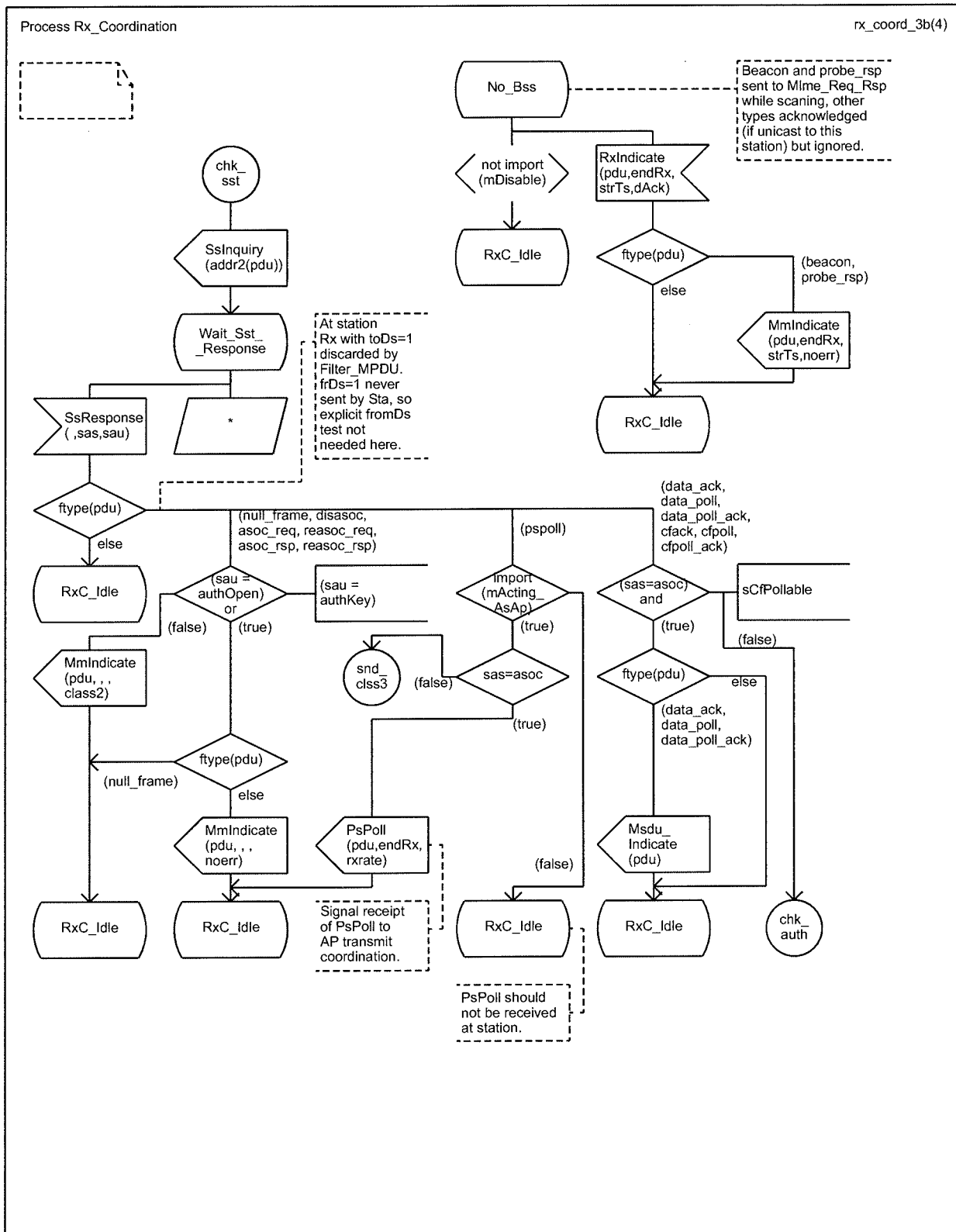


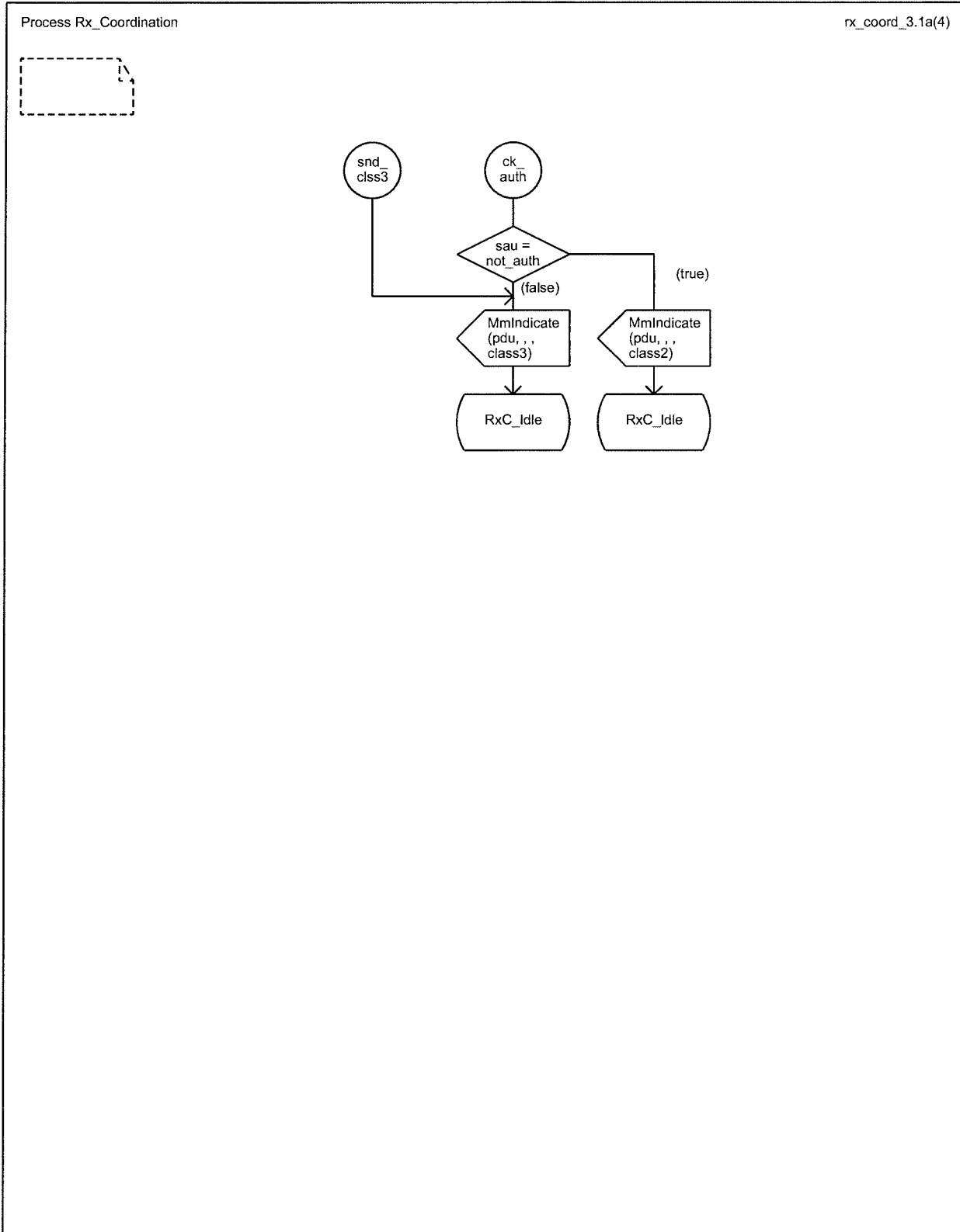


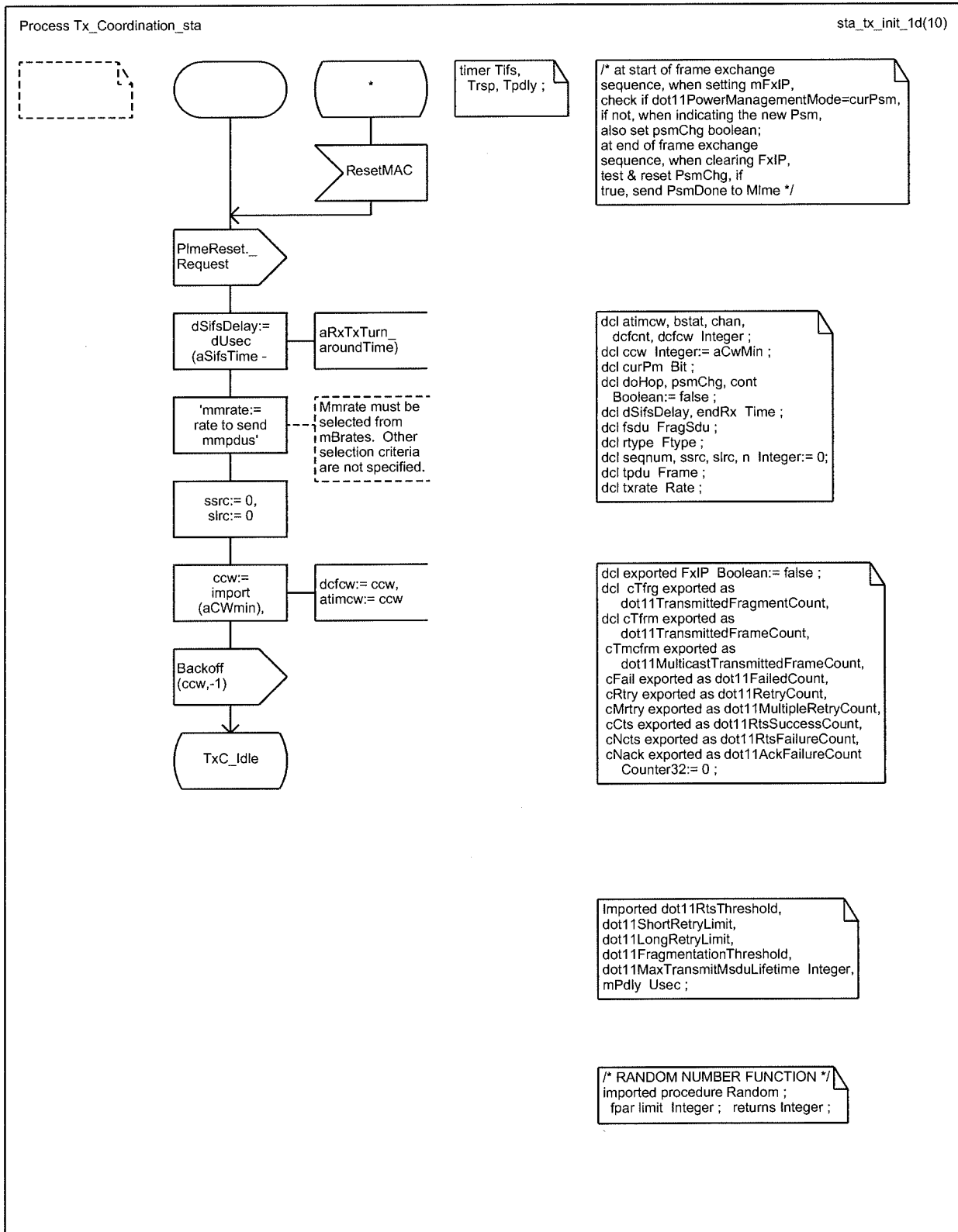


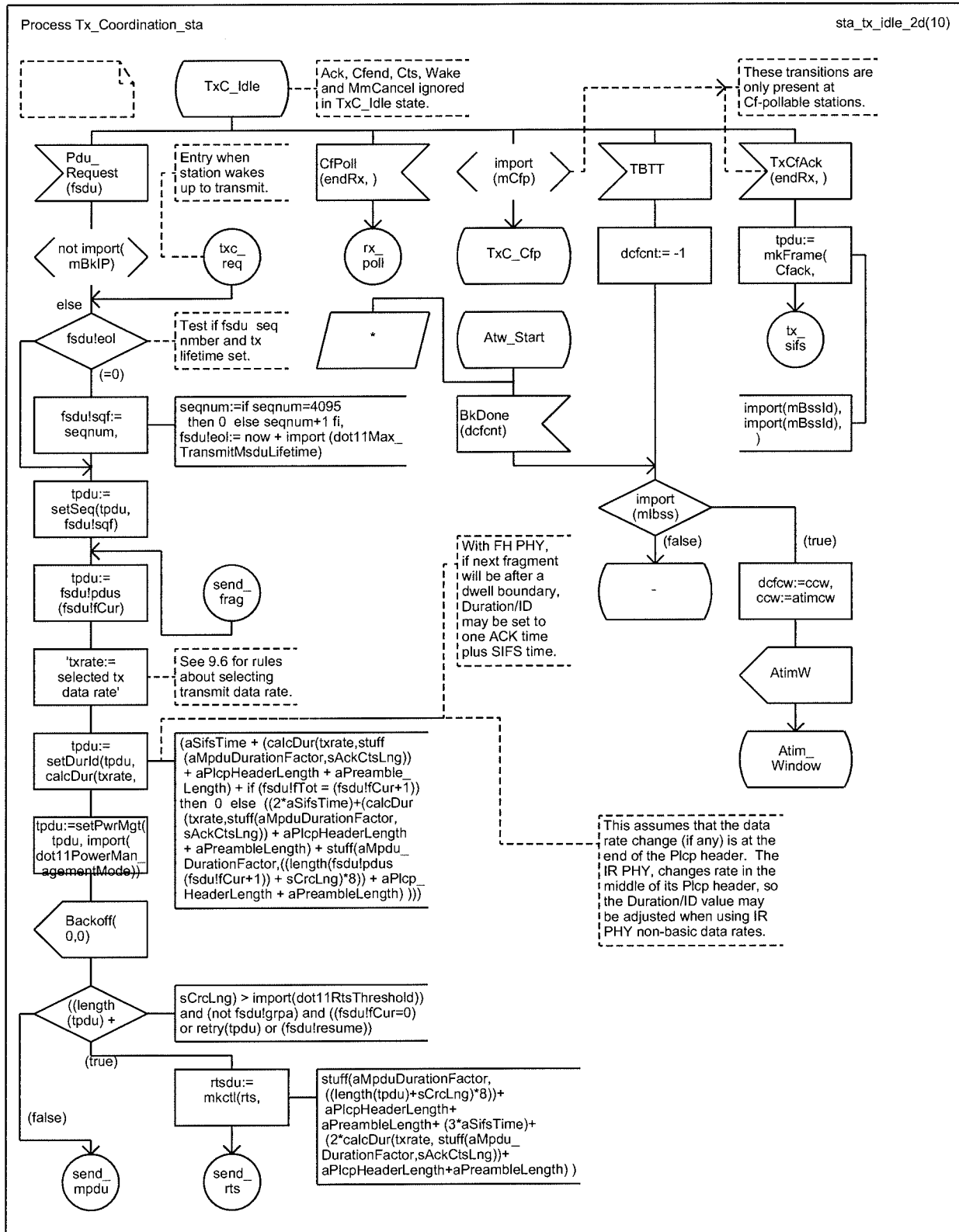


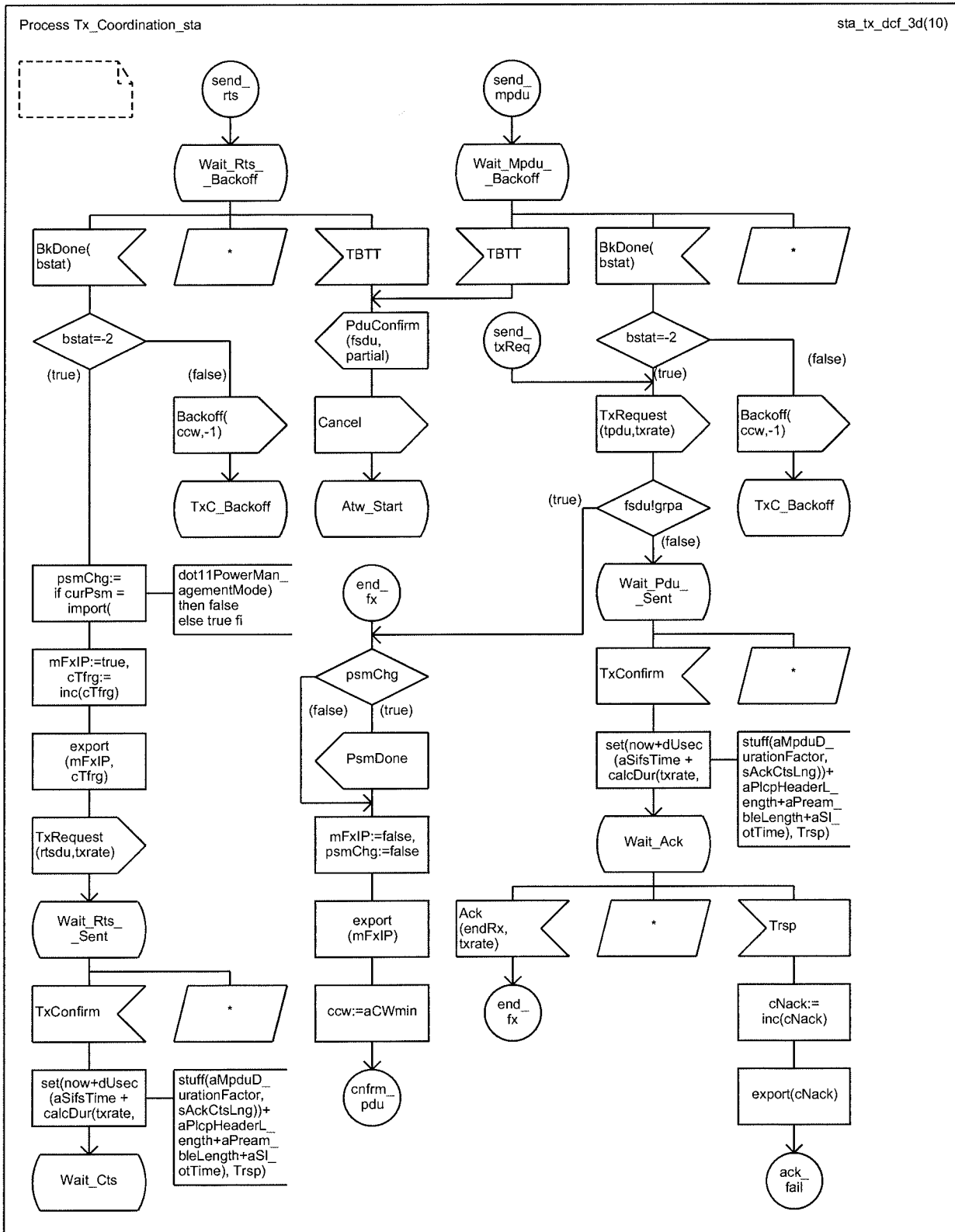


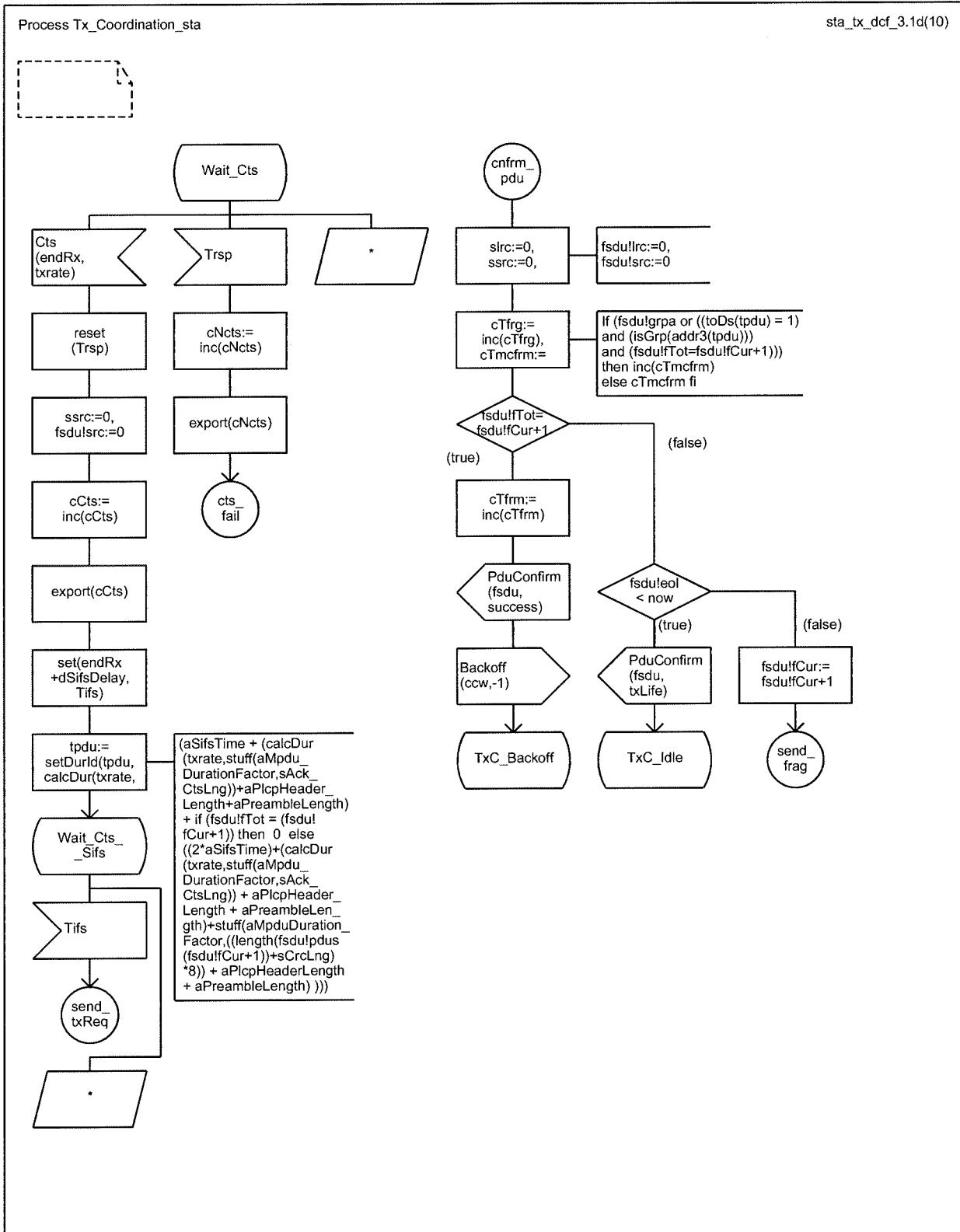


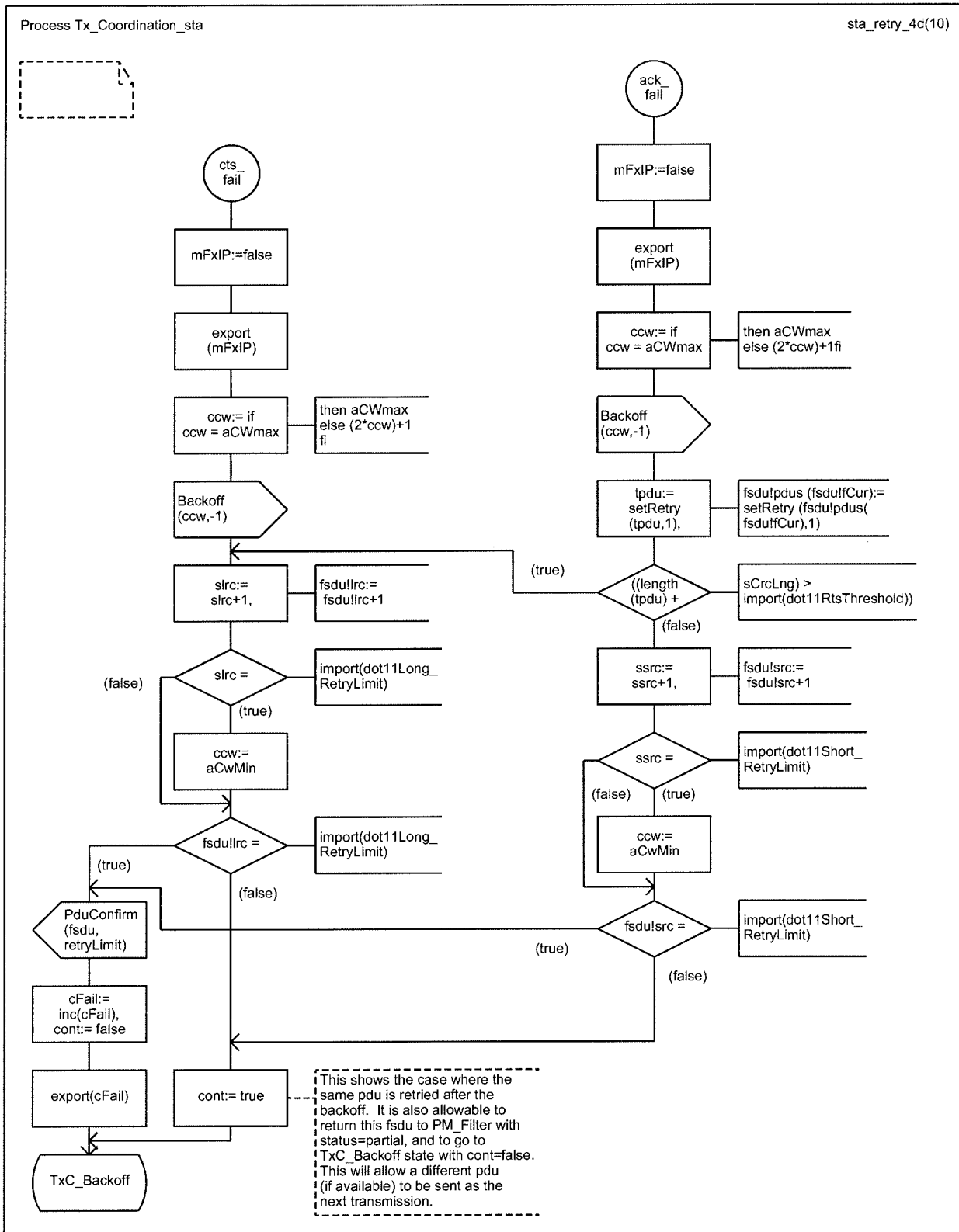


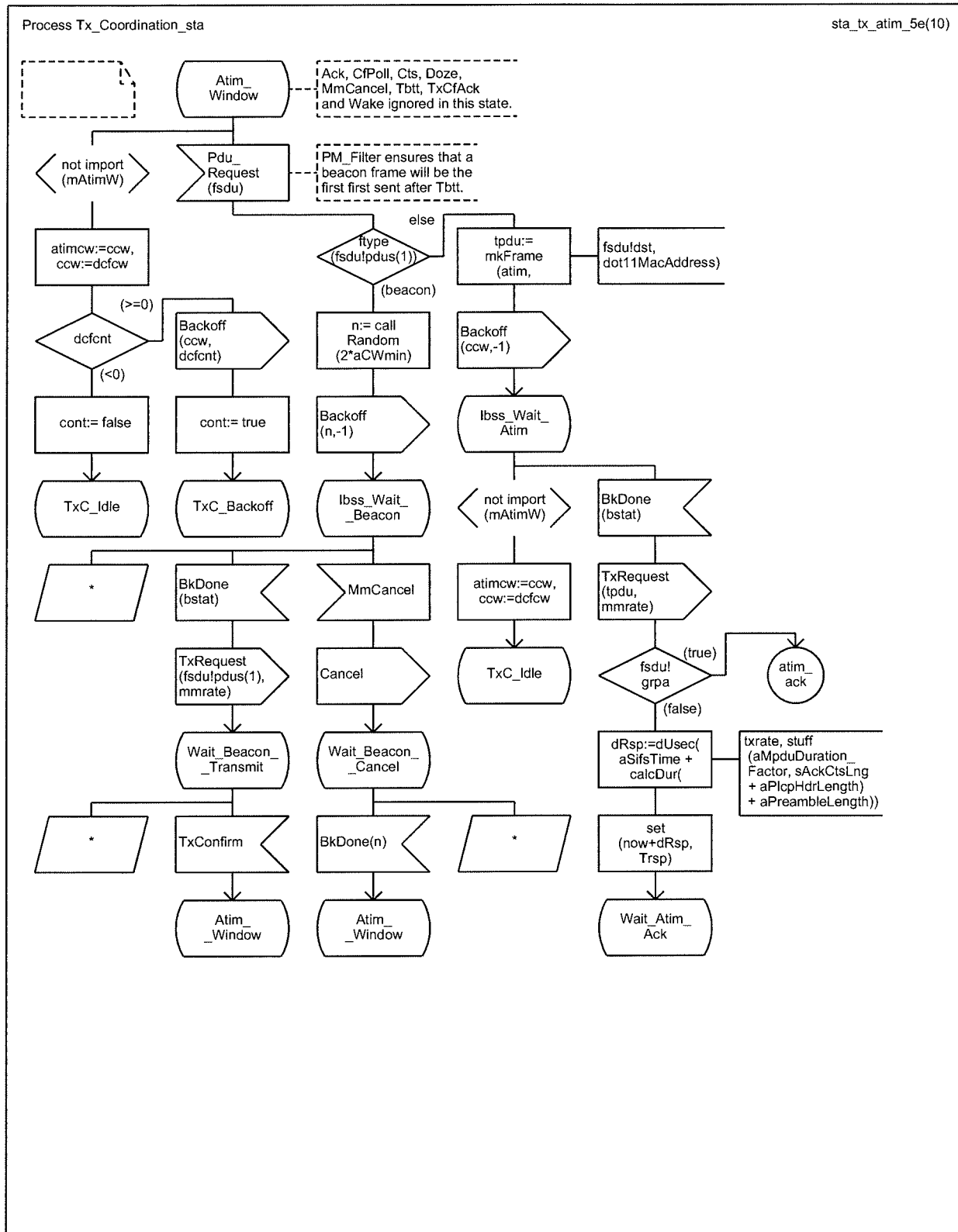






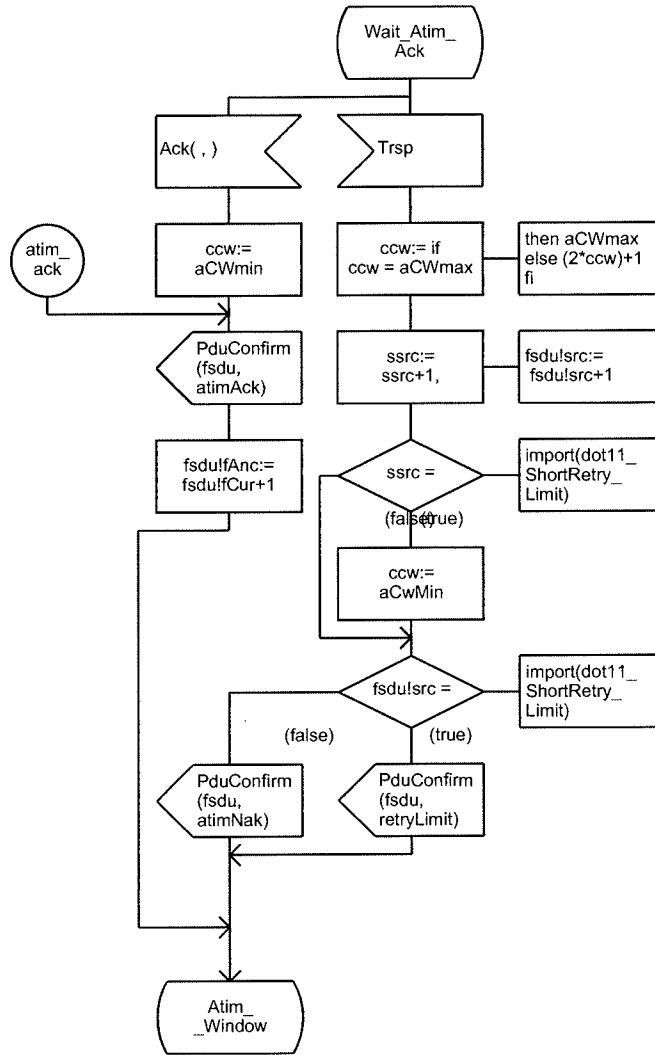


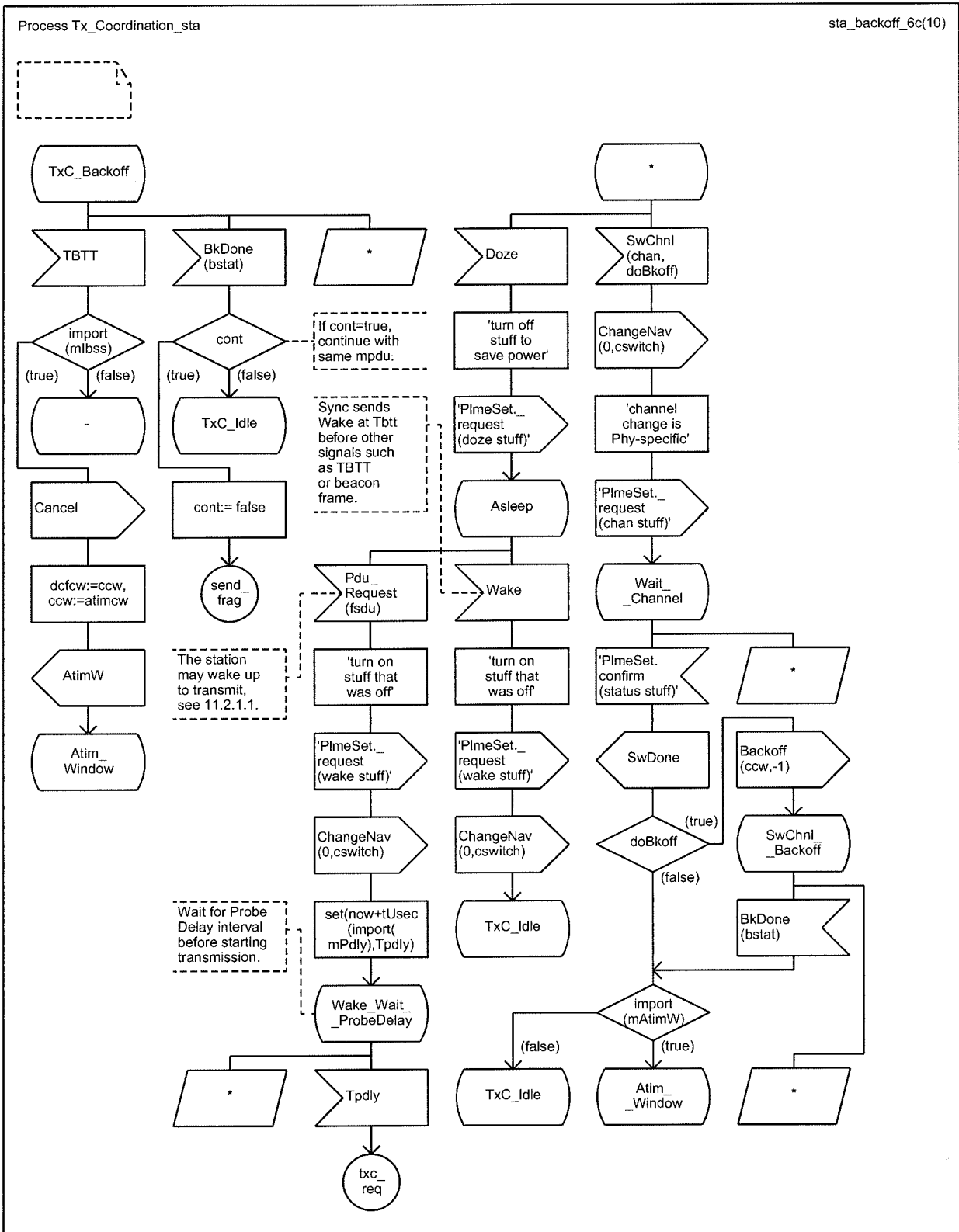


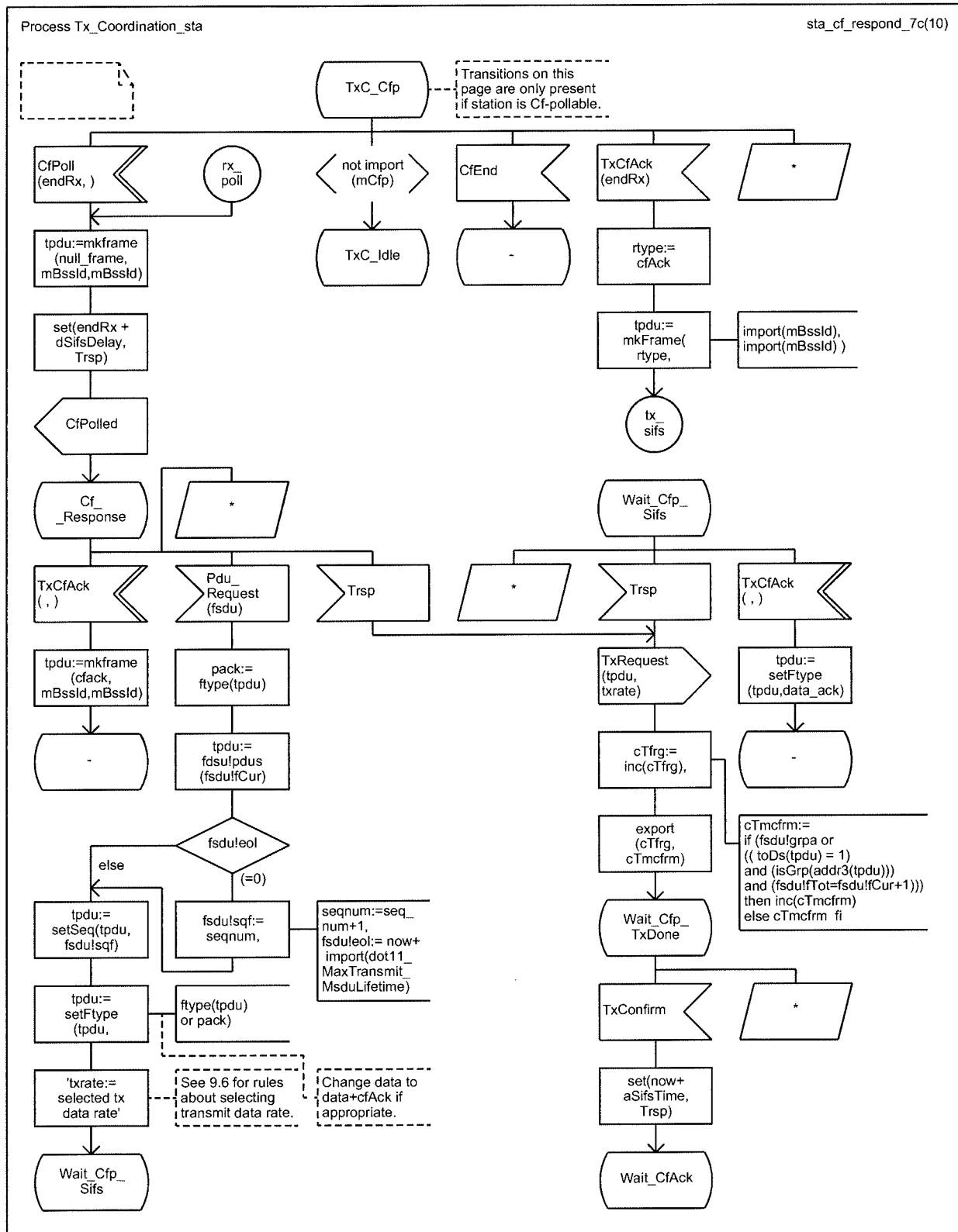


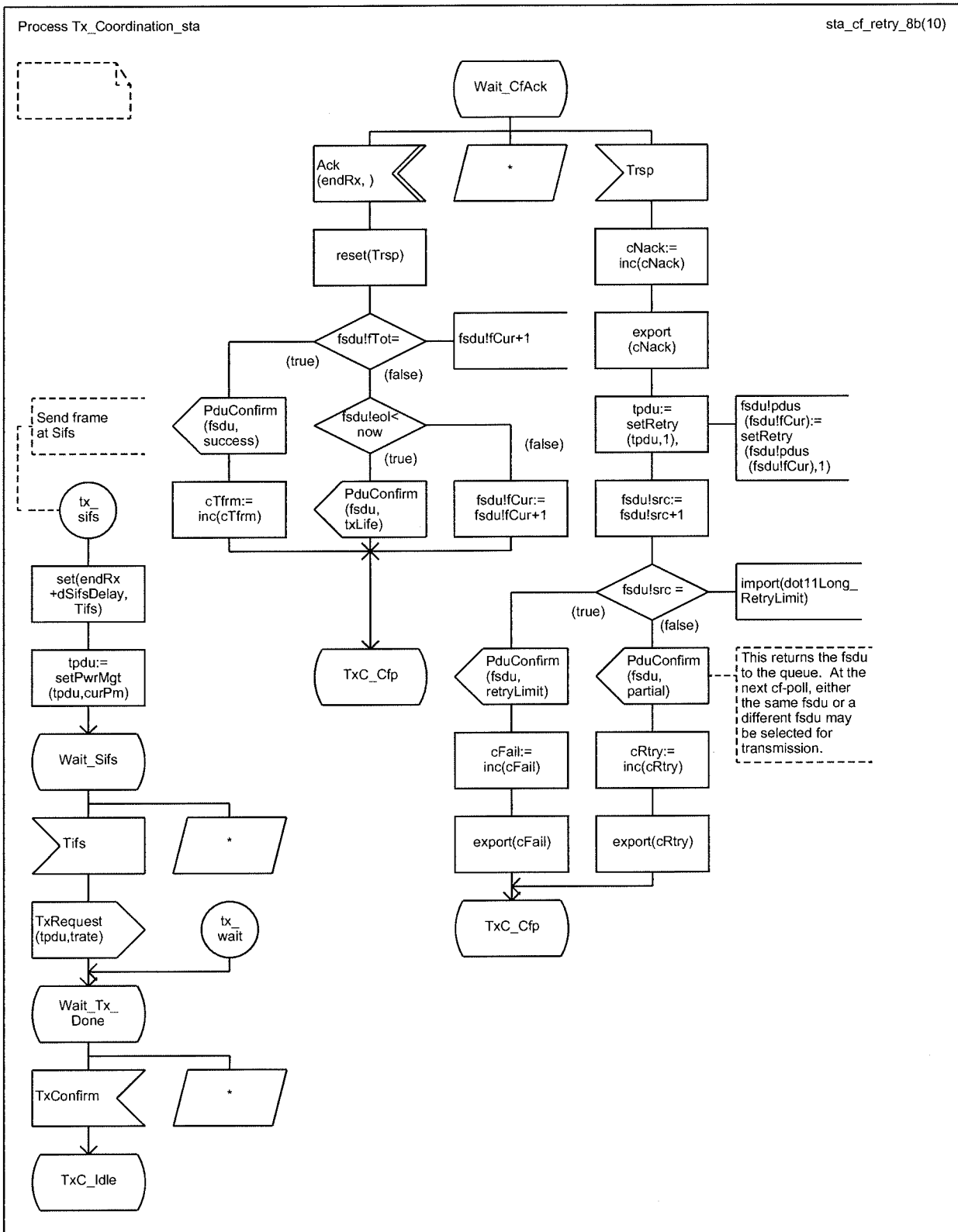
Process Tx_Coordination_sta

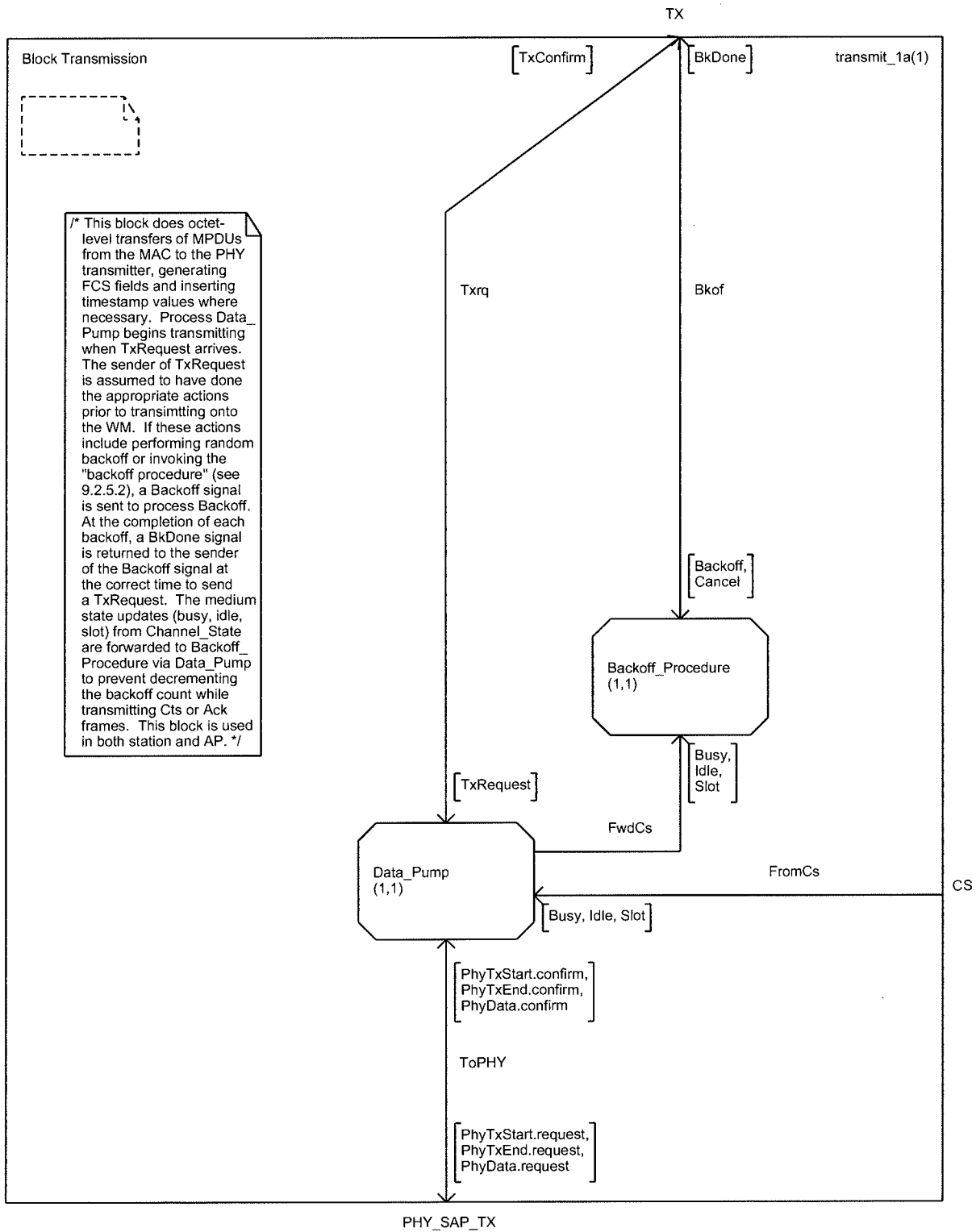
sta_tx_atim_5.1a(10)

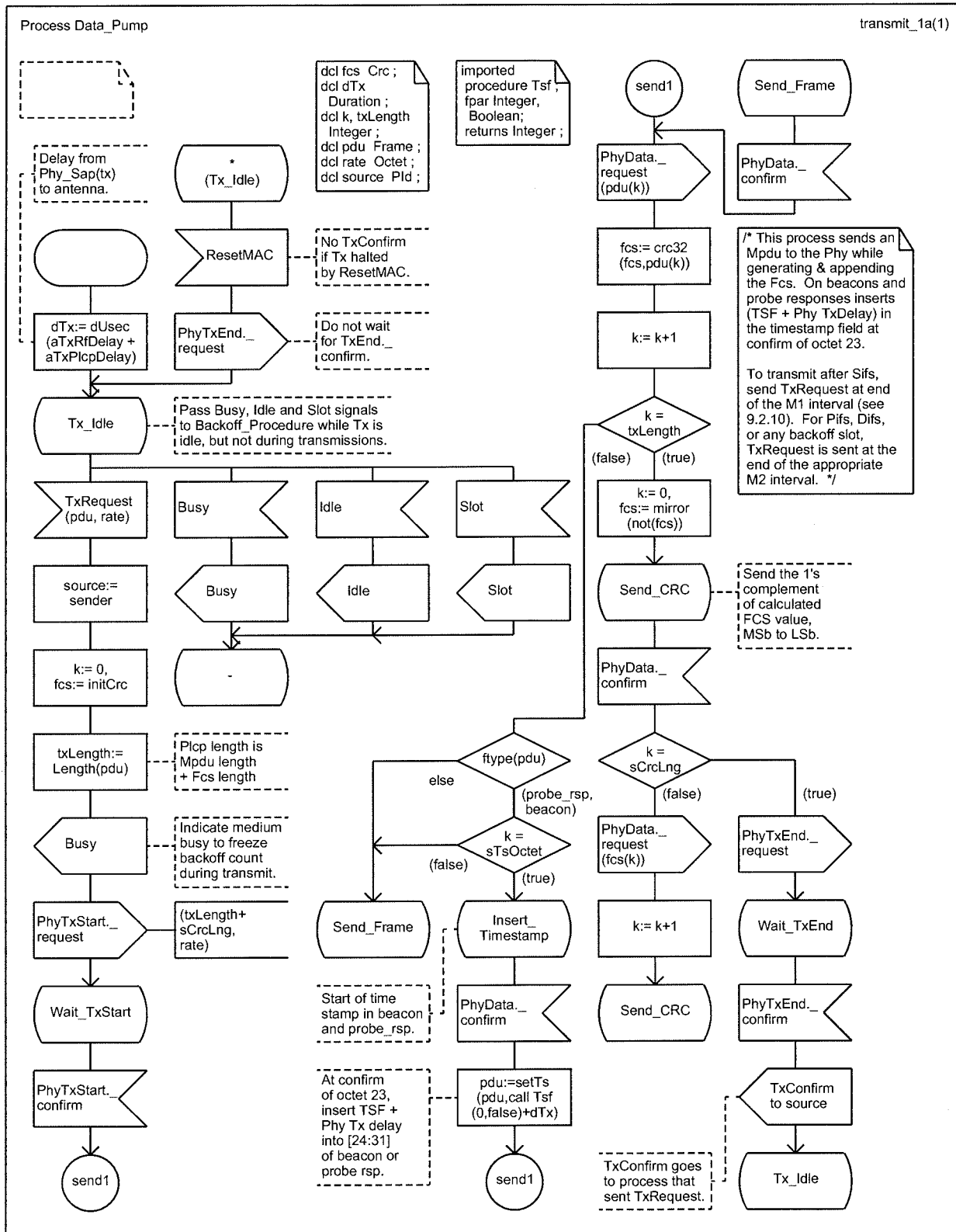


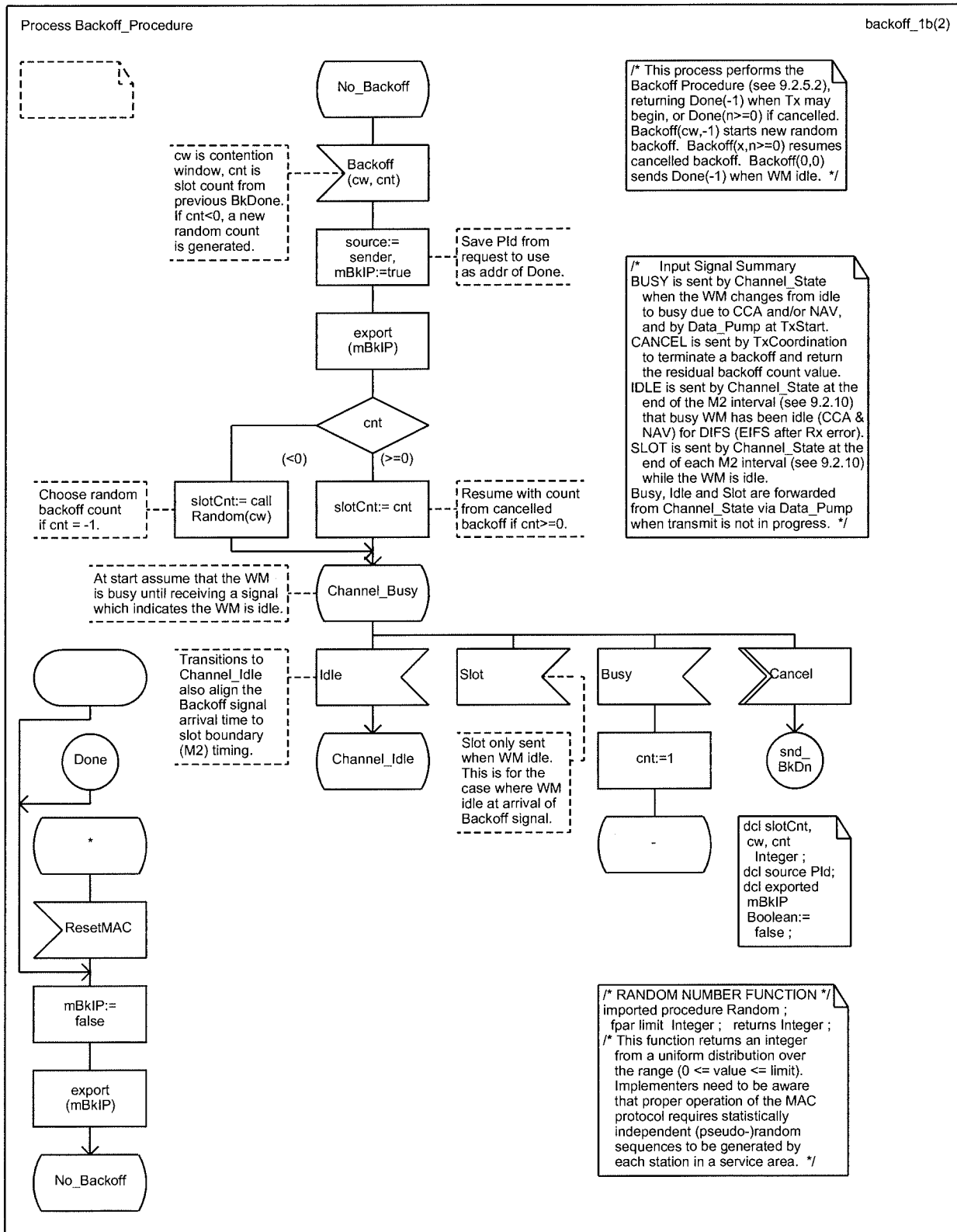


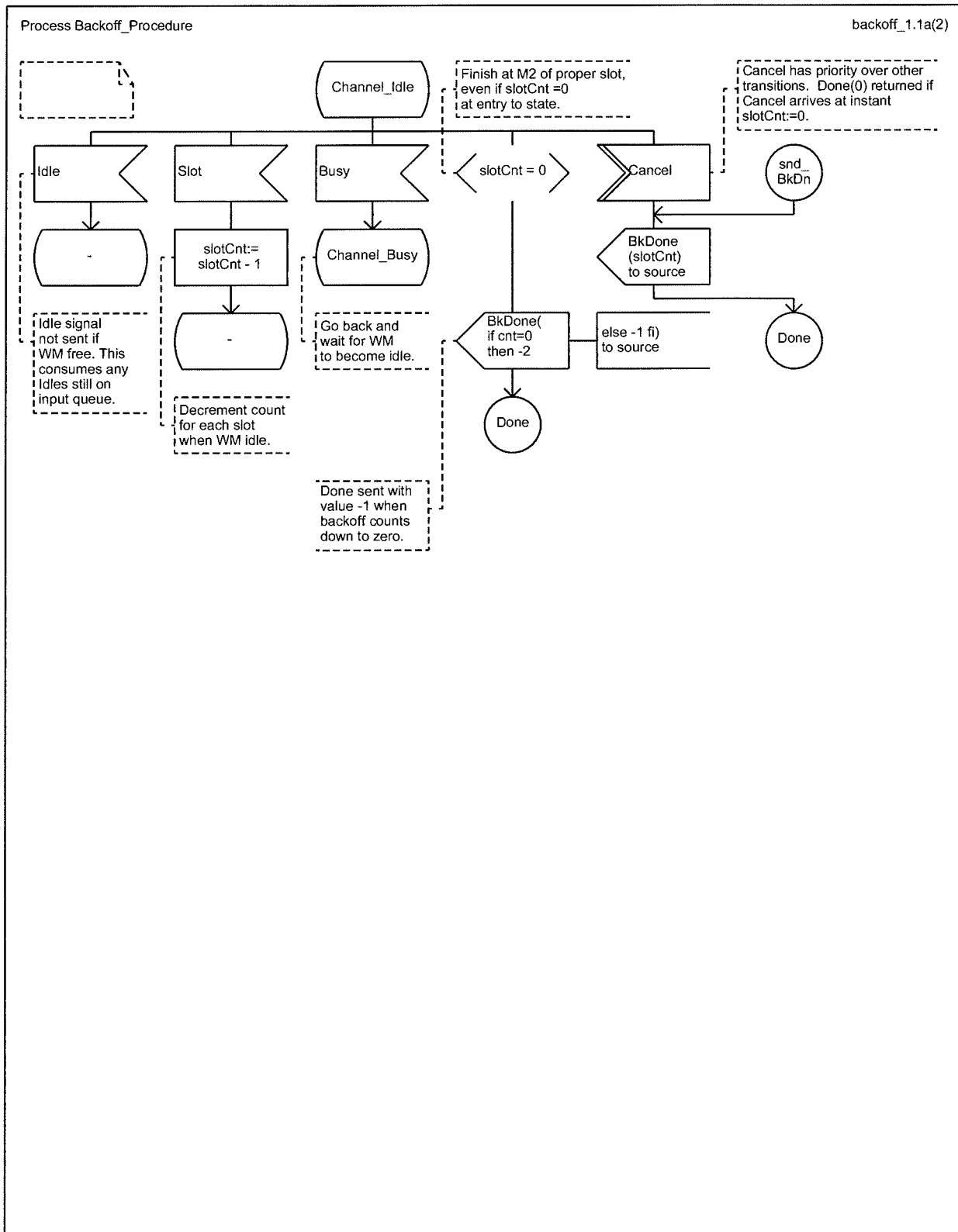


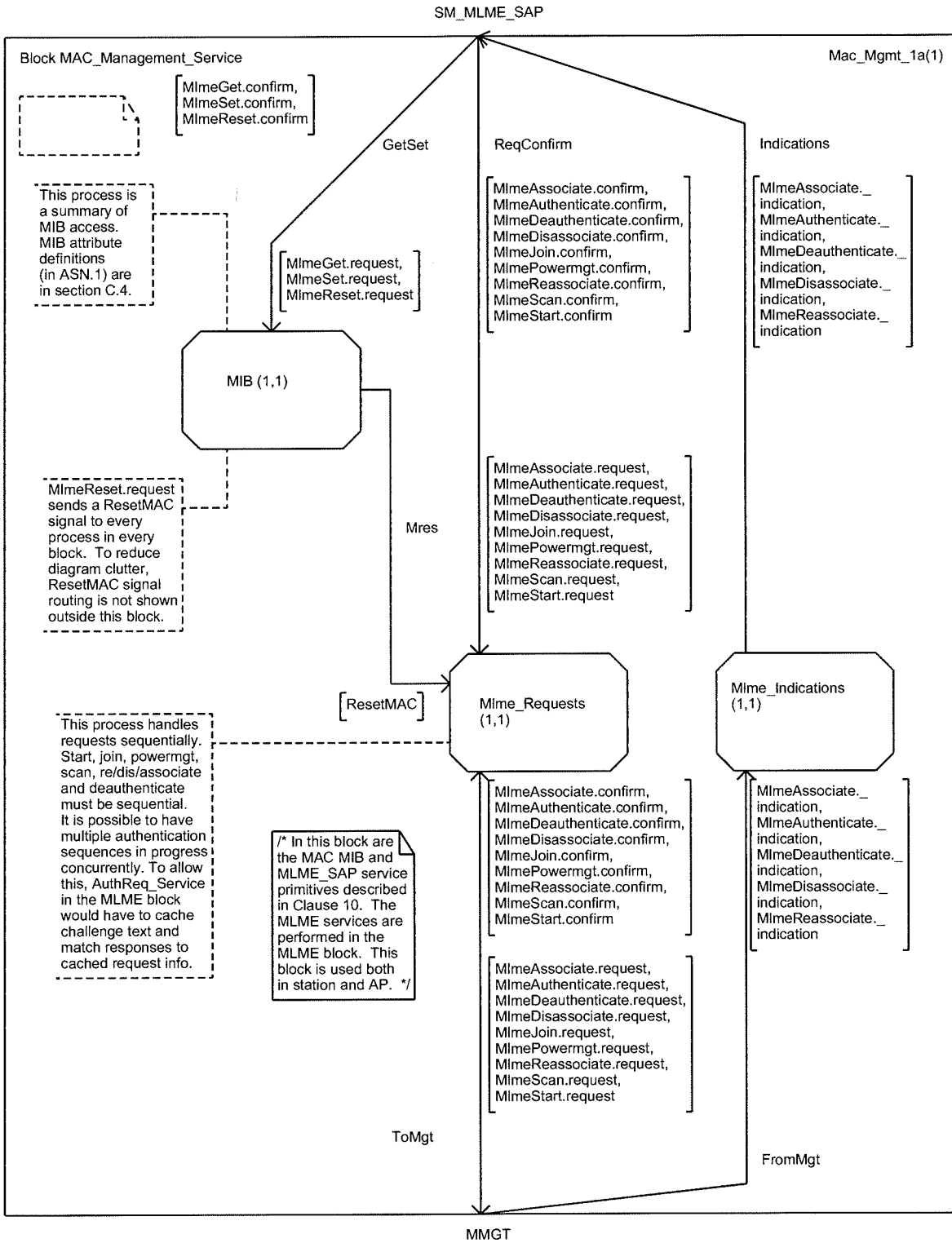


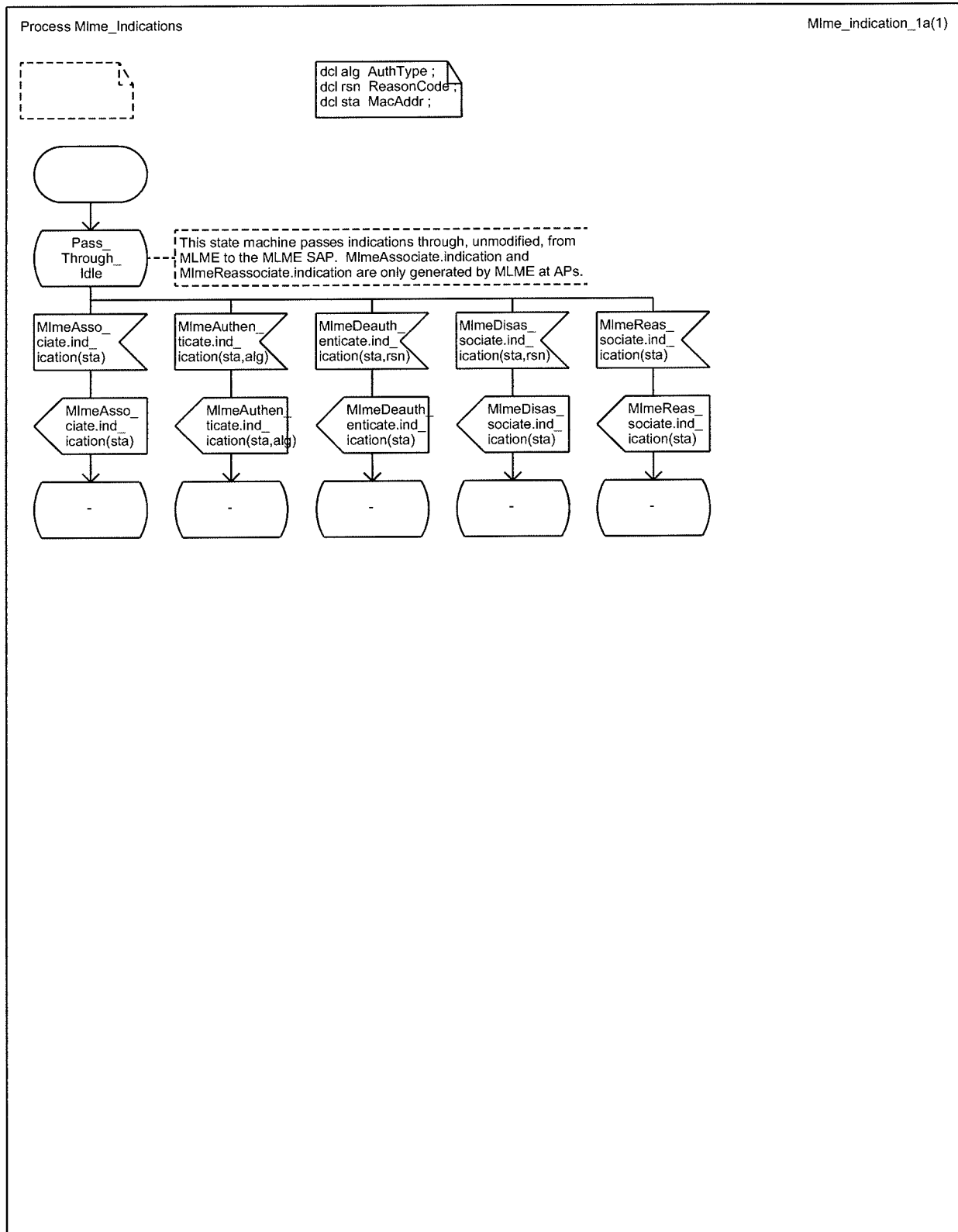


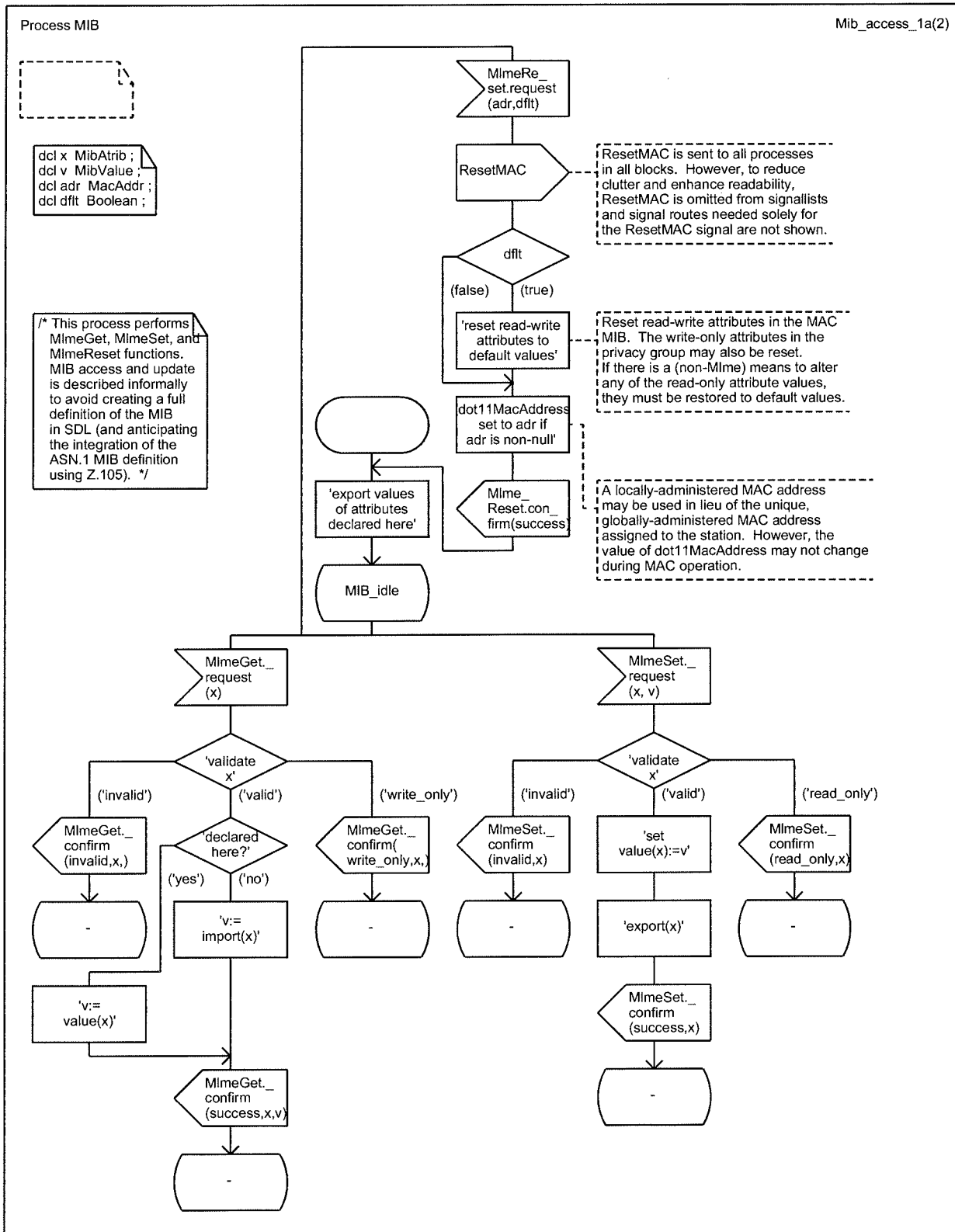












Process MIB

Mib_import_export_2b(2)



```

/* Import of {Read-Only} MIB counter
values exported from other processes */
imported
dot11AckFailureCount,
dot11FailedCount,
dot11FcsErrorCount,
dot11FrameDuplicateCount,
dot11MulticastReceivedFrameCount,
dot11MulticastTransmittedFrameCount,
dot11MultipleRetryCount,
dot11ReceivedFragmentCount,
dot11RetryCount,
dot11RtsFailureCount,
dot11RtsSuccessCount,
dot11TransmittedFragmentCount,
dot11WepExcludedCount,
dot11WepIcvErrorCount,
dot11WepUndecryptableCount Counter32 ;

```

```

/* Declarations of MIB attributes exported from
this process */

/* Read-Write attributes */
dcl exported
dot11AuthenticationAlgorithms AuthTypeSet:=
incl(open_system, shared_key),
dot11ExcludeUnencrypted Boolean:= false,
dot11FragmentationThreshold Integer:= 2346,
dot11GroupAddresses MacAddrSet:= empty,
dot11LongRetryLimit Integer:= 4,
dot11MaxReceiveLifetime Kusec:= 512,
dot11MaxTransmitMsduLifetime Kusec:= 512,
dot11MediumOccupancyLimit Kusec:= 100,
dot11PrivacyInvoked Boolean:= false,
mReceiveDTIMs Boolean:= true,
dot11CfpPeriod Integer:= 1,
dot11CfpMaxDuration Kusec:= 200,
dot11AuthenticationResponseTimeout Kusec:= 512,
dot11RtsThreshold Integer:= 3000,
dot11ShortRetryLimit Integer:= 7,
dot11WepDefaultKeyId KeyIndex:= 0,
dot11CurrentChannelNumber Integer:= 0,
dot11CurrentSet Integer:= 0,
dot11CurrentPattern Integer:= 0,
dot11CurrentIndex Integer:= 0 ;

/* Write-Only attributes */
dcl exported
dot11WepDefaultKeys KeyVector:= nullKey,
dot11WepKeyMappings
KeyMapArray:= (. nullAddr, false, nullKey .) ;

```

```

/* The following Read-Only attributes in the
MAC MIB are defined as synonyms (named
constants) rather than remote variables
because they describe properties of the
station which are static, at least during
any single instance of MAC operation:
dot11AuthenticationAlgorithms AuthTypeSet,
dot11CfPollable Boolean,
dot11MacAddress MacAddr,
dot11ManufacturerID Octetstring,
dot11PrivacyOptionImplemented Boolean,
dot11ProductID Octetstring,
aStationID MacAddr,
dot11WepKeyMappingLength Integer ;

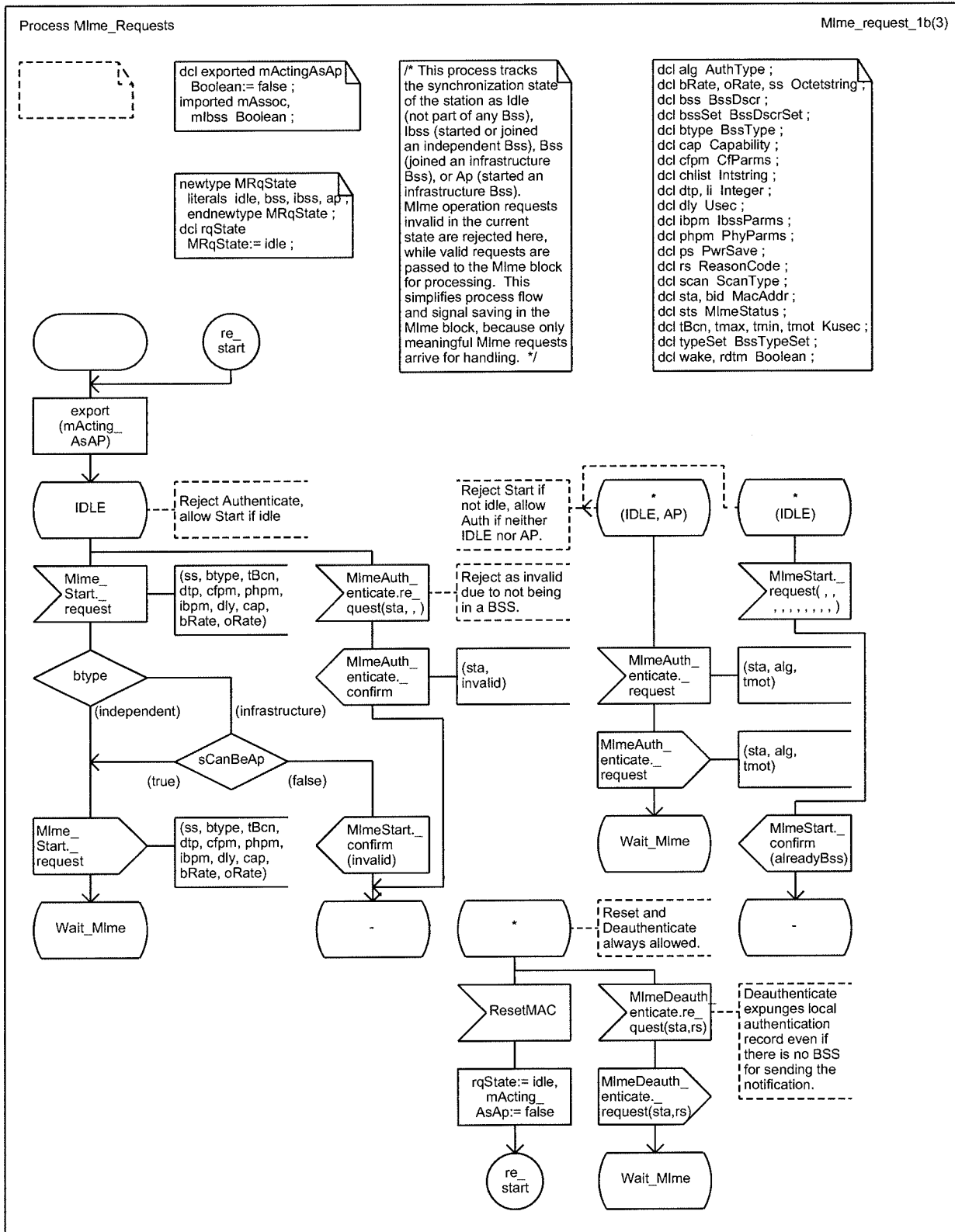
In addition, all Read-Only attributes in the
PHY MIB which are accessed by the MAC
are defined as synonyms.
*/

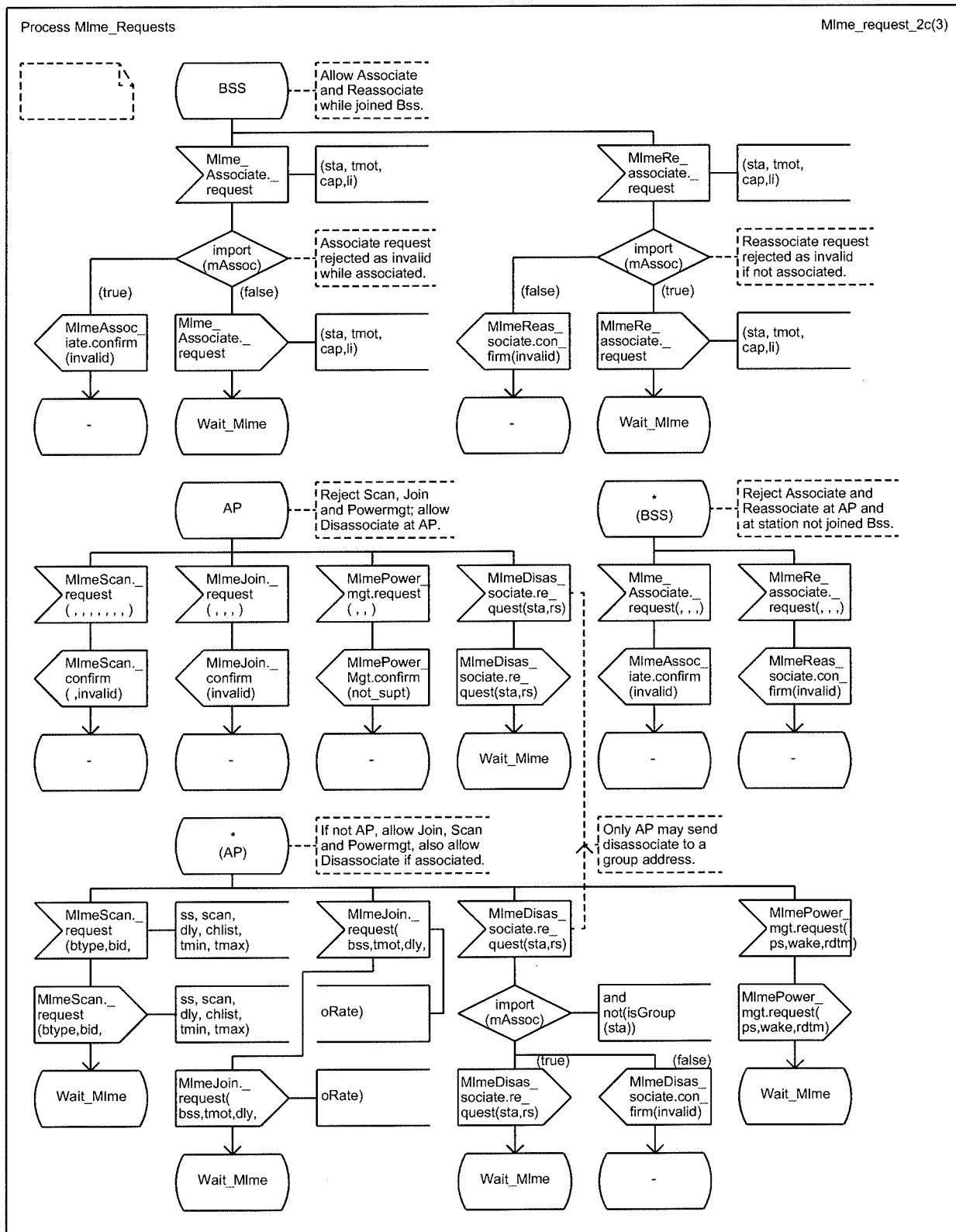
```

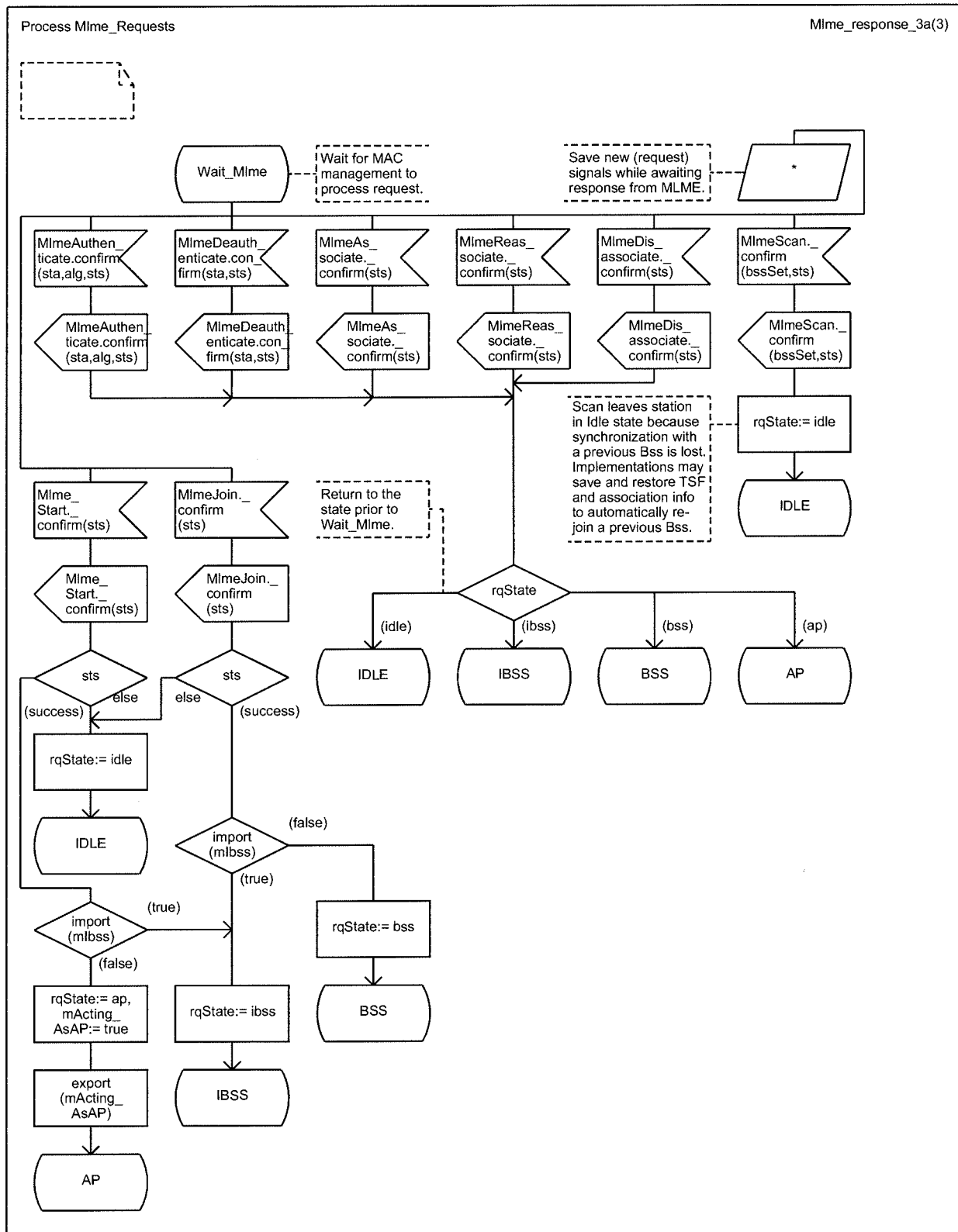
```

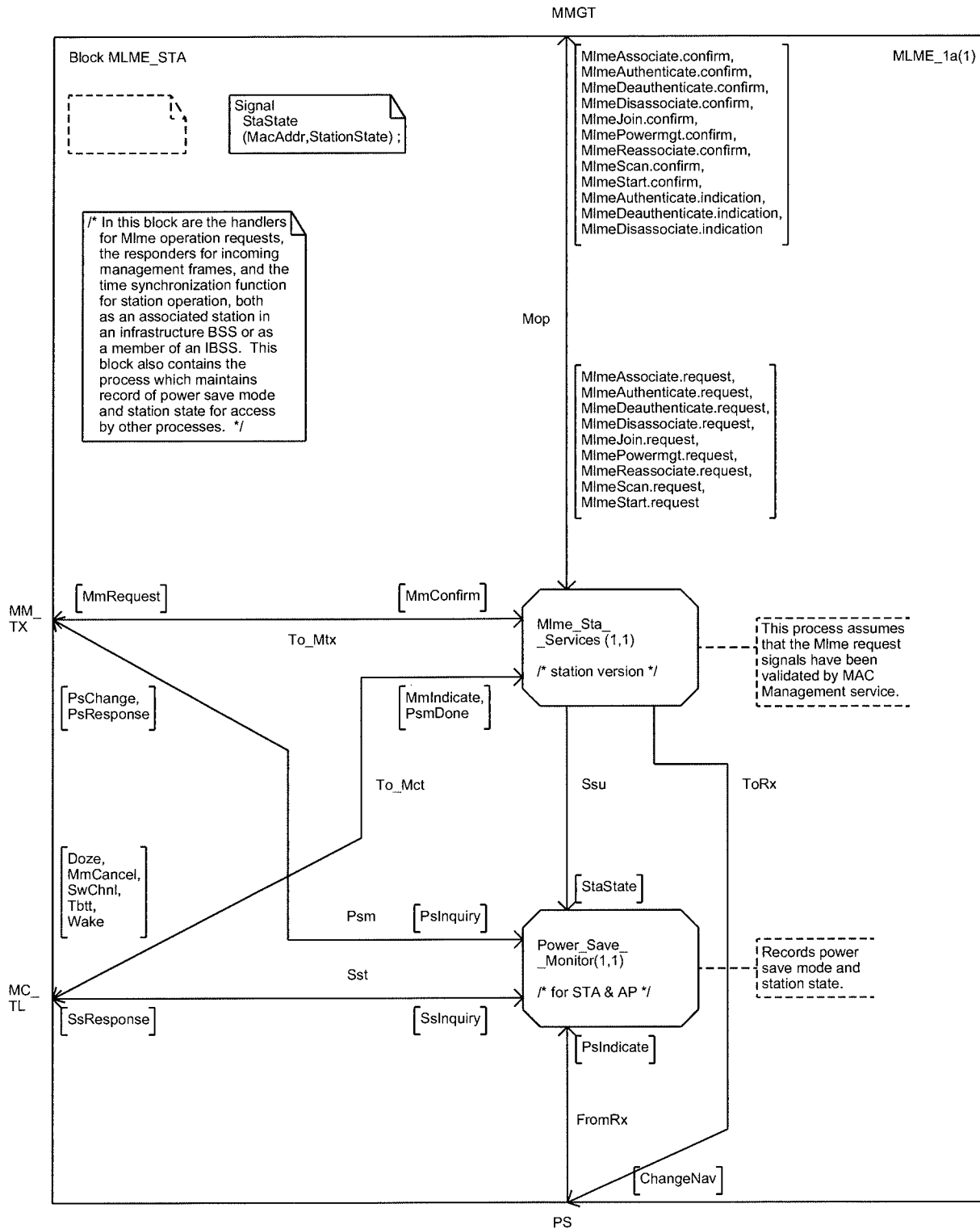
/* NOTE:
The values listed for MAC MIB attributes are the
specified default values for those attributes.
The values listed for PHY MIB attributes are either
the default values for the FH PHY, or arbitrary
values within the specified range. The specific
values for PHY attributes in this SDL description
of the MAC do not have normative significance.
*/

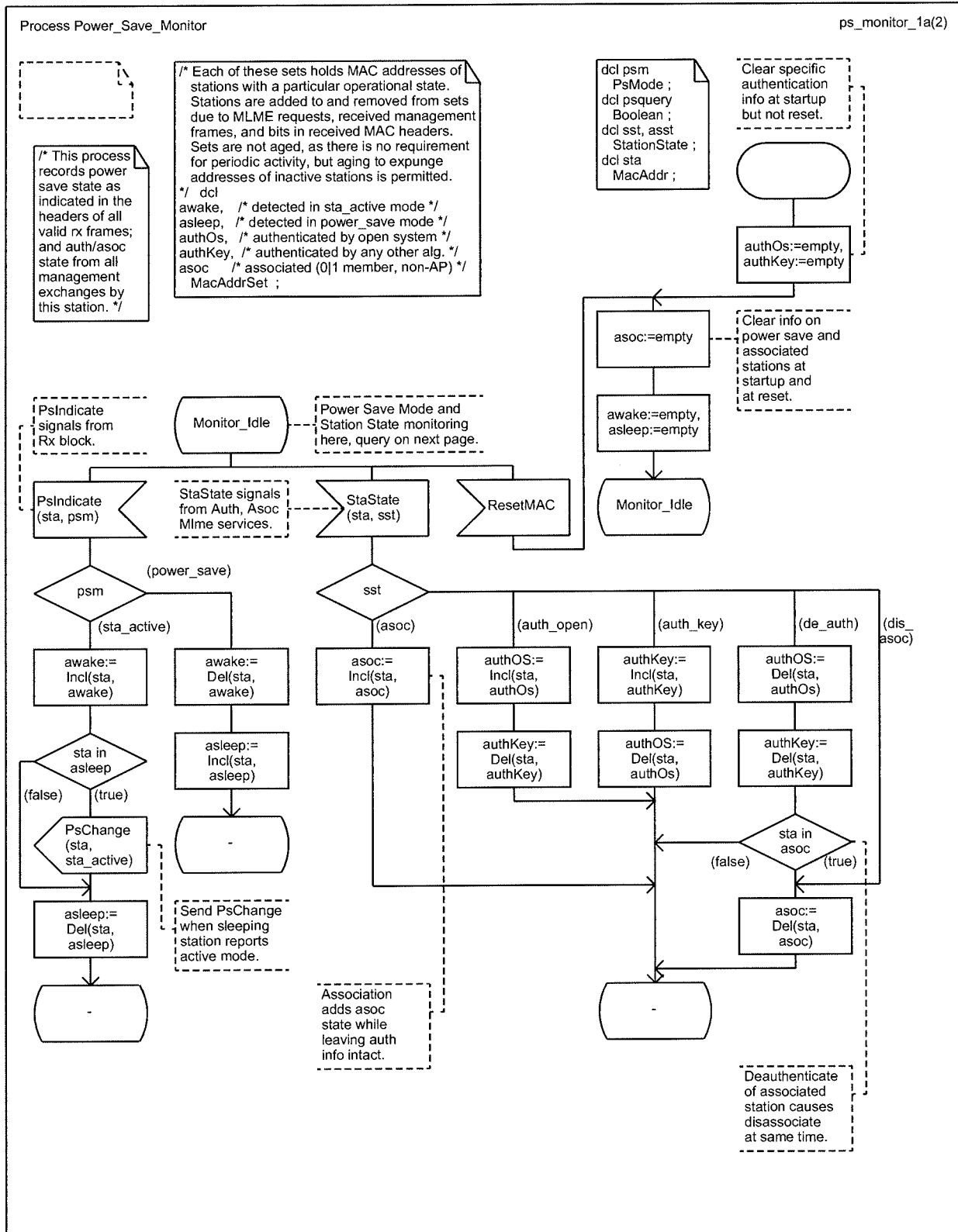
```

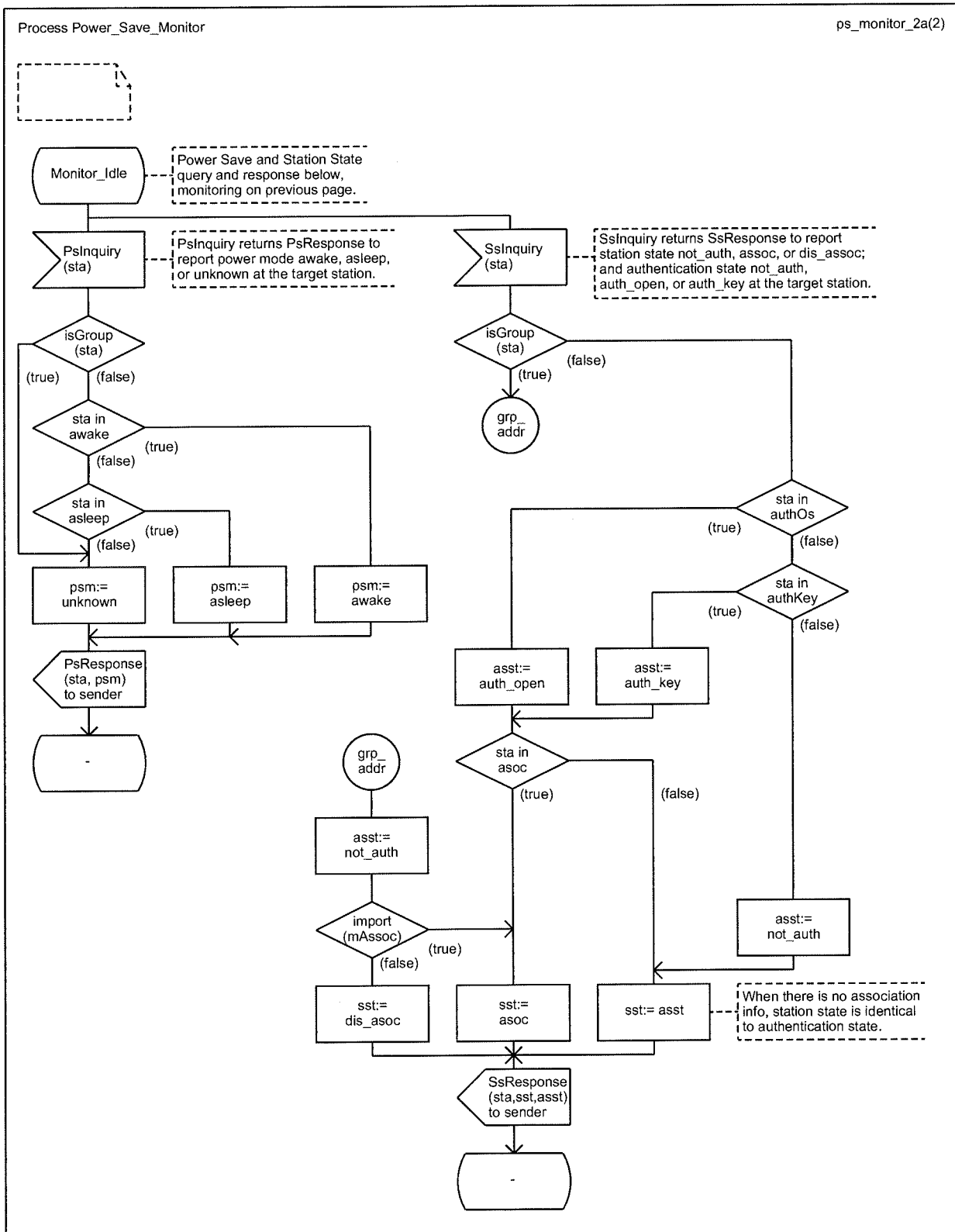


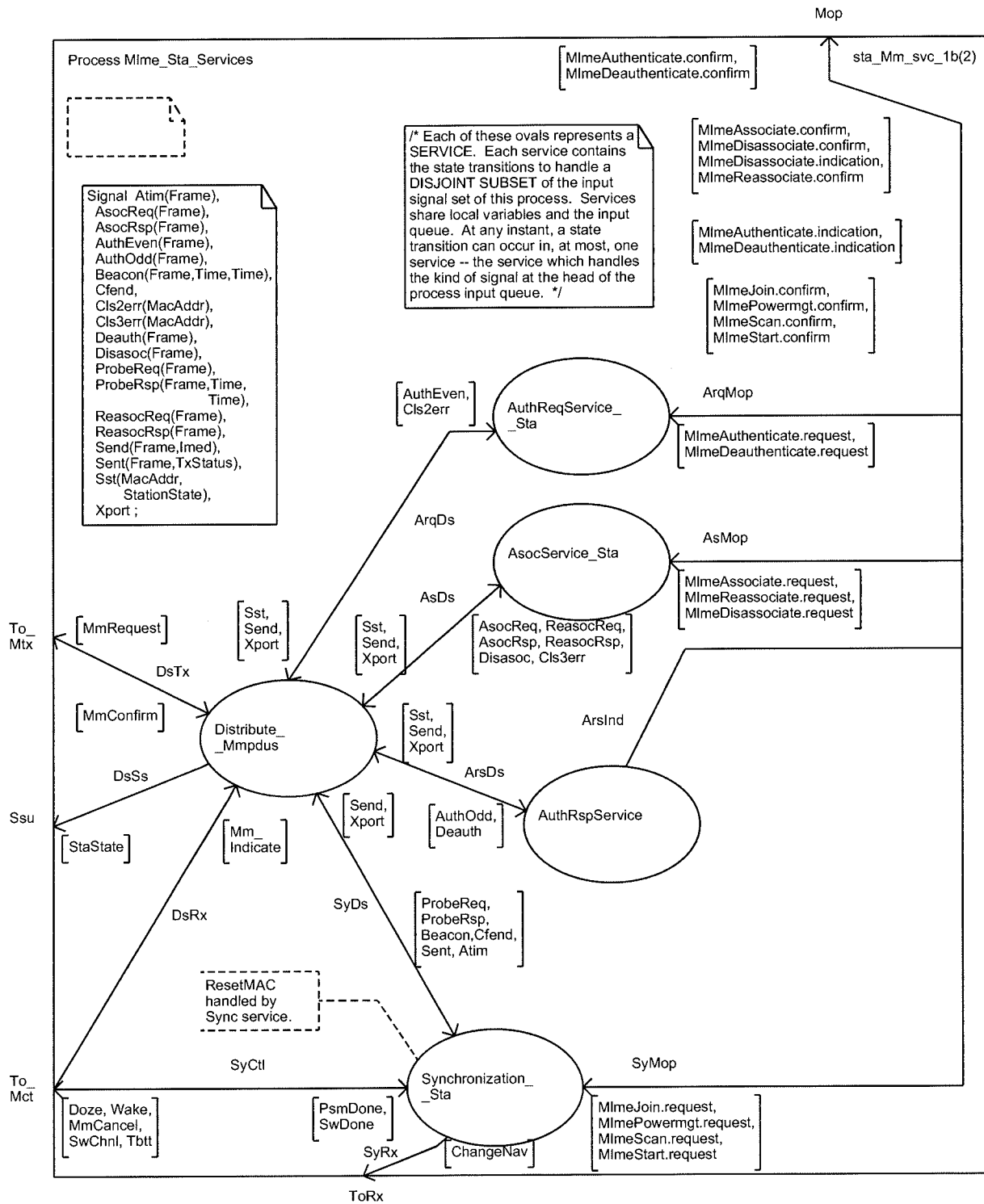












Process Mlme_Sta_Services

sta_Mm_svc_1.1a(2)

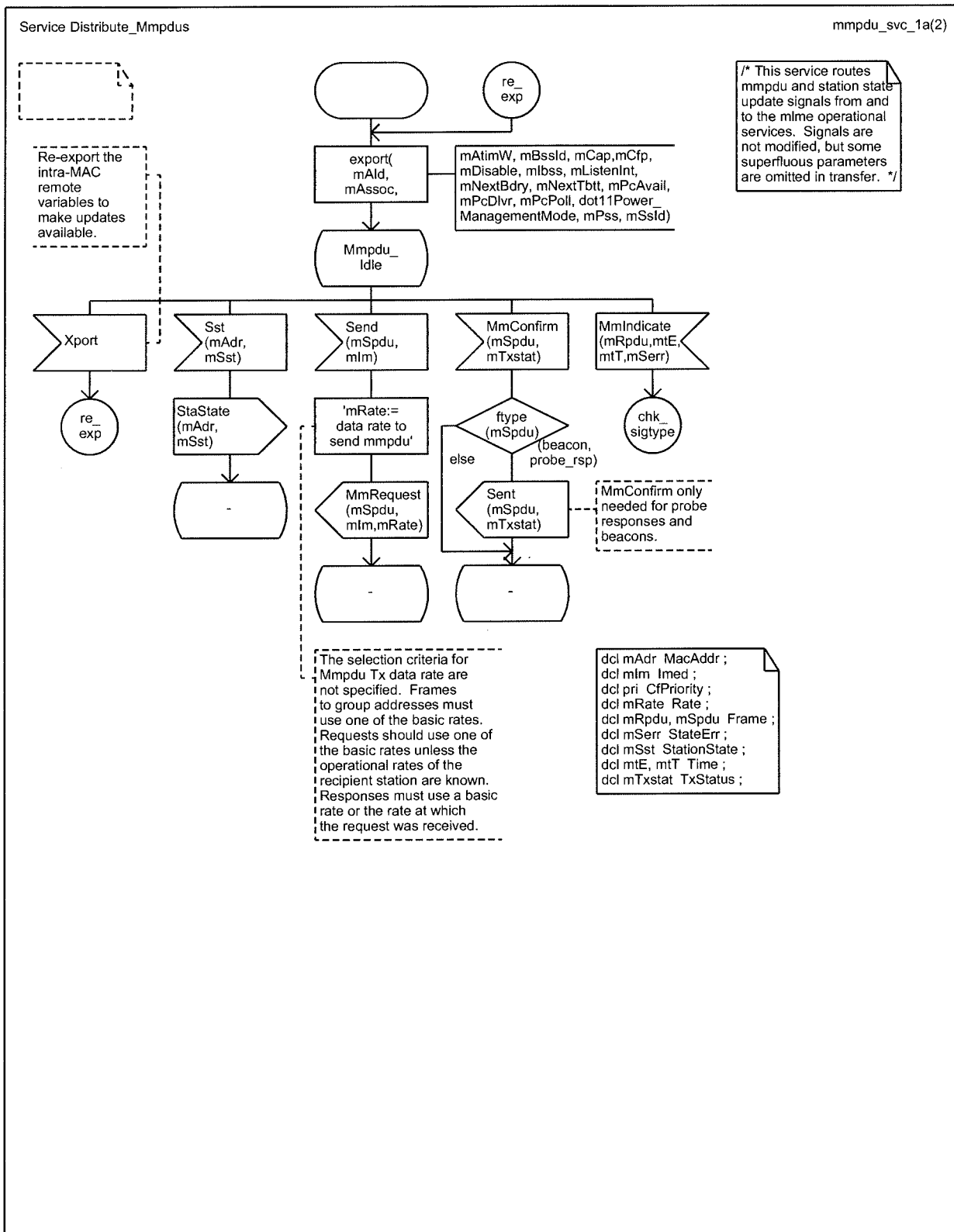


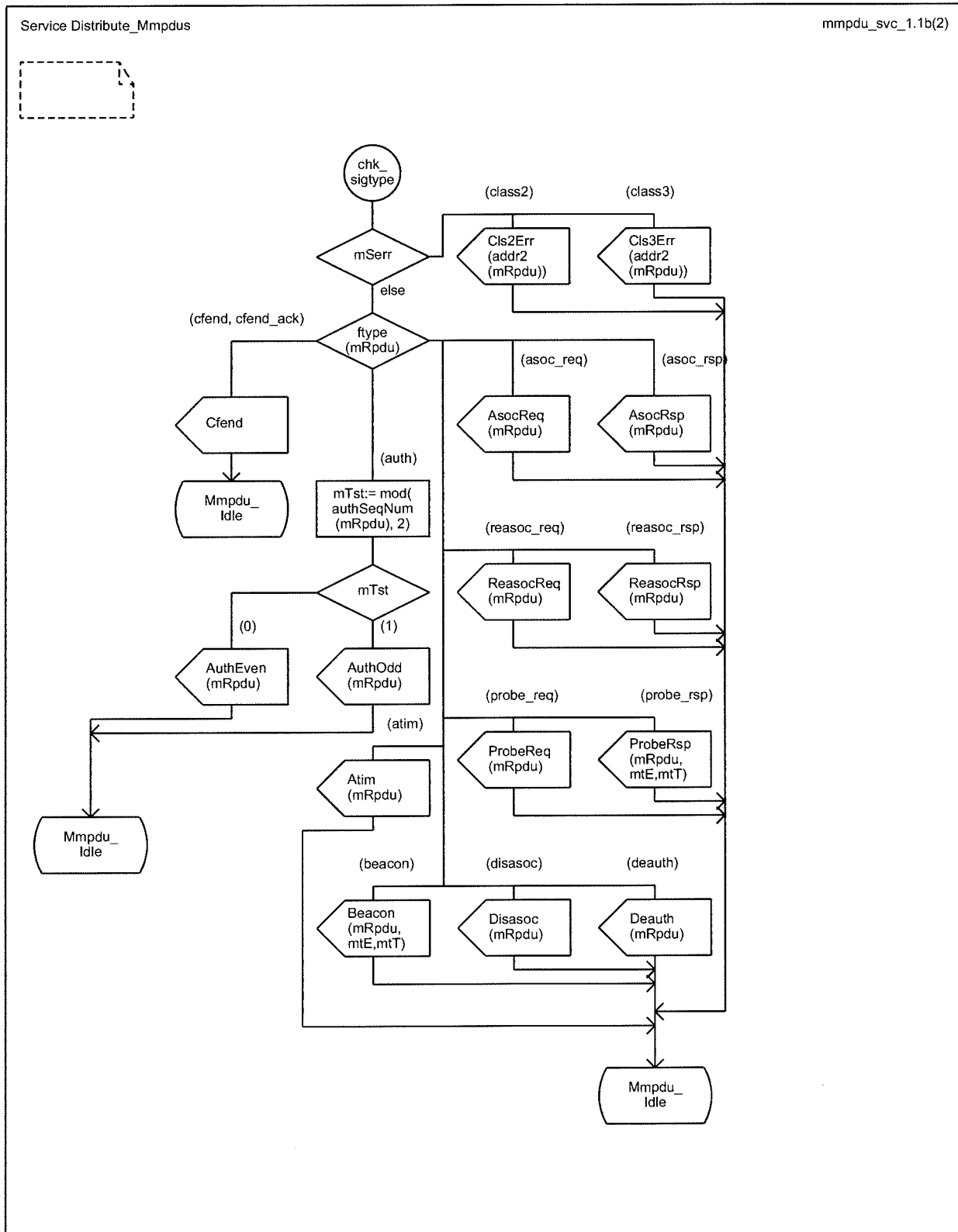
Timer Tasoc,
Tauth, Tchal,
Tbcn, Tatim ;

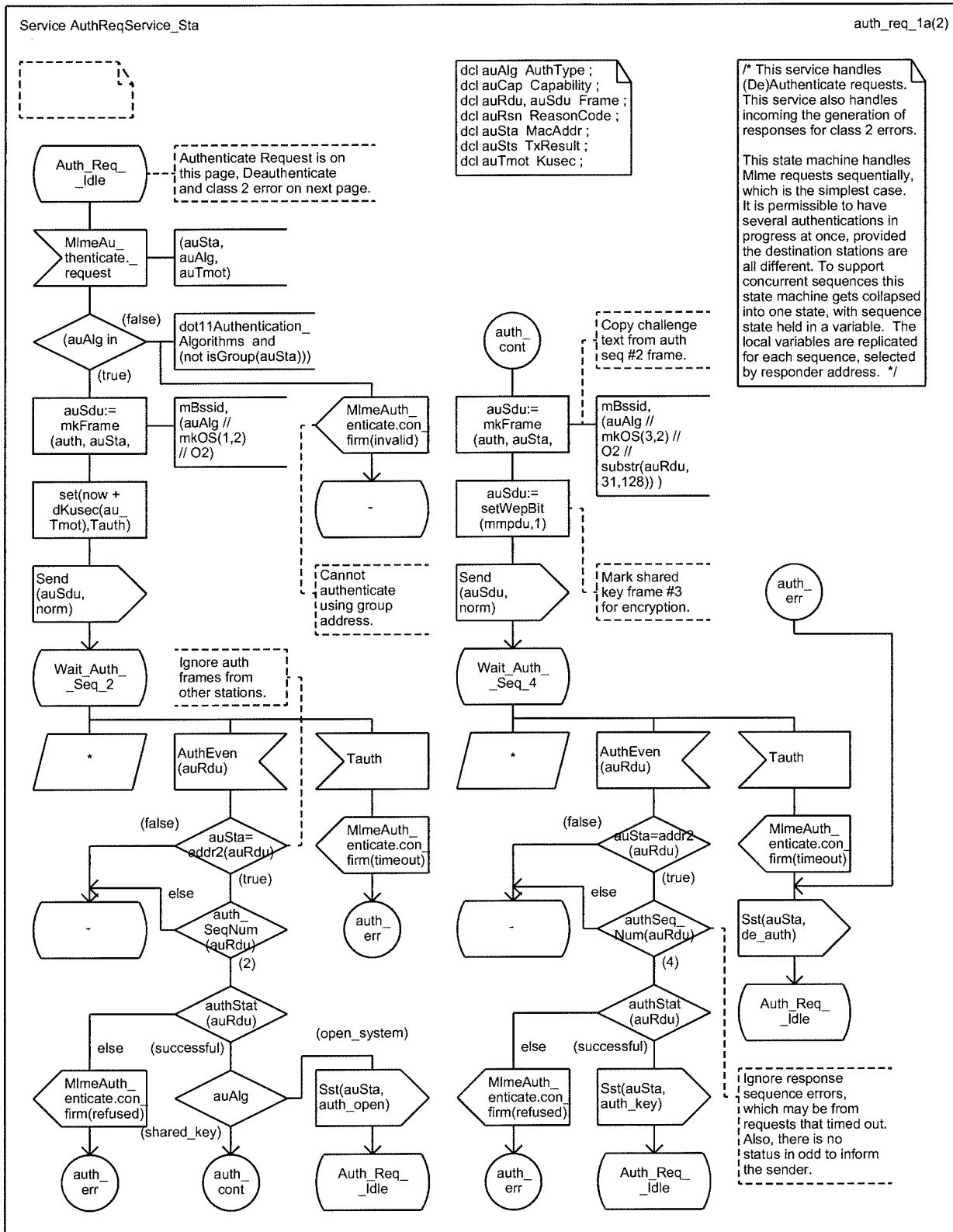
```

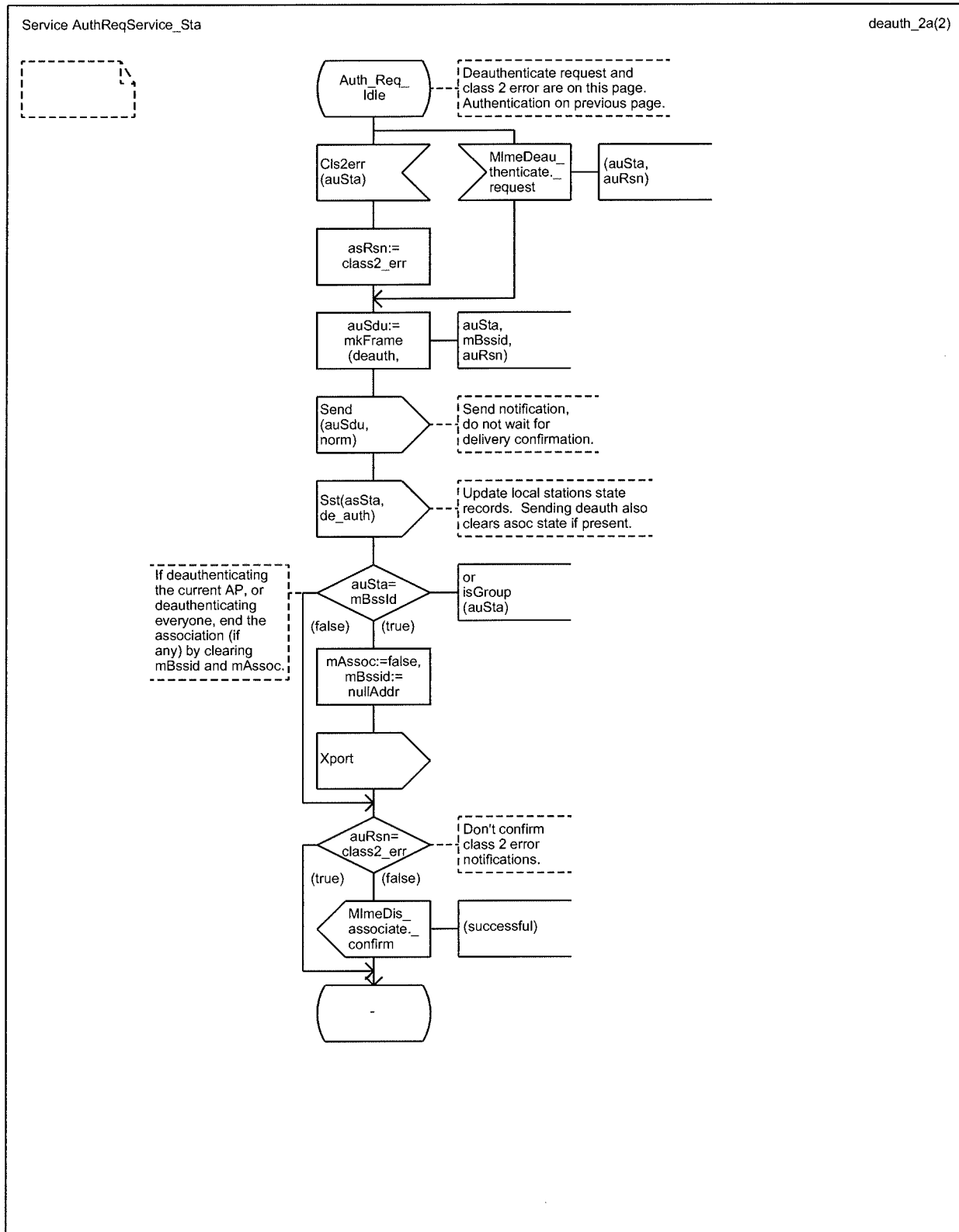
/* Intra-MAC remote variables */
dcl exported
dot11PowerManagementMode PwrSave:= sta_active,
dot11DesiredSsid Octetstring,
dot11DesiredBssType,
dot11OperationalRateSet Ratestring:= mkOS(2,1),
dot11BeaconPeriod TU,
dot11DtimPeriod Integer:= 1,
dot11AssociationResponseTimeOut TU,
mAld AsocId:= 0,
mAssoc Boolean:= false,
mAtimW Boolean:= false,
mBrates Ratestring:=mkOS(2,1),
mBssid MacAddr:= nullAddr,
mCap Octetstring:= O2,
mCfp Boolean:= false,
mDisable Boolean:= true,
mDtimCount Integer:= 1,
mlbss Boolean:= false,
mNextBdry Time:= 0,
mNextTbtt Time:= 0,
mPcAvail Boolean:= false,
mPcPoll Boolean:= false,
mPdly Usec:= 0,
mPss PsState:= awake,
mSsid Octetstring:= null;

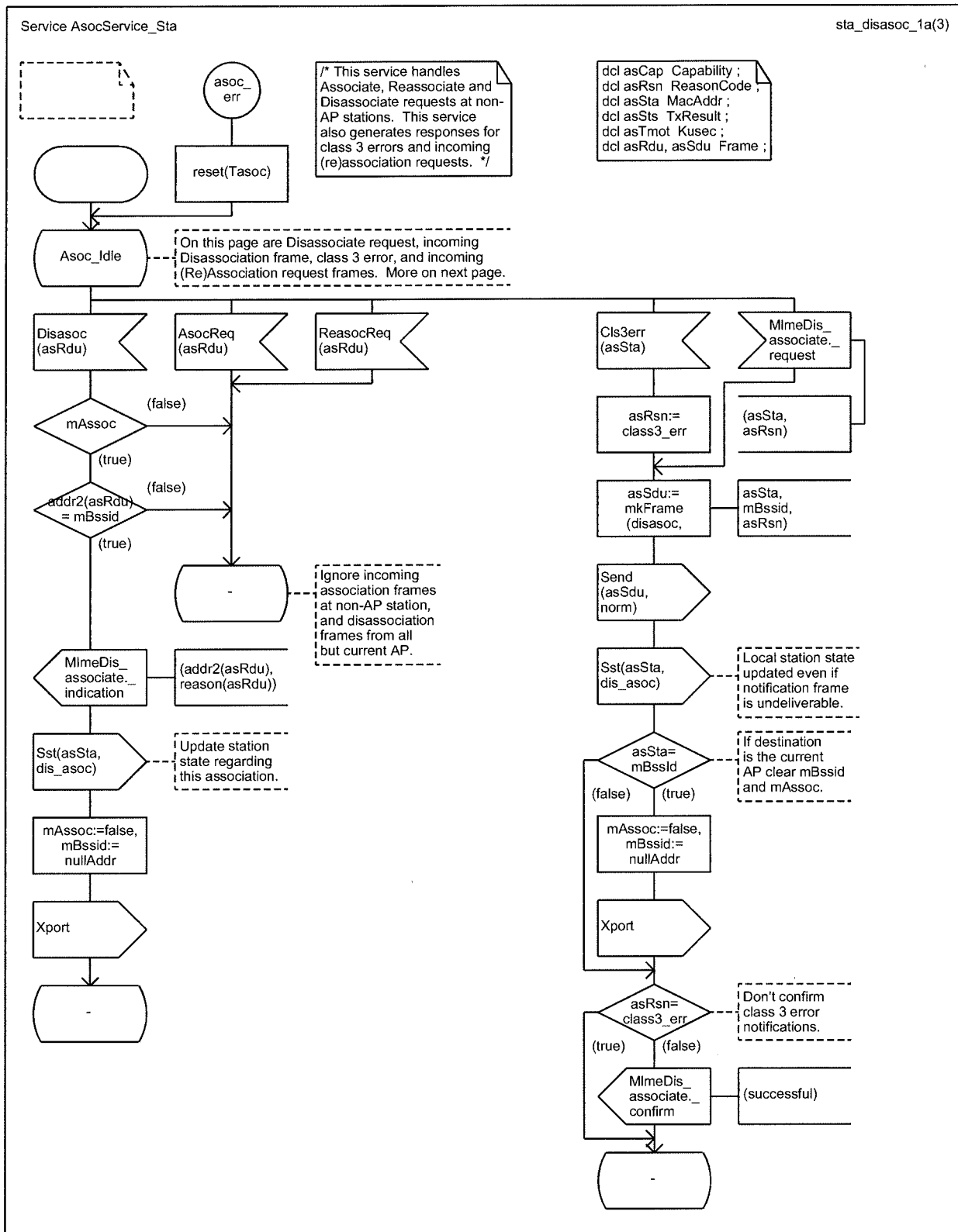
```

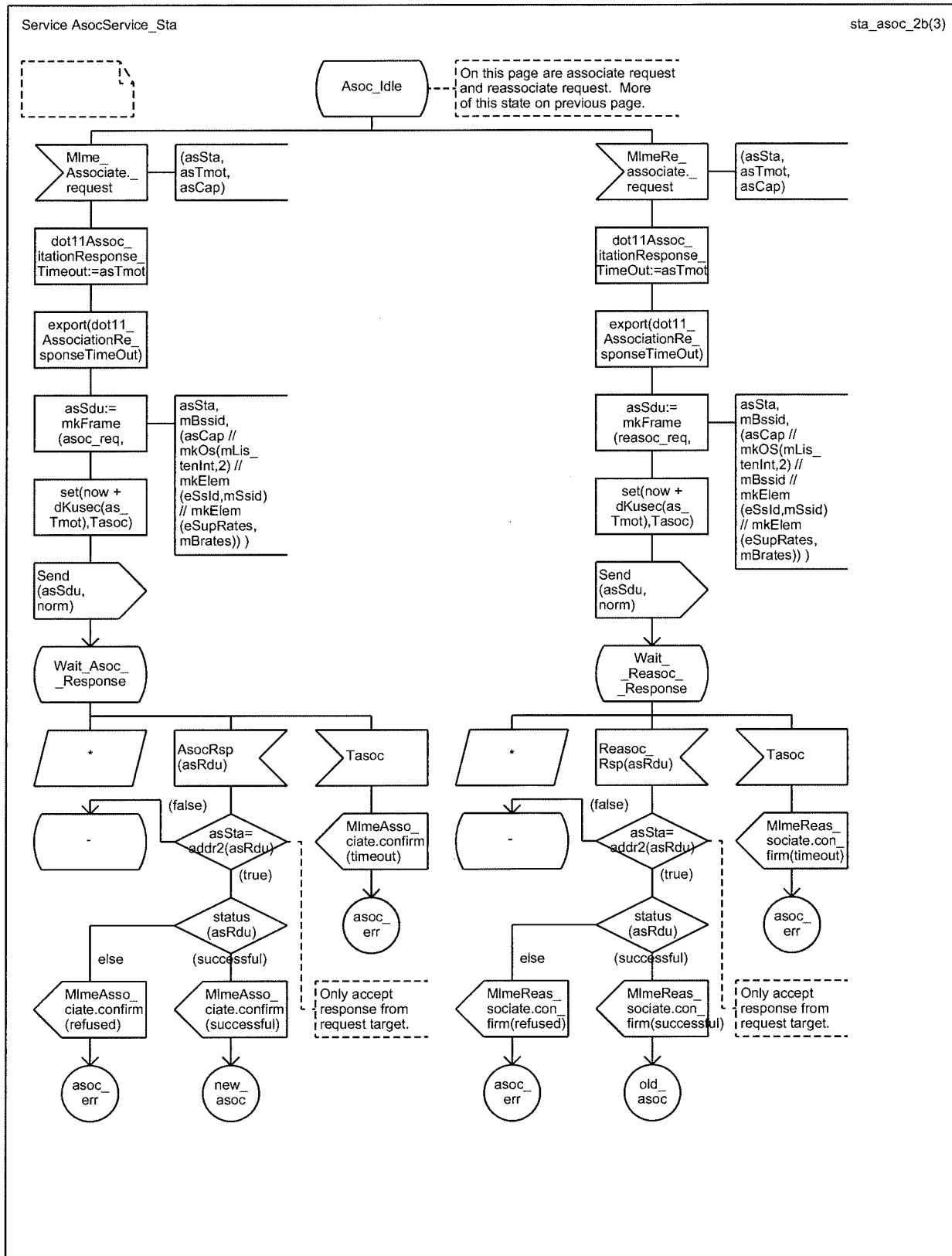


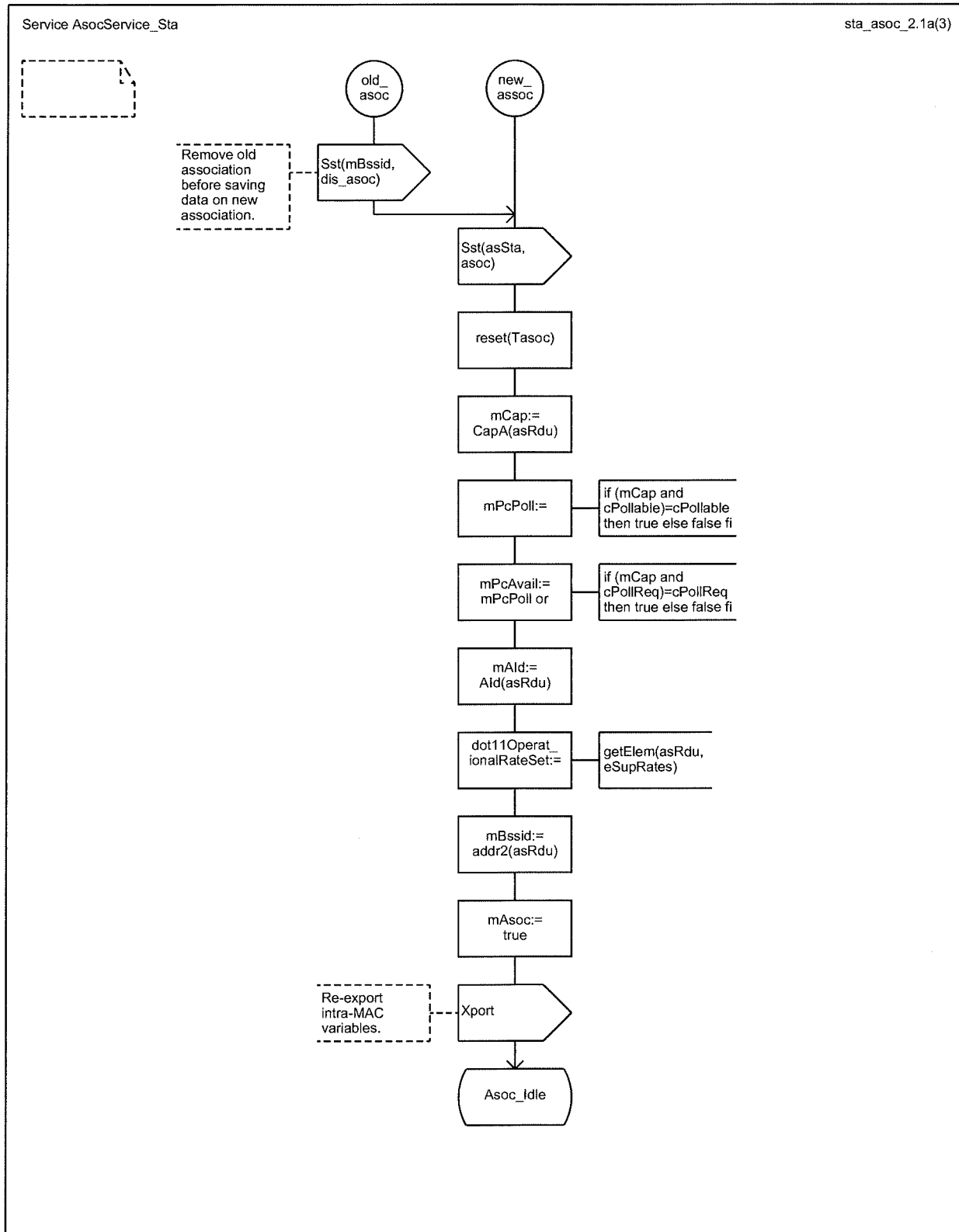


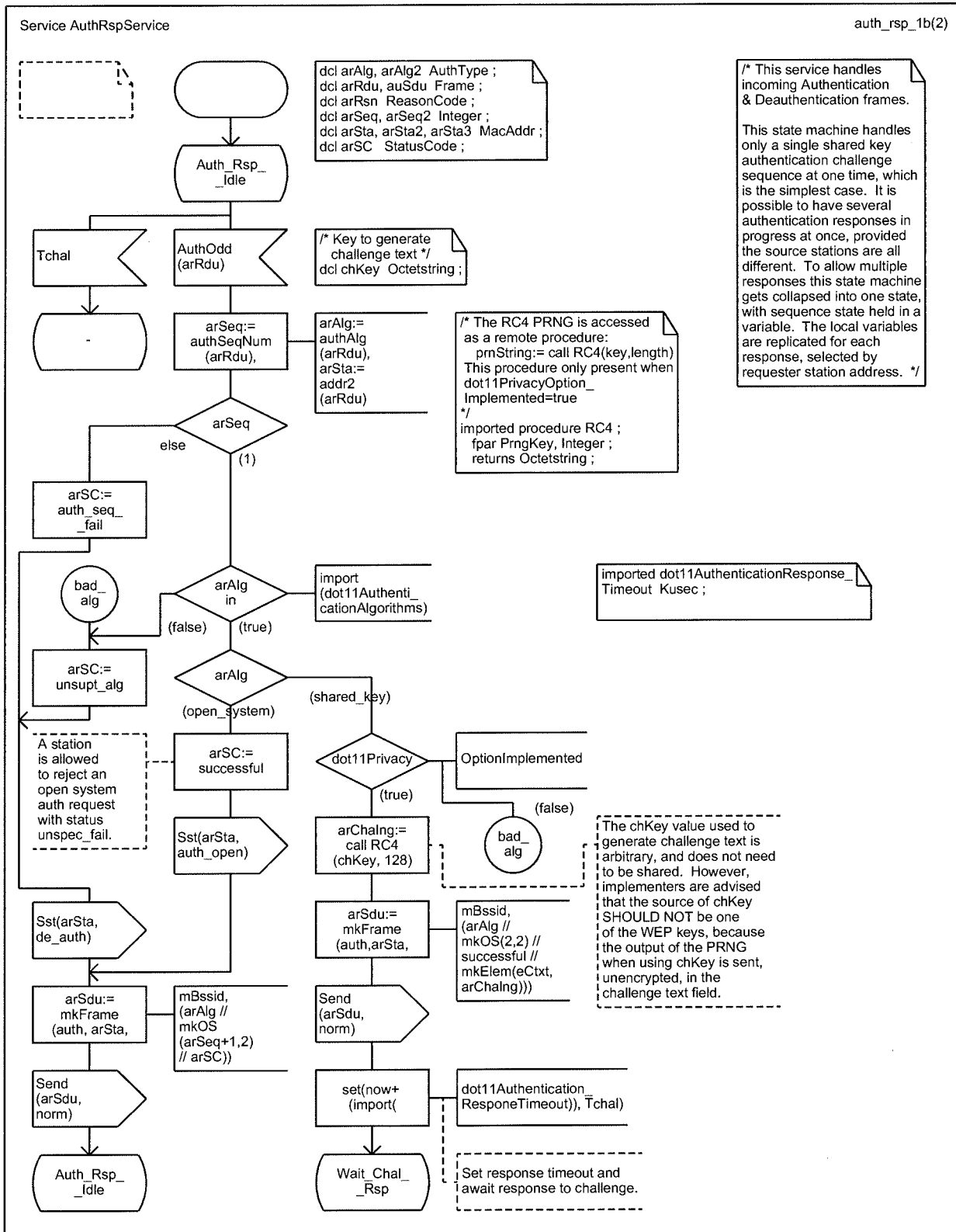


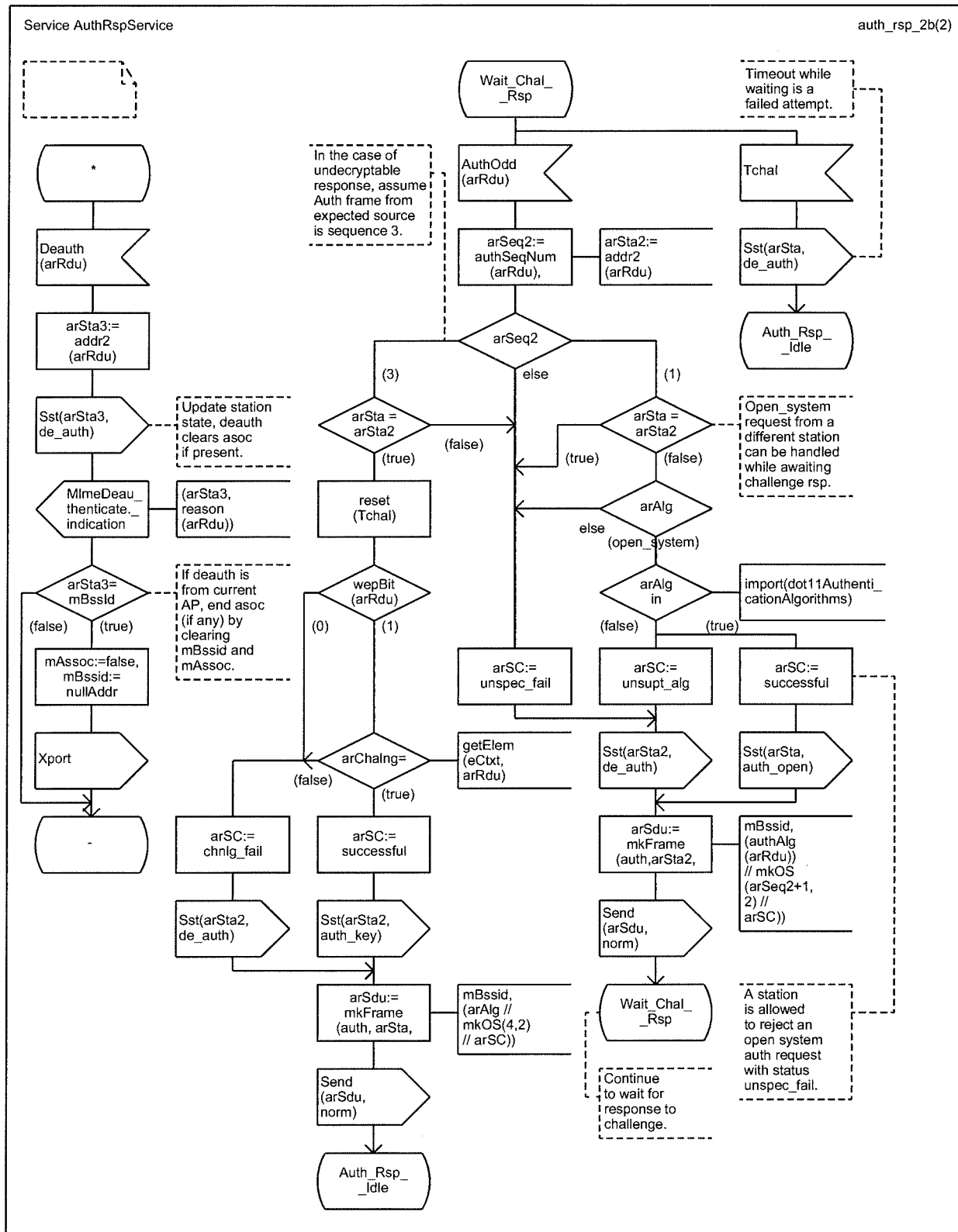


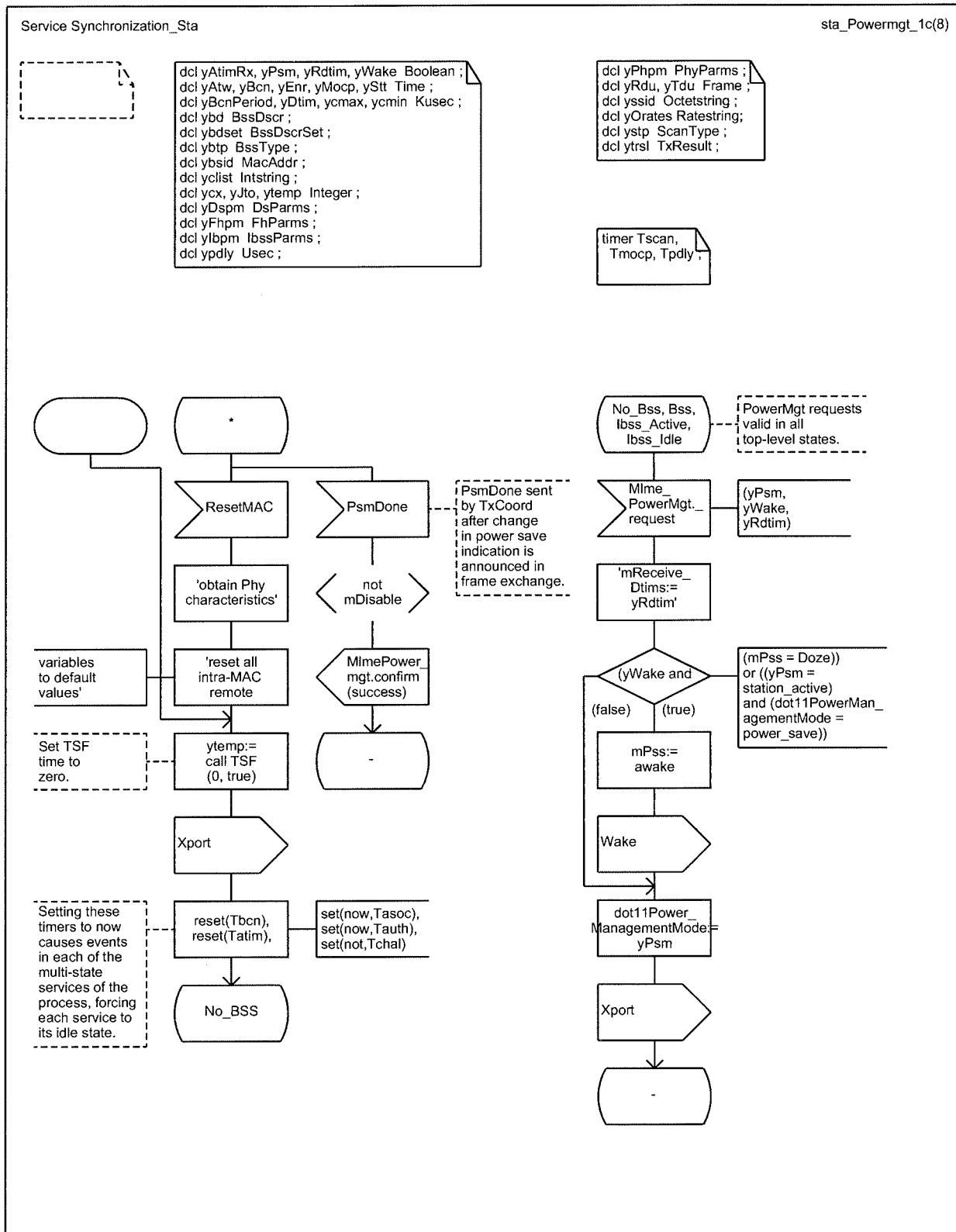


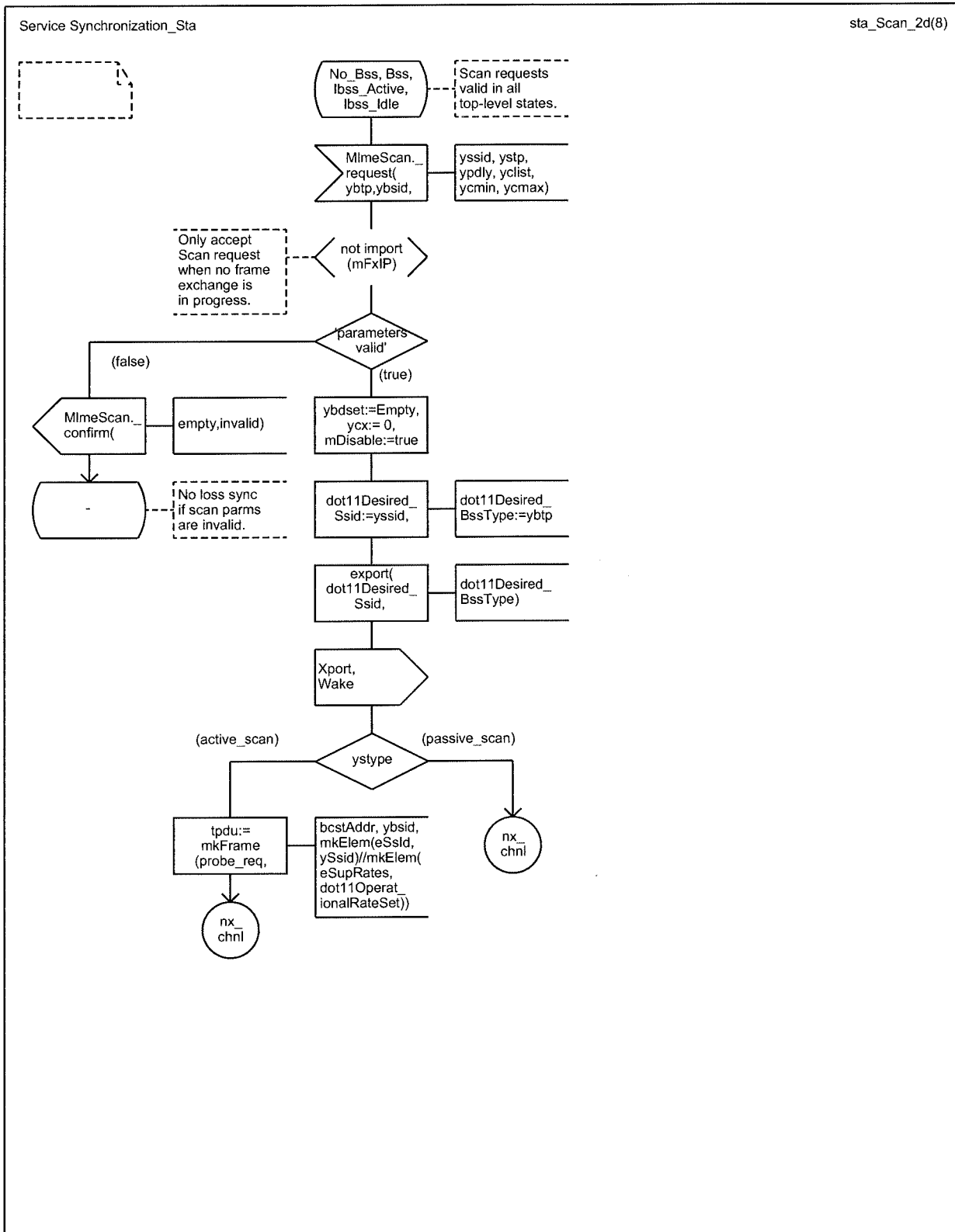


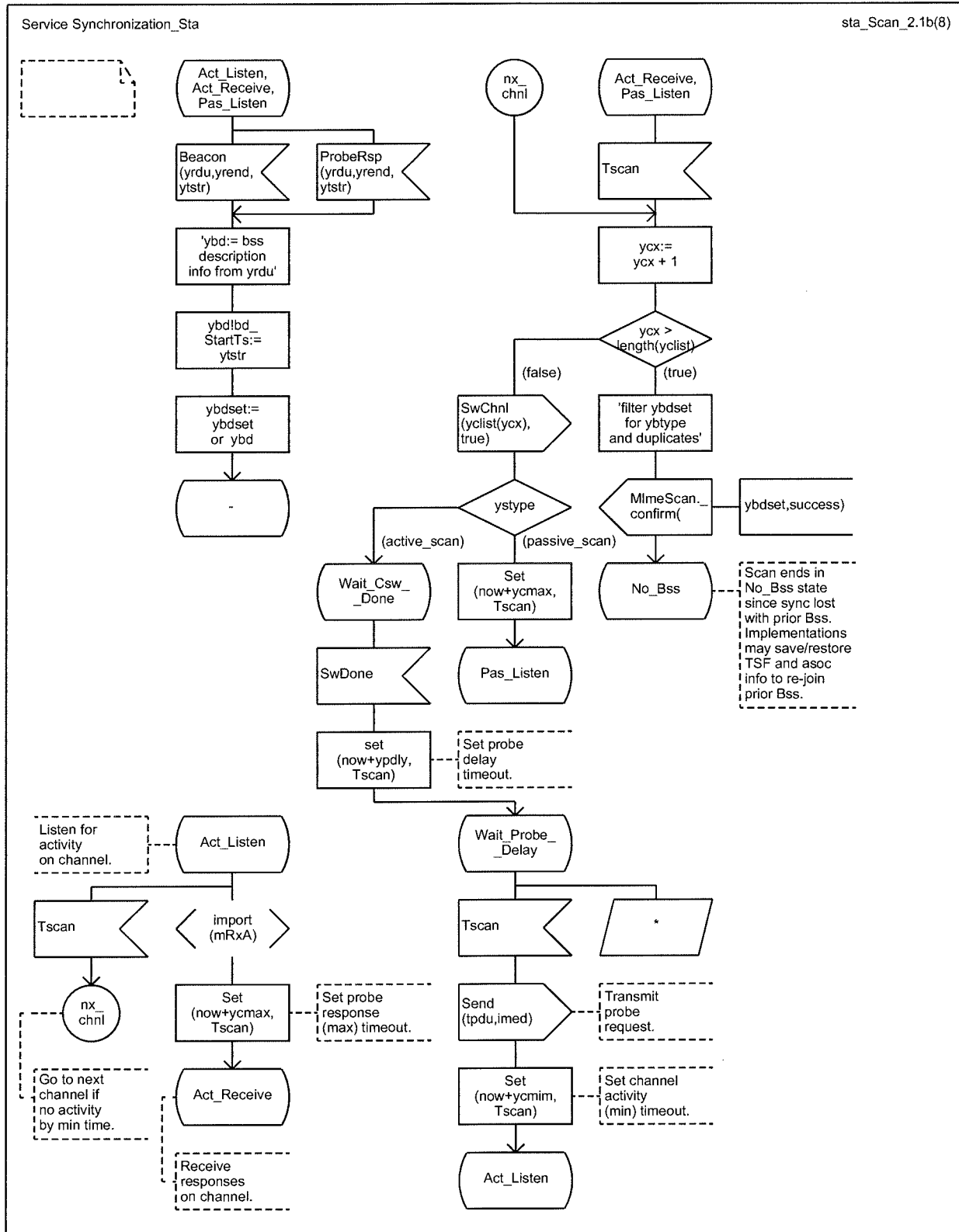


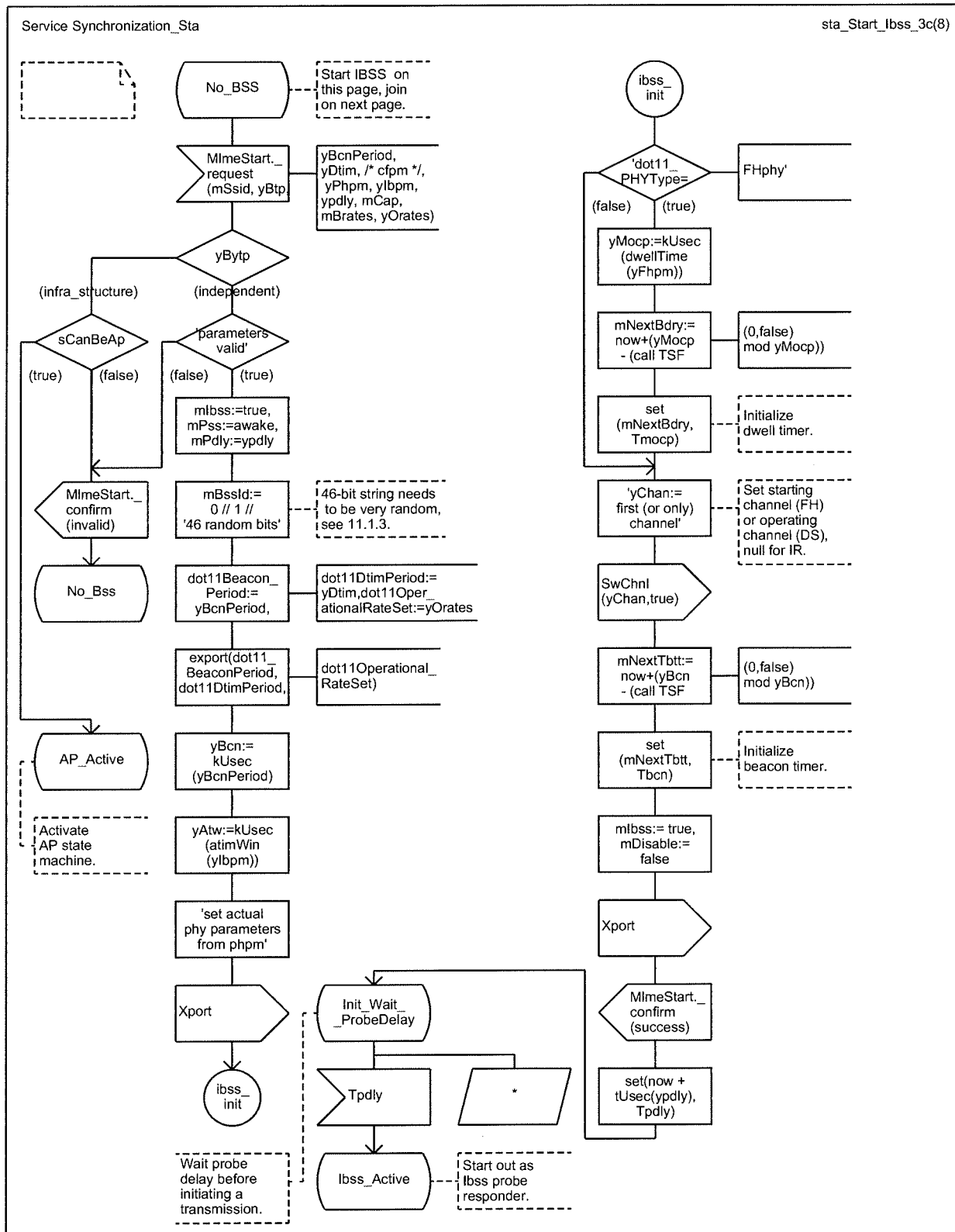


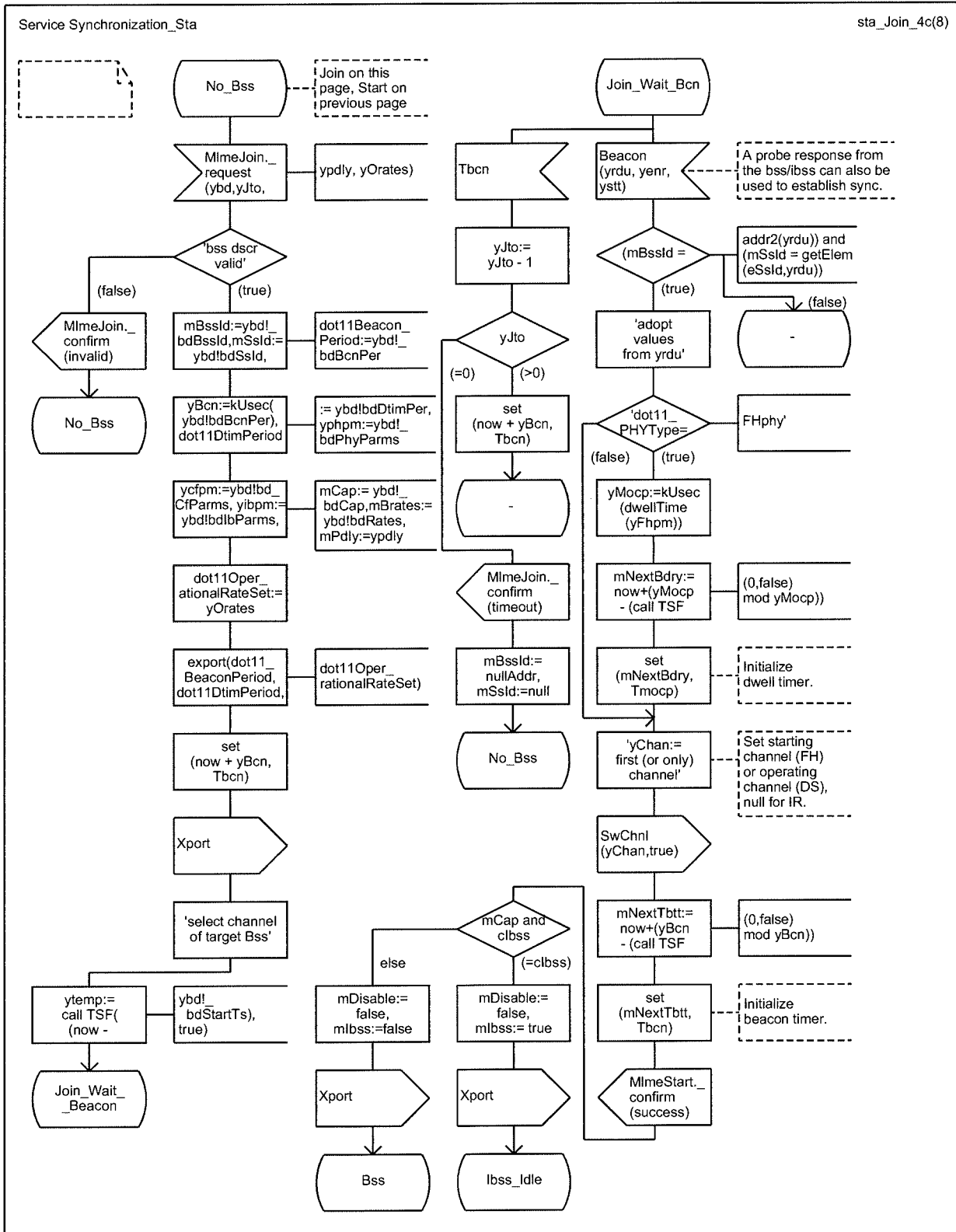


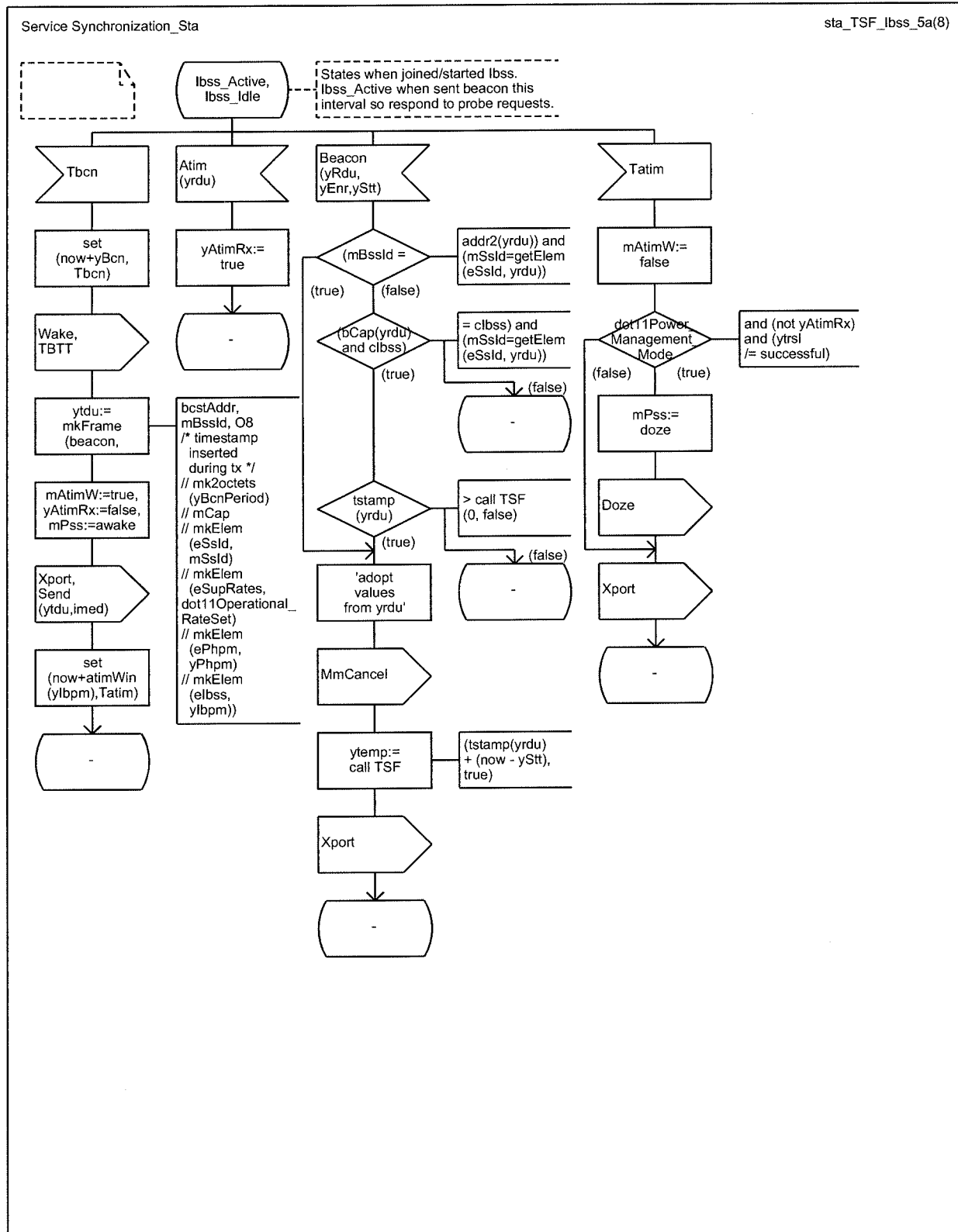


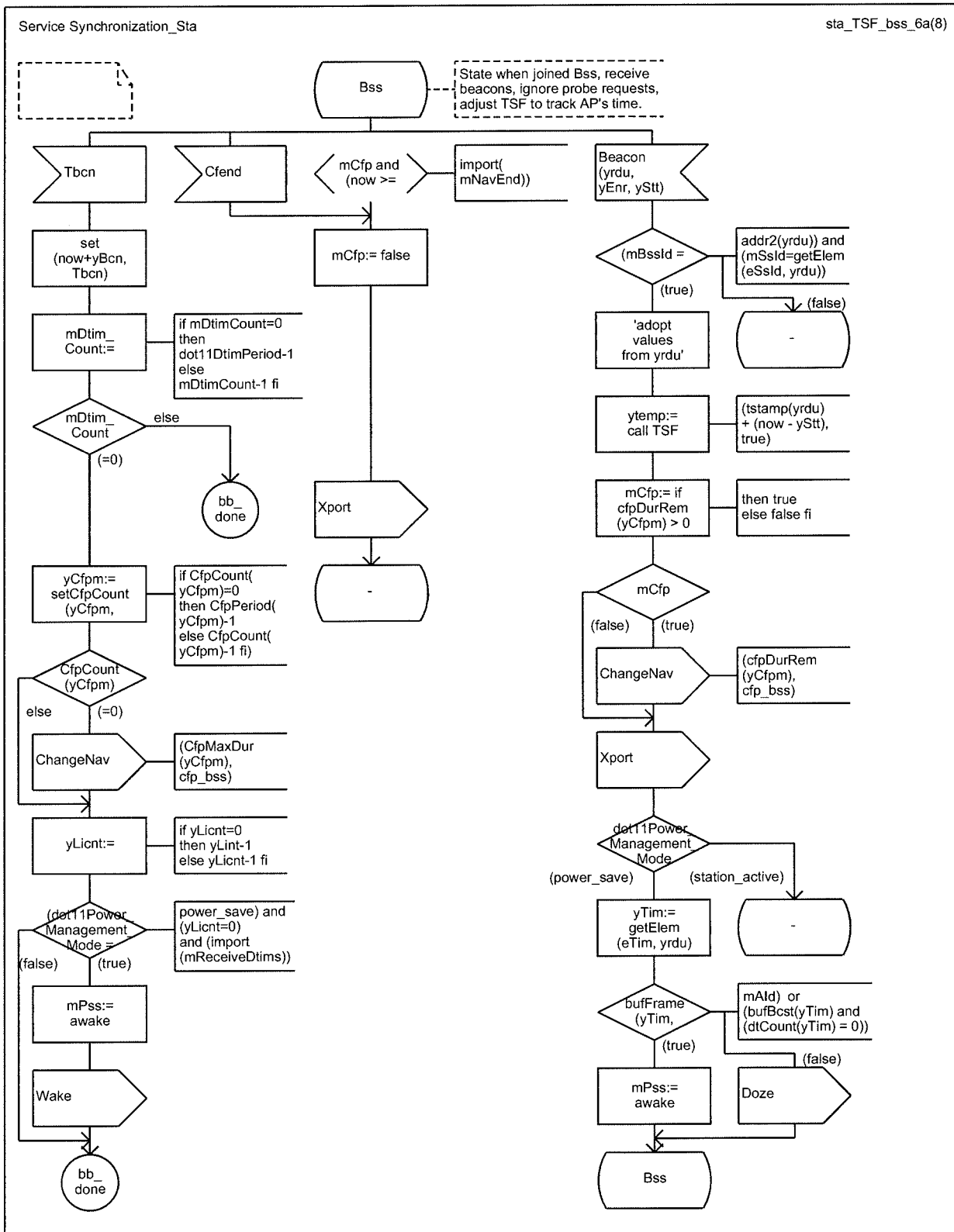


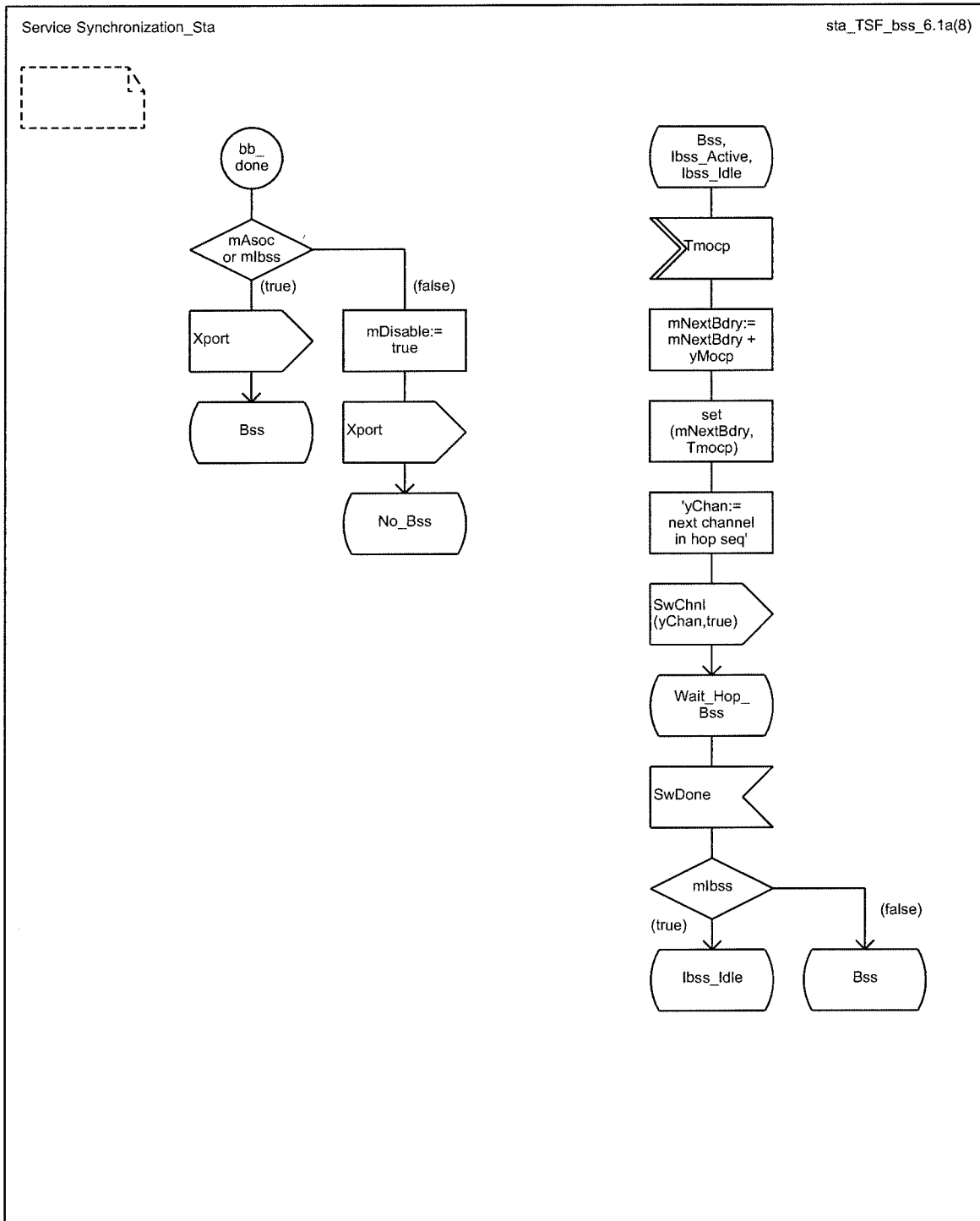


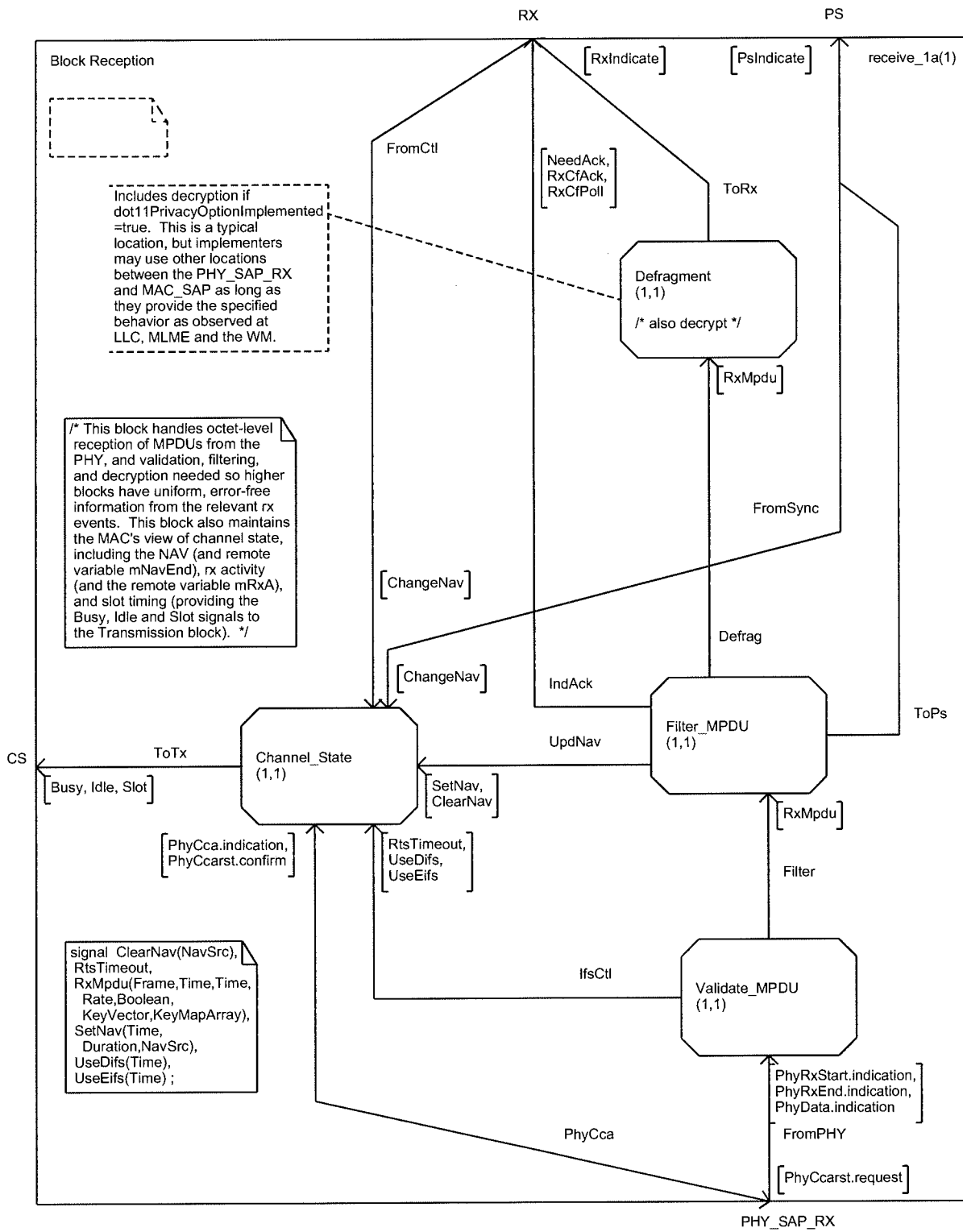


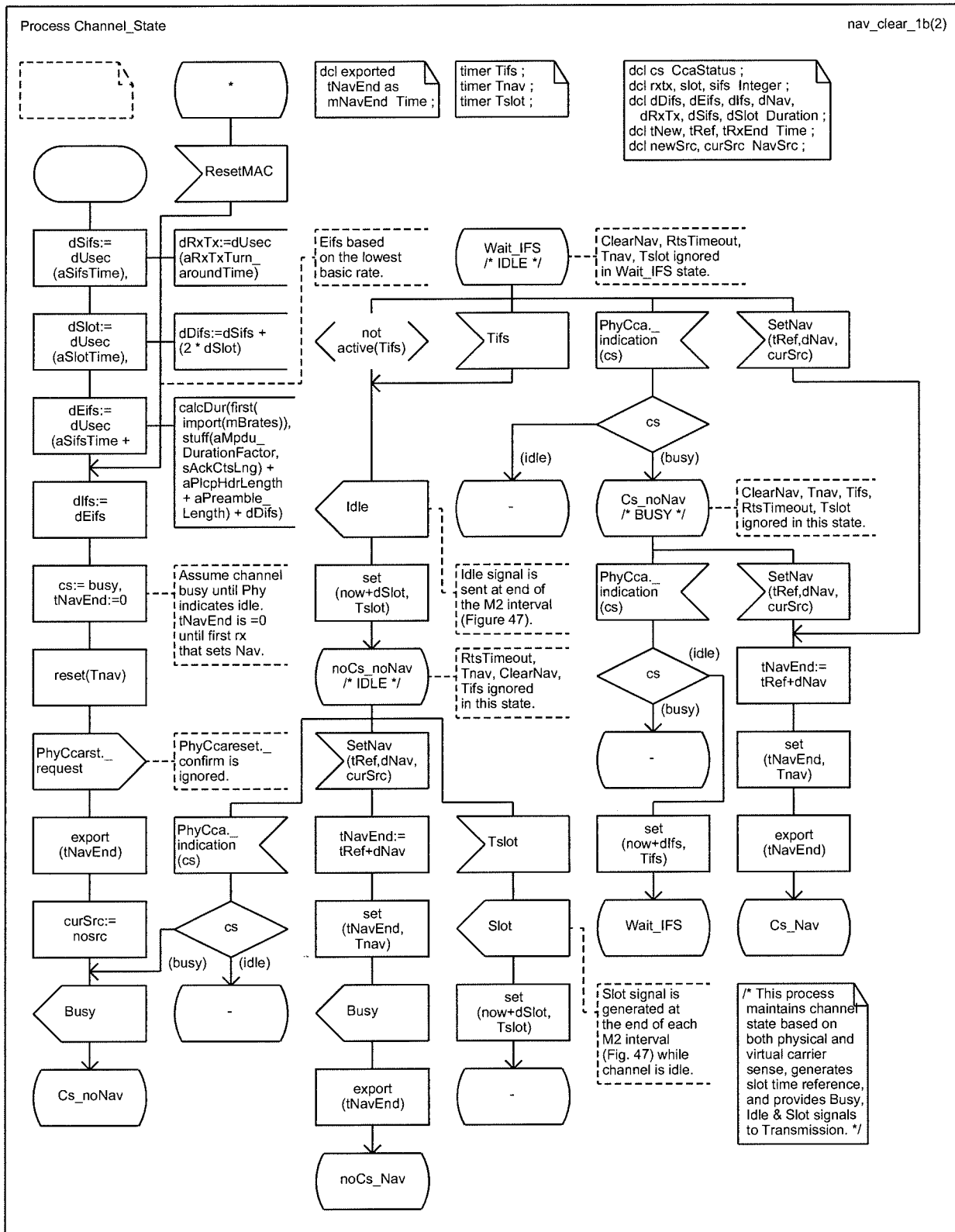


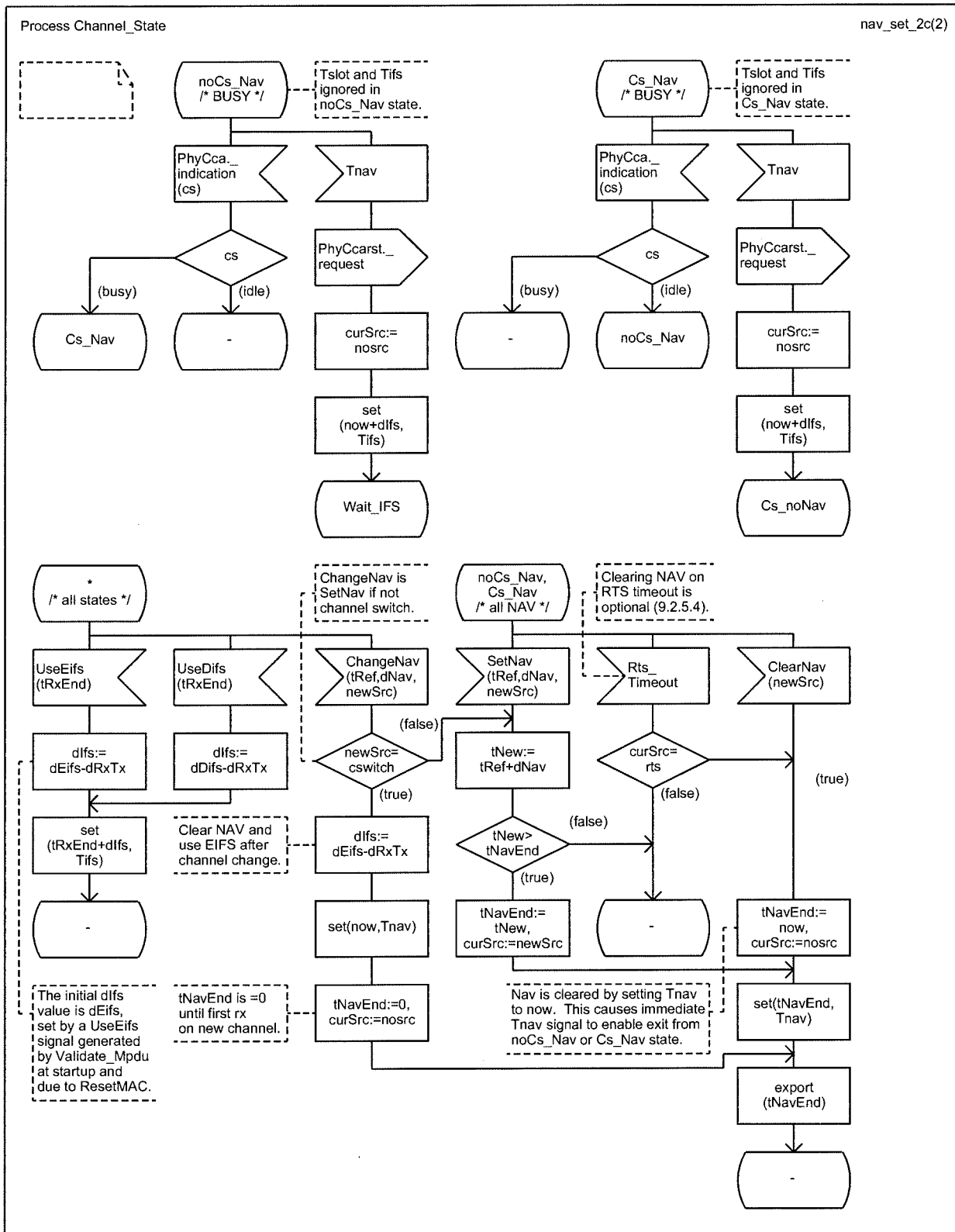


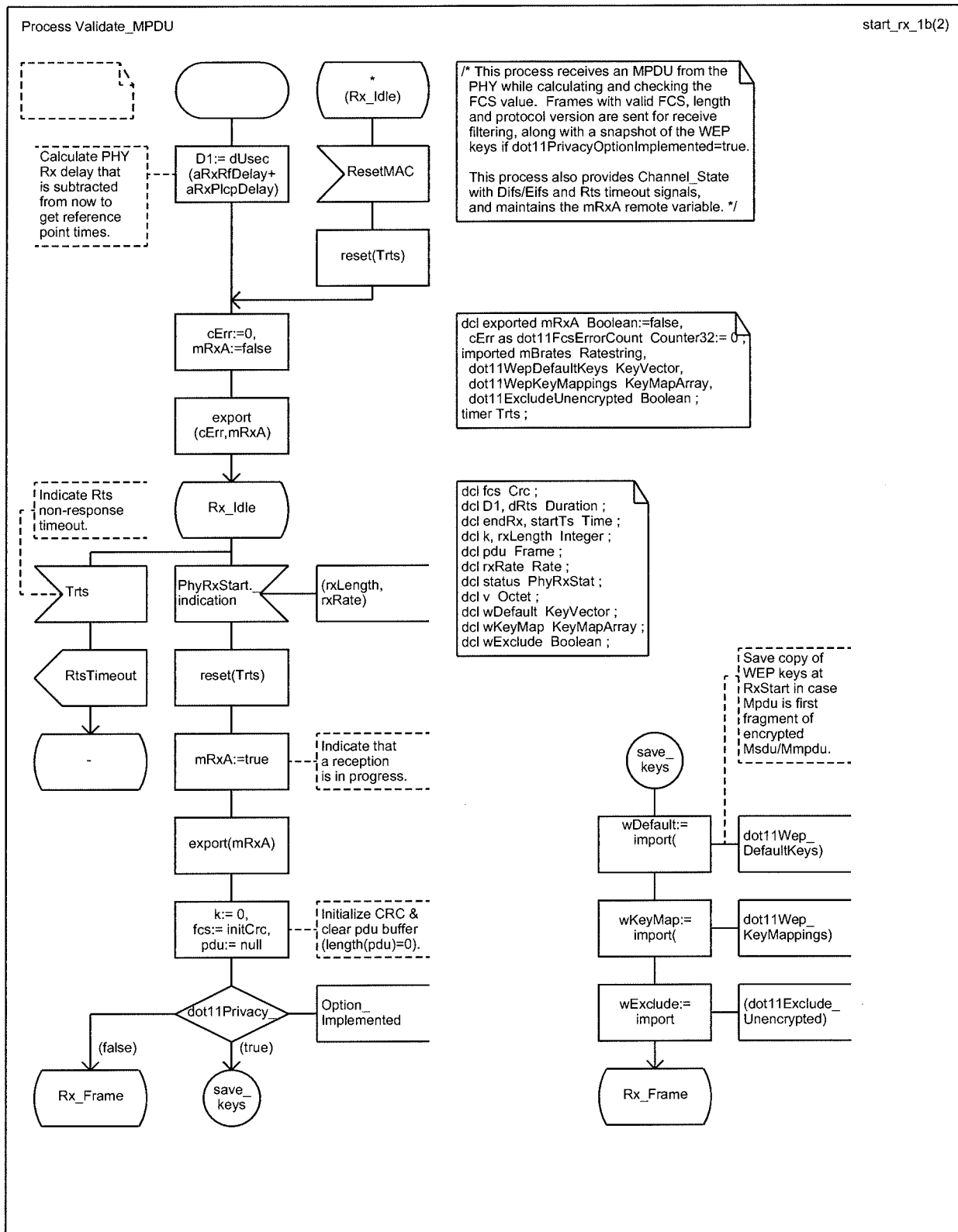


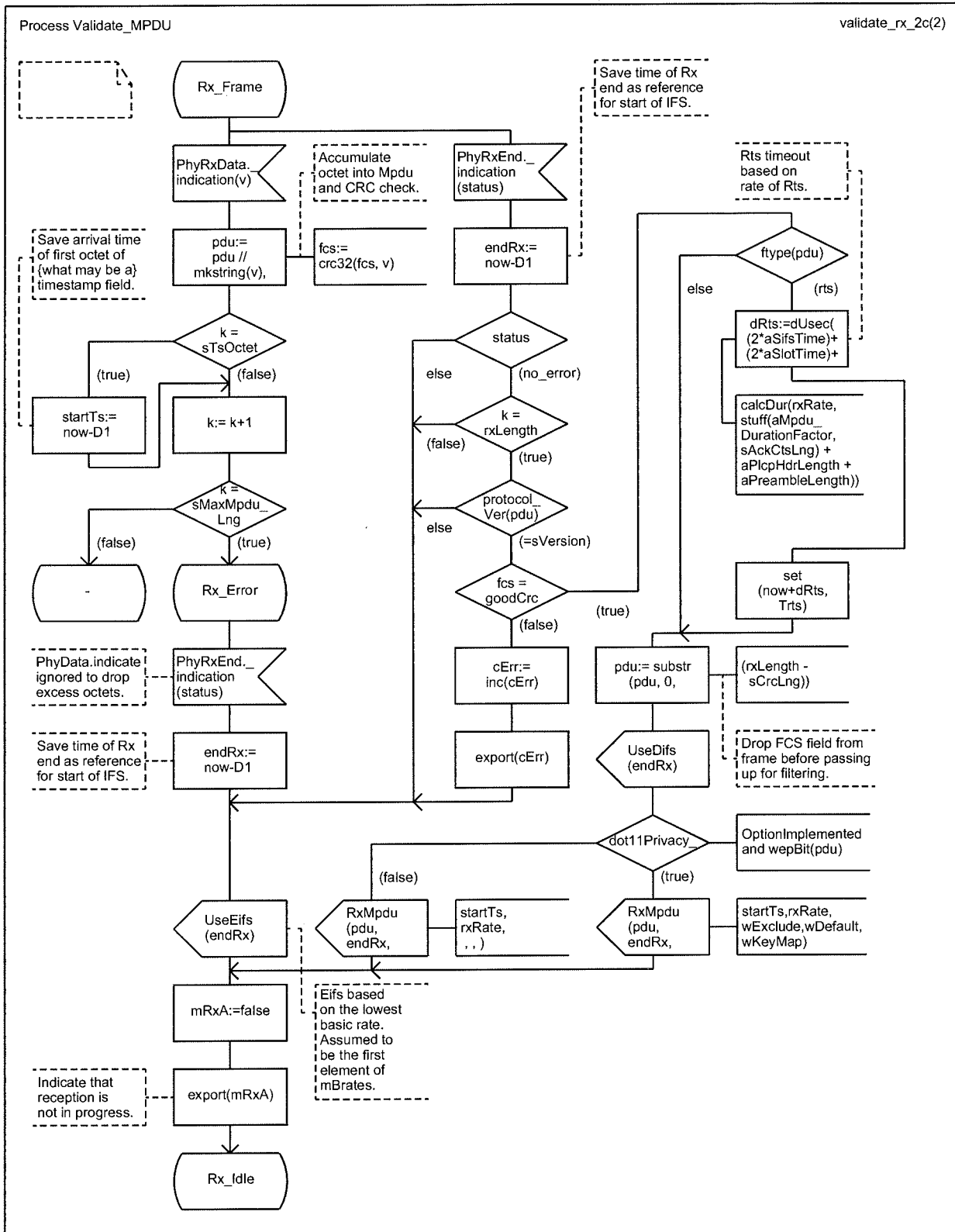


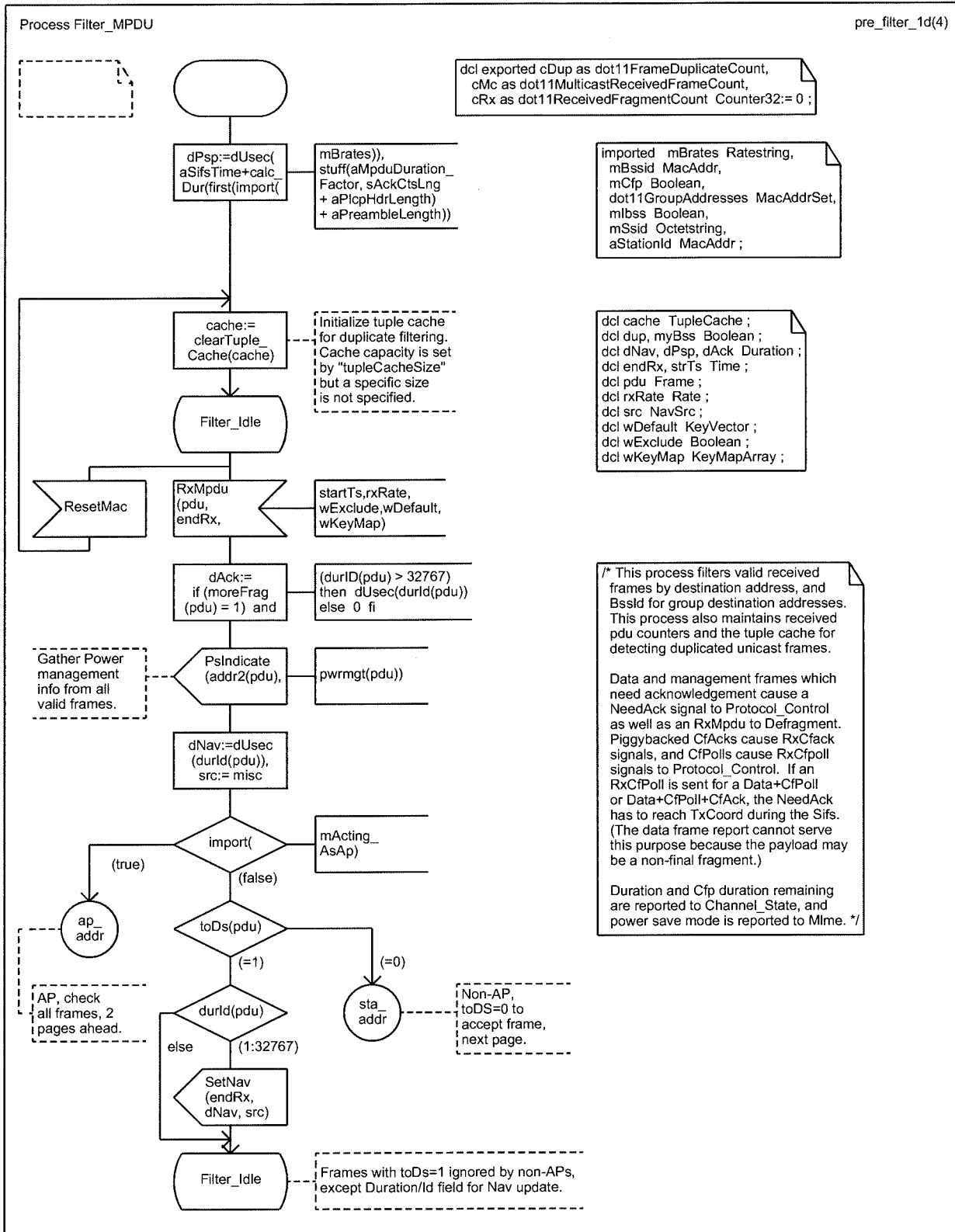


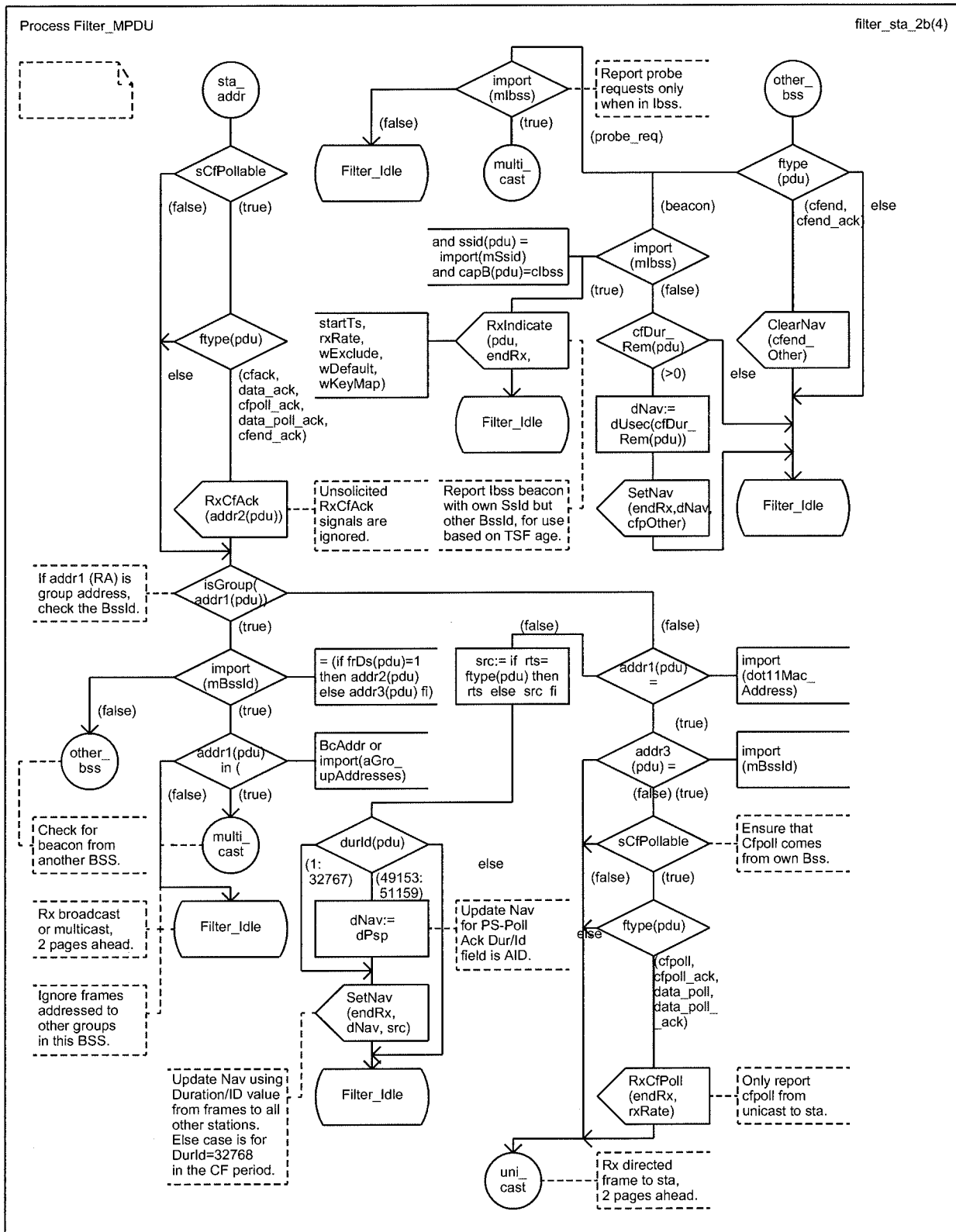


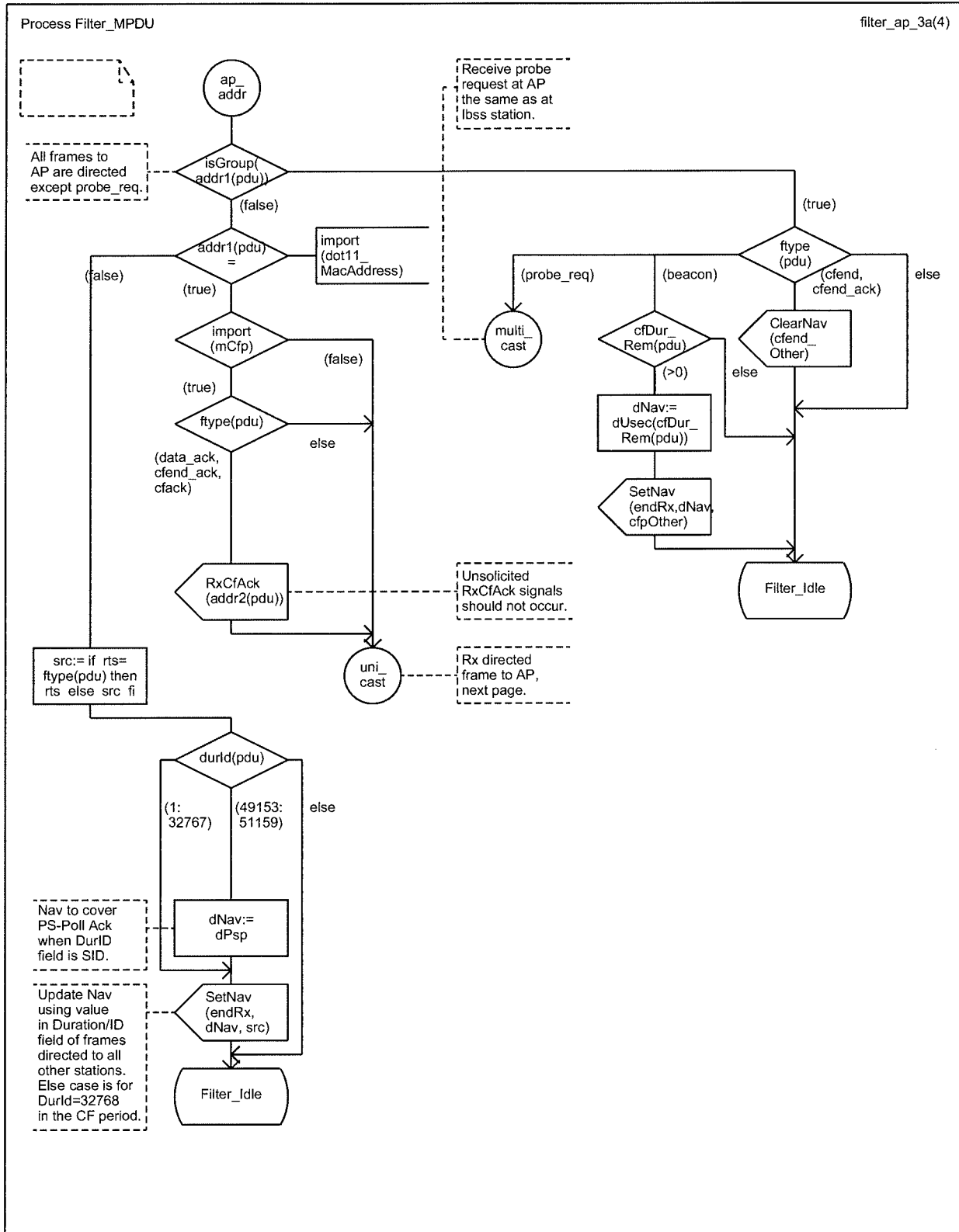


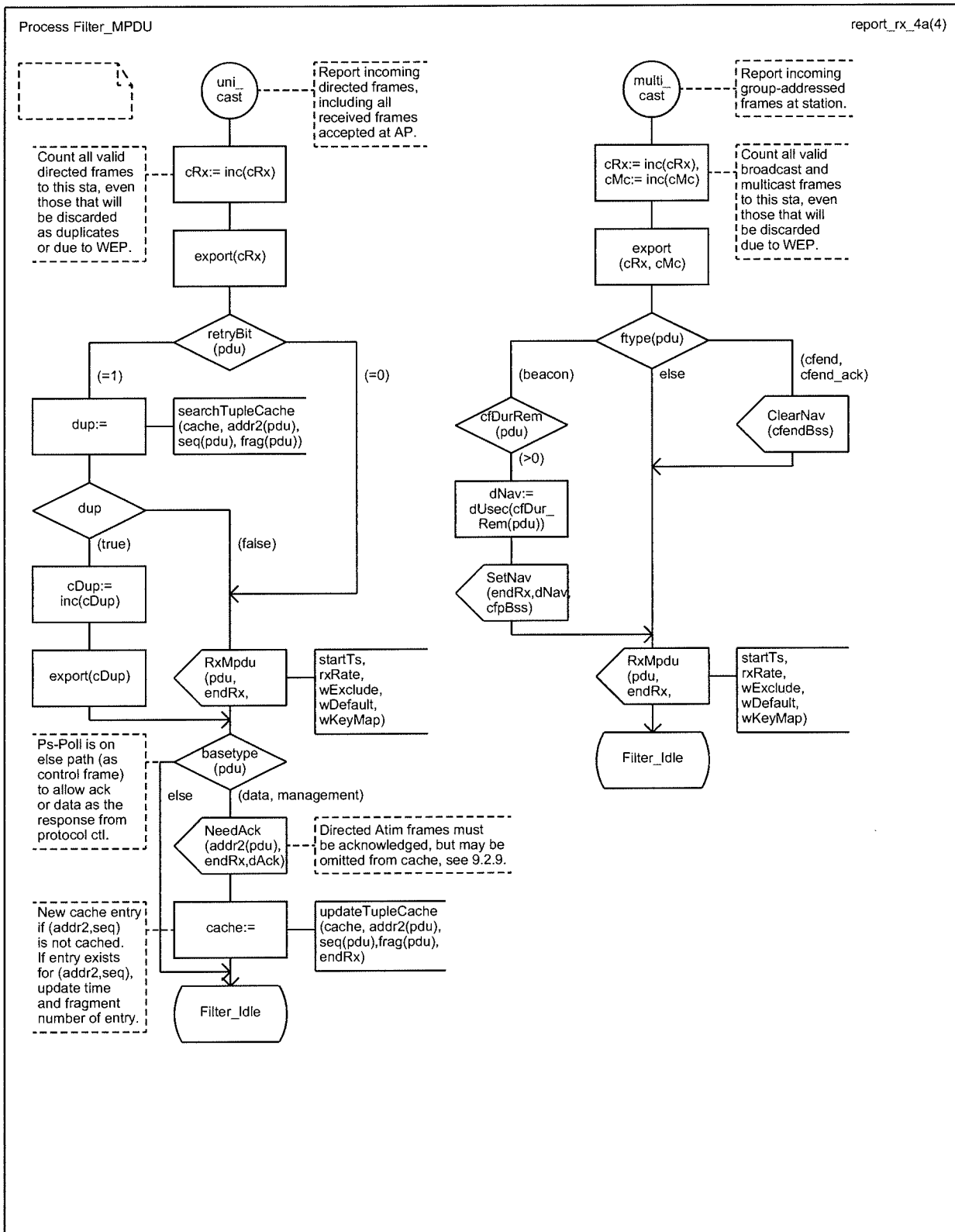


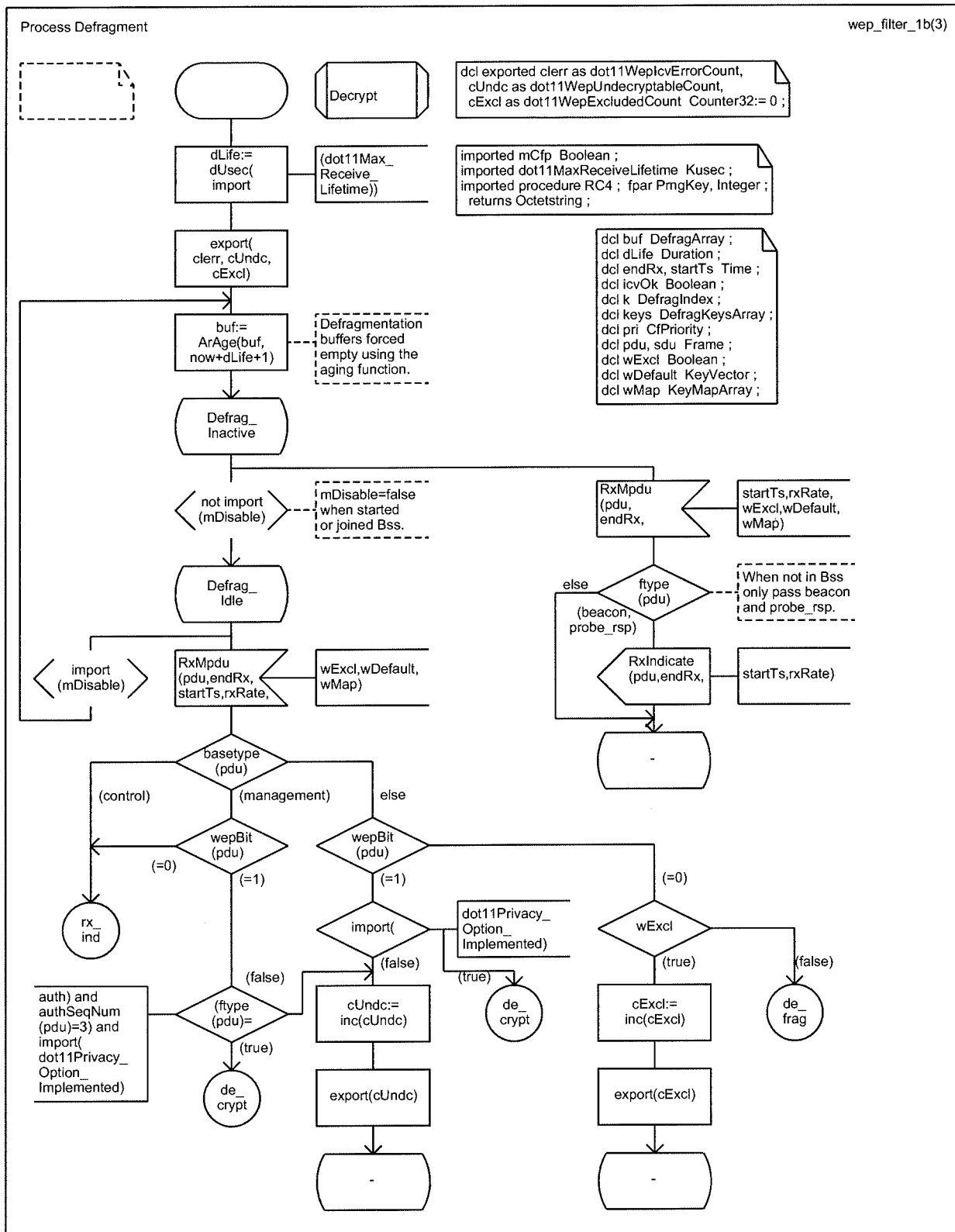


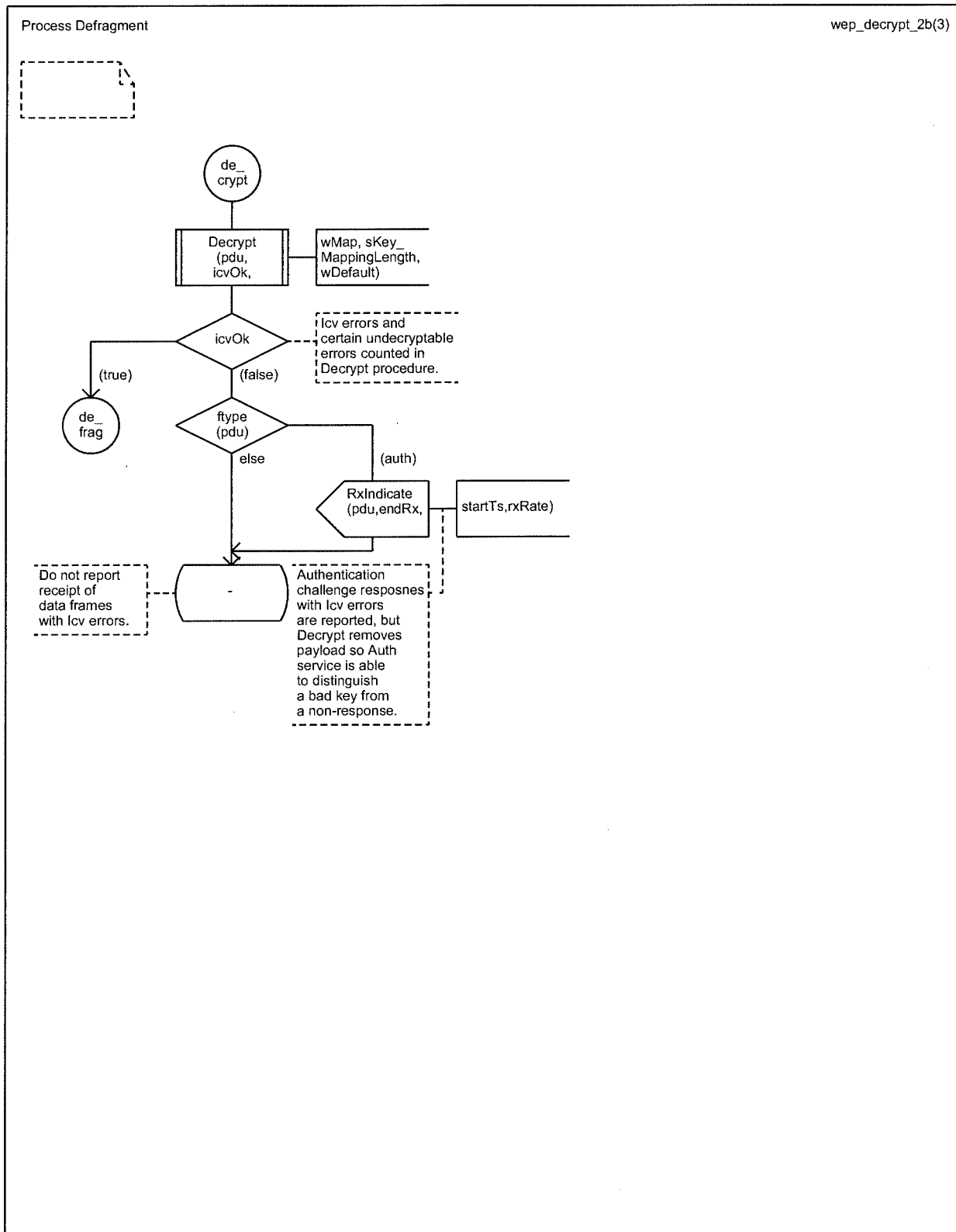


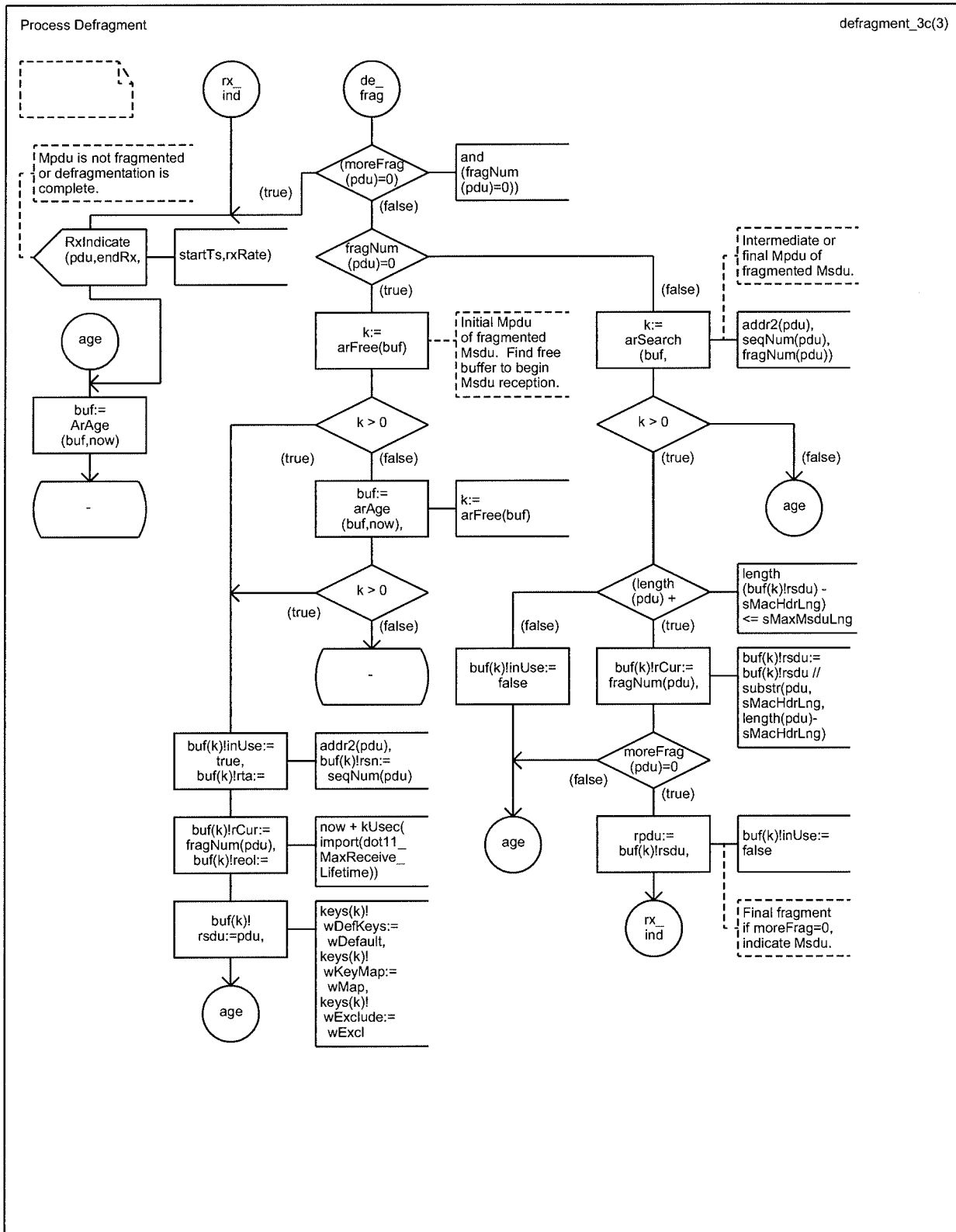


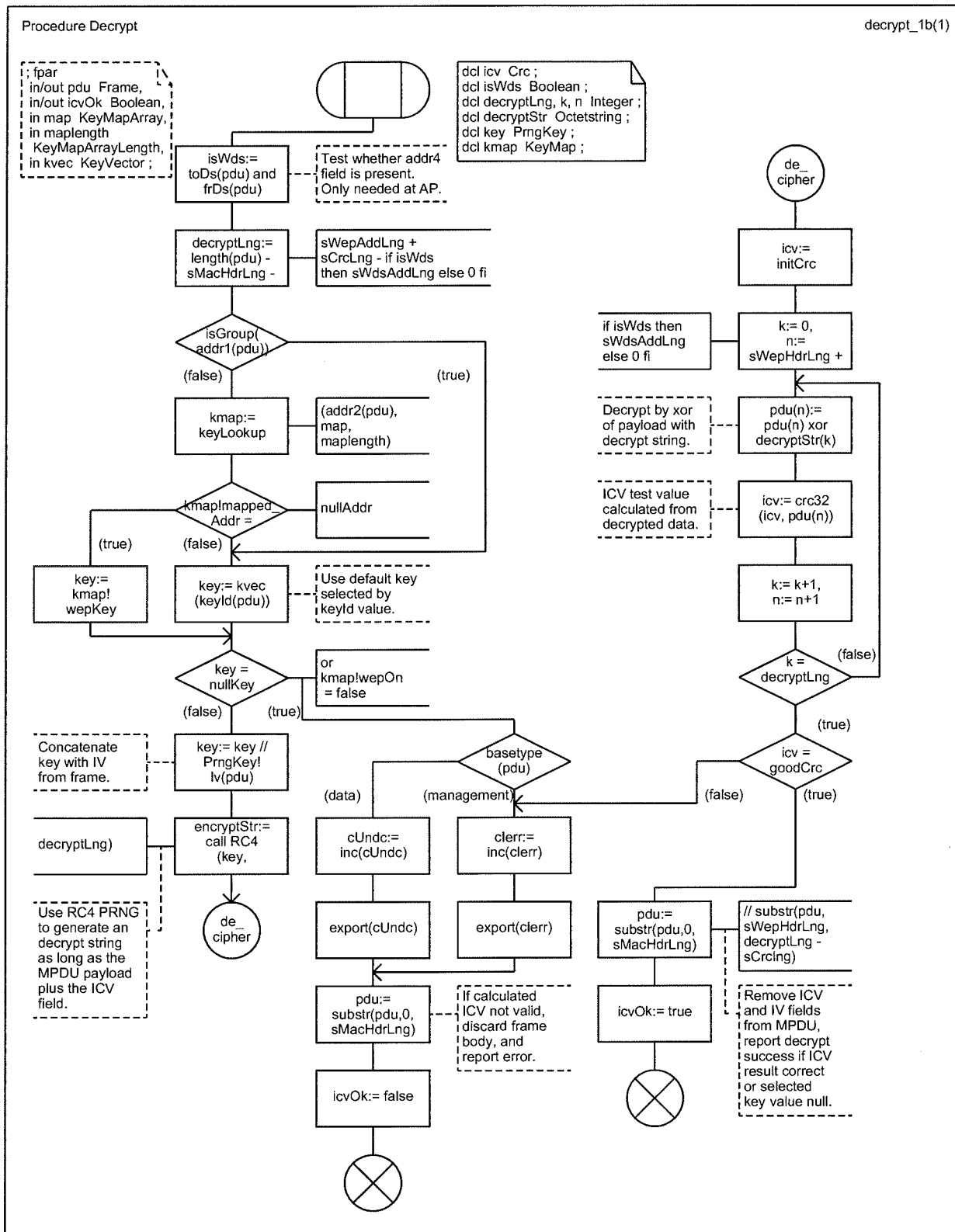








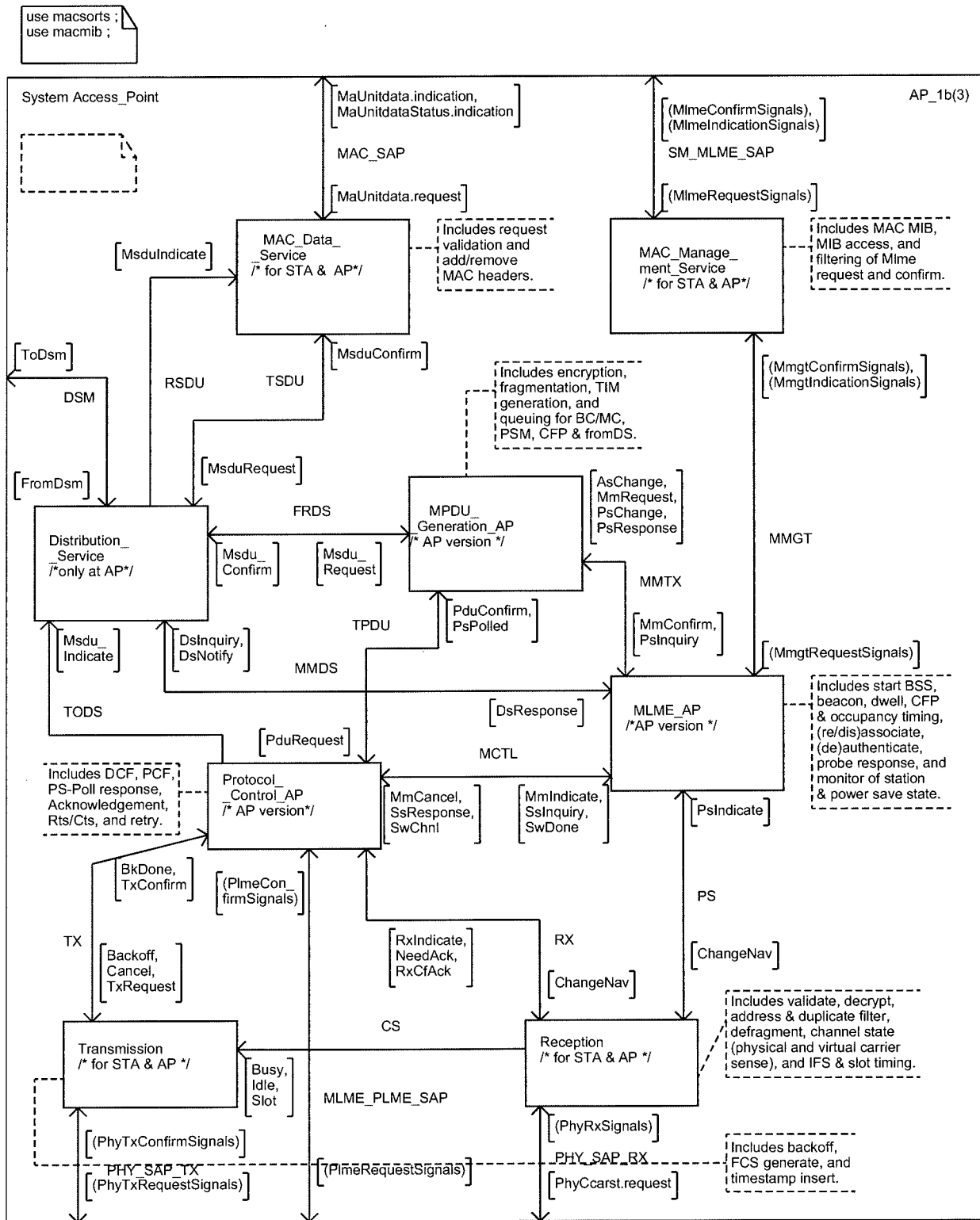




C.4 State machines for MAC access point

The following SDL-92 system specification defines operation of the MAC protocol at an IEEE 802.11 AP. Many aspects of AP operation are identical to the STA operation. These are defined in blocks and processes referenced from both the STA and AP system specifications. Blocks and processes used in both STA and AP are identifiable by the SDL comment `/* for STA & AP */` below the block or process name. Blocks and processes specific to AP operation are identifiable by the SDL comment `/* AP version */` below the block or process name. Definitions for the `/* AP version */` and the `/* STA & AP */` blocks and processes appear in this subclause.

The remainder of this clause is the formal description, in SDL/GR, of an IEEE 802.11 AP.



```
use macsorts ;
use macmib ;
```

System Access_Point

AP_signals_2d(3)

```
newtype DsStatus literals
assoc, disassoc, reassoc, unknown
endnewtype DsStatus ;
```

```
signal
AsChange(Frame,DsStatus),
Backoff(Integer,Integer),
BkDone(Integer),
Busy,
Cancel,
ChangeNav(Time,Duration,NavSrc),
DsInquiry(MacAddr,MacAddr),
DsNotify(MacAddr,DsStatus),
DsResponse(MacAddr,MacAddr,DsStatus),
FromDsm(MacAddr,MacAddr,Octetstring),
Idle,
MaUnitdata.indication(MacAddr,MacAddr,
    Routing,Octetstring,RxStatus,
    CFPriority,ServiceClass),
MaUnitdata.request(MacAddr,MacAddr,
    Routing,Octetstring,CFPriority,ServiceClass),
MaUnitdataStatus.indication(MacAddr,
    MacAddr,TxStatus,CFPriority,ServiceClass),
MlmeAssociate.confirm(MlmeStatus),
MlmeAssociate.indication(MacAddr),
MlmeAssociate.request(MacAddr,Kusec,Capability,Integer),
MlmeAuthenticate.confirm
    (MacAddr,AuthType,MlmeStatus),
MlmeAuthenticate.indication(MacAddr,AuthType),
MlmeAuthenticate.request(MacAddr,AuthType,Kusec),
MlmeDeauthenticate.confirm(MacAddr,MlmeStatus),
MlmeDeauthenticate.indication(MacAddr,ReasonCode),
MlmeDeauthenticate.request(MacAddr,ReasonCode),
MlmeDisassociate.confirm(MlmeStatus),
MlmeDisassociate.indication(MacAddr,ReasonCode),
MlmeDisassociate.request(MacAddr,ReasonCode),
MlmeGet.confirm(MibStatus,MibAtrib,MibValue),
MlmeGet.request(MibAtrib),
MlmeJoin.confirm(MlmeStatus),
MlmeJoin.request(BssDscr,Integer,Usec,Ratestring),
MlmePowermgt.confirm(MlmeStatus),
MlmePowermgt.request(PwrSave,Boolean,Boolean),
MlmeReassociate.confirm(MlmeStatus),
MlmeReassociate.indication(MacAddr),
MlmeReassociate.request(MacAddr,Kusec,Capability,Integer),
MlmeReset.confirm(MlmeStatus),
MlmeReset.request,
MlmeScan.confirm(BssDscrSet,MlmeStatus),
MlmeScan.request(BssTypeSet,MacAddr,Octetstring,
    ScanType,Usec,Intstring,Kusec,Kusec),
MlmeSet.confirm(MibStatus,MibAtrib),
MlmeSet.request(MibAtrib,MibValue),
MlmeStart.confirm(MlmeStatus),
MlmeStart.request(Octetstring,BssType,Kusec,
    Integer,CFParms,PhyParms,IbssParms,Usec,
    Capability,Ratestring,Ratestring) ;
```

```
signal
MmCancel,
MmConfirm(Frame,TxStatus),
MmIndicate(Frame,Time,Time,StateErr),
MmRequest(Frame,Imed,Rate),
MsduConfirm(Frame,CfPriority,TxStatus),
MsduIndicate(Frame,CfPriority),
MsduRequest(Frame,CfPriority),
NeedAck(MacAddr,Time,Duration,Rate),
PduConfirm(FragSdu,TxResult),
PduRequest(FragSdu),
PhyCca.indication(Ccstatus),
PhyCcarst.confirm,
PhyCcarst.request,
PhyData.confirm,
PhyData.indication(Octet),
PhyData.request(Octet),
PhyRxEnd.indication(PhyRxStat),
PhyRxStart.indication(Integer,Rate),
PhyTxEnd.confirm,
PhyTxEnd.request,
PhyTxStart.confirm,
PhyTxStart.request(Integer,Rate),
PlmeCharacteristics.confirm(PhyChrctcs),
PlmeCharacteristics.request,
PlmeGet.confirm(MibStatus,
    MibAtrib,MibValue),
PlmeGet.request(MibAtrib),
PlmeReset.confirm(Boolean),
PlmeReset.request,
PlmeSet.confirm(MibStatus,MibAtrib),
PlmeSet.request(MibAtrib,MibValue),
PsmDone,
PsPolled(MacAddr,AsocId),
PsChange(MacAddr,PsMode),
PsIndicate(MacAddr,PsMode),
PsInquiry(MacAddr),
PsResponse(MacAddr,PsMode),
ResetMAC,
RxCfAck(MacAddr),
RxIndicate(Frame,Time,Time,Rate),
Slot,
SsInquiry(MacAddr),
SsResponse(MacAddr,
    StationState,StationState),
SwChnl(Integer,Boolean),
SwDone,
ToDsm(MacAddr,MacAddr,Octetstring),
TxConfirm,
TxRequest(Frame,Rate) ;
```

```
use macsorts ;
use macmib ;
```

System Access_Point

AP_signallists_3b(3)



```
signallist
MlmeRequestSignals=
  MlmeAssociate.request,
  MlmeAuthenticate.request,
  MlmeDeauthenticate.request,
  MlmeDisassociate.request,
  MlmeGet.request,
  MlmeJoin.request,
  MlmePowermgt.request,
  MlmeReassociate.request,
  MlmeReset.request,
  MlmeScan.request,
  MlmeSet.request,
  MlmeStart.request ;
```

```
signallist
MlmeConfirmSignals=
  MlmeAssociate.confirm,
  MlmeAuthenticate.confirm,
  MlmeDeauthenticate.confirm,
  MlmeDisassociate.confirm,
  MlmeGet.confirm,
  MlmeJoin.confirm,
  MlmePowermgt.confirm,
  MlmeReassociate.confirm,
  MlmeReset.confirm,
  MlmeScan.confirm,
  MlmeSet.confirm,
  MlmeStart.confirm ;
```

```
signallist
MlmeIndicationSignals=
  MlmeAuthenticate.indication,
  MlmeDeauthenticate.indication,
  MlmeDisassociate.indication,
  MlmeAssociate.indication,
  MlmeReassociate.indication ;
```

```
signallist
SmtRequestSignals=
  MlmeAssociate.request,
  MlmeAuthenticate.request,
  MlmeDeauthenticate.request,
  MlmeDisassociate.request,
  MlmeJoin.request,
  MlmeReassociate.request,
  MlmeScan.request,
  MlmeStart.request ;
```

```
signallist
SmtConfirmSignals=
  MlmeAssociate.confirm,
  MlmeAuthenticate.confirm,
  MlmeDeauthenticate.confirm,
  MlmeDisassociate.confirm,
  MlmeJoin.confirm,
  MlmeReassociate.confirm,
  MlmeScan.confirm,
  MlmeStart.confirm ;
```

```
signallist
SmtIndicationSignals=
  MlmeAuthenticate.indication,
  MlmeDeauthenticate.indication,
  MlmeDisassociate.indication,
  MlmeAssociate.indication,
  MlmeReassociate.indication ;
```

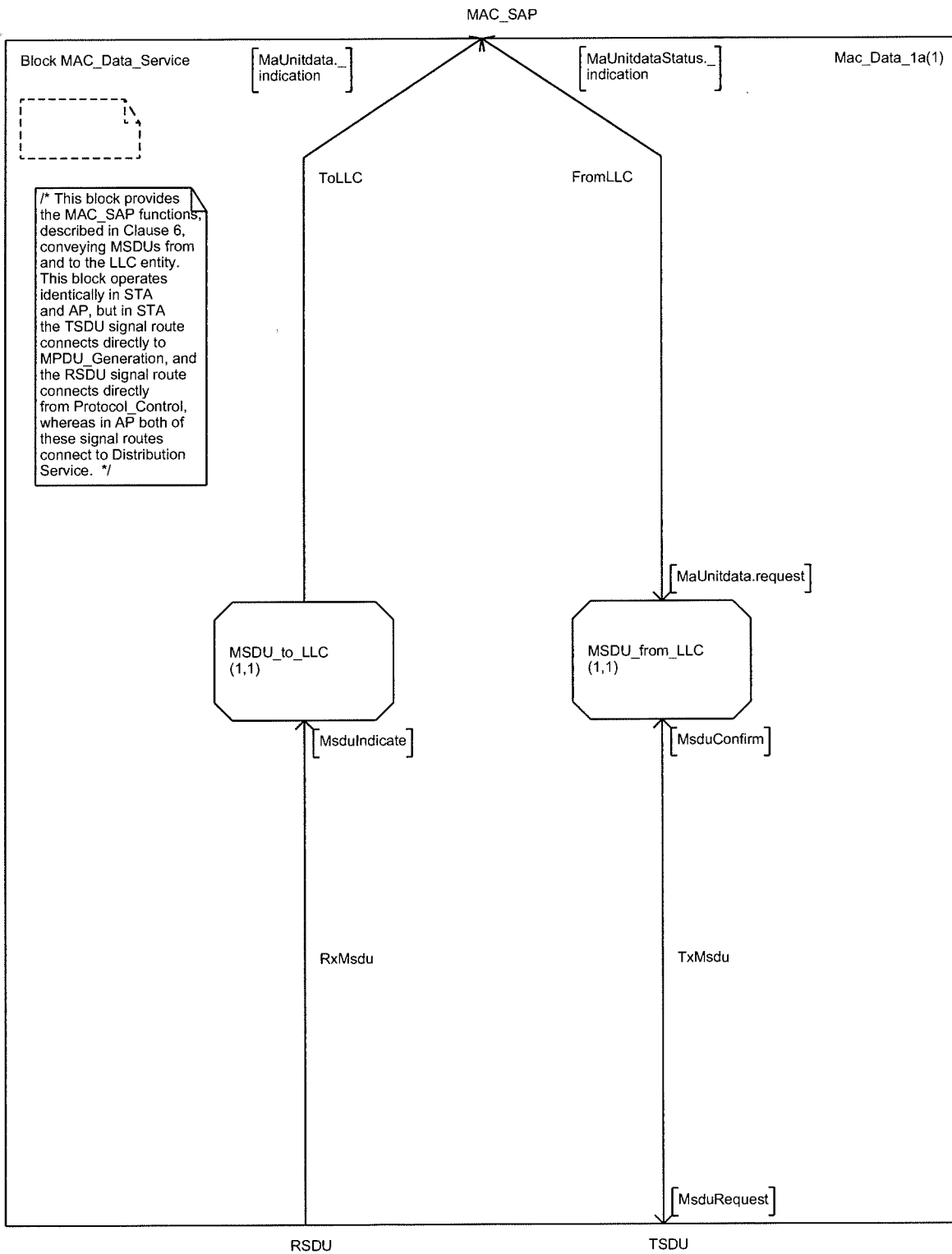
```
signallist
PhyTxRequestSignals=
  PhyTxStart.request,
  PhyTxEnd.request,
  PhyData.request ;
```

```
signallist
PhyTxConfirmSignals=
  PhyTxStart.confirm,
  PhyTxEnd.confirm,
  PhyData.confirm ;
```

```
signallist
PhyRxSignals=
  PhyRxStart.indication,
  PhyRxEnd.indication,
  PhyData.indication,
  PhyCca.indication,
  PhyCcarst.confirm ;
```

```
signallist
PlmeRequestSignals=
  PlmeCharacteristics.request,
  PlmeGet.request,
  PlmeSet.request,
  PlmeReset.request ;
```

```
signallist
PlmeConfirmSignals=
  PlmeCharacteristics.confirm,
  PlmeGet.confirm,
  PlmeSet.confirm,
  PlmeReset.confirm ;
```



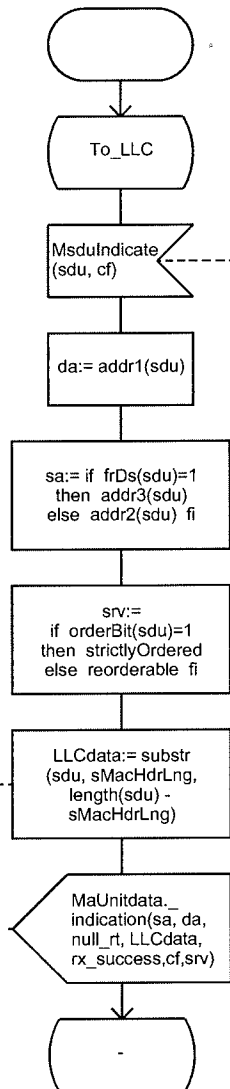
Process MSDU_to_LLC

Msd_u_to_LLC_1a(1)



```
dcl cf CfPriority ;
dcl LLCdata Octetstring ;
dcl sa, da MacAddr ;
dcl sdu Frame ;
dcl srv ServiceClass ;
```

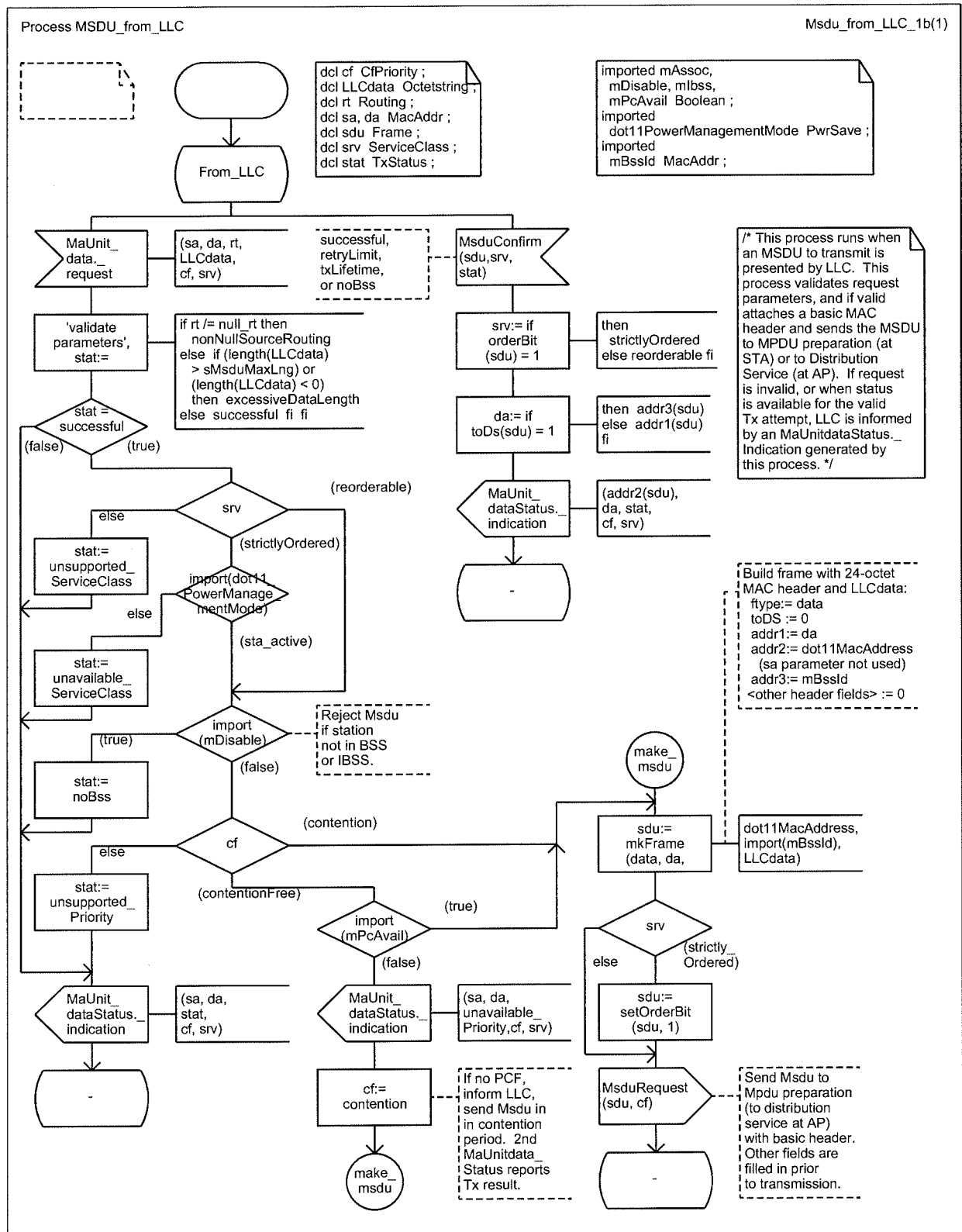
/* This process runs when reception is successfully completed on an MSDU addressed to the local LLC entity. This process extracts the appropriate address and status info, removes the MAC header from the MSDU data field (the FCS and IV/ICV are removed much earlier in reception handling), and generates the indication to LLC. Reception status is always "successful" because a receive error causes the MSDU to be discarded before reaching MAC Data Service. */

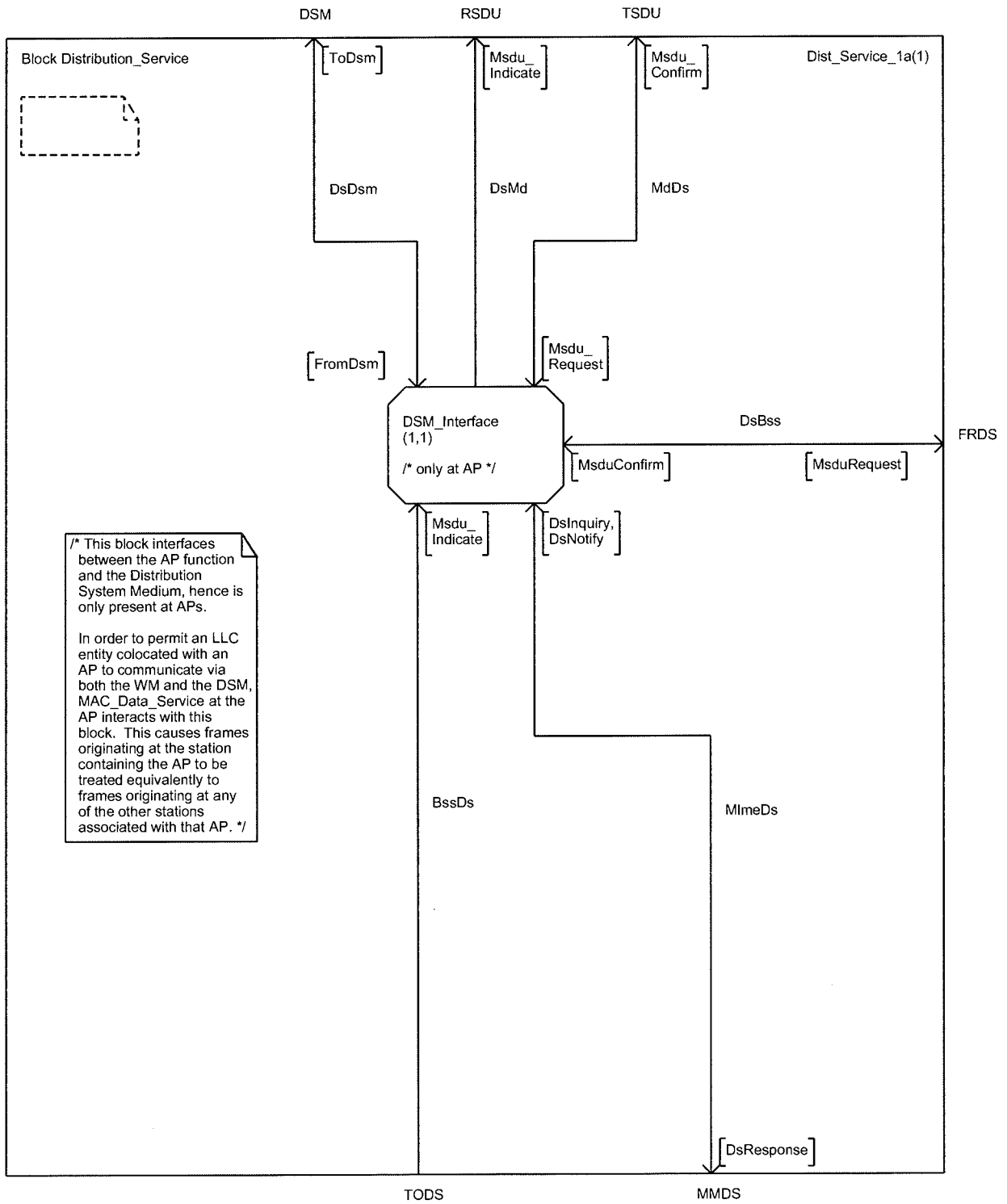


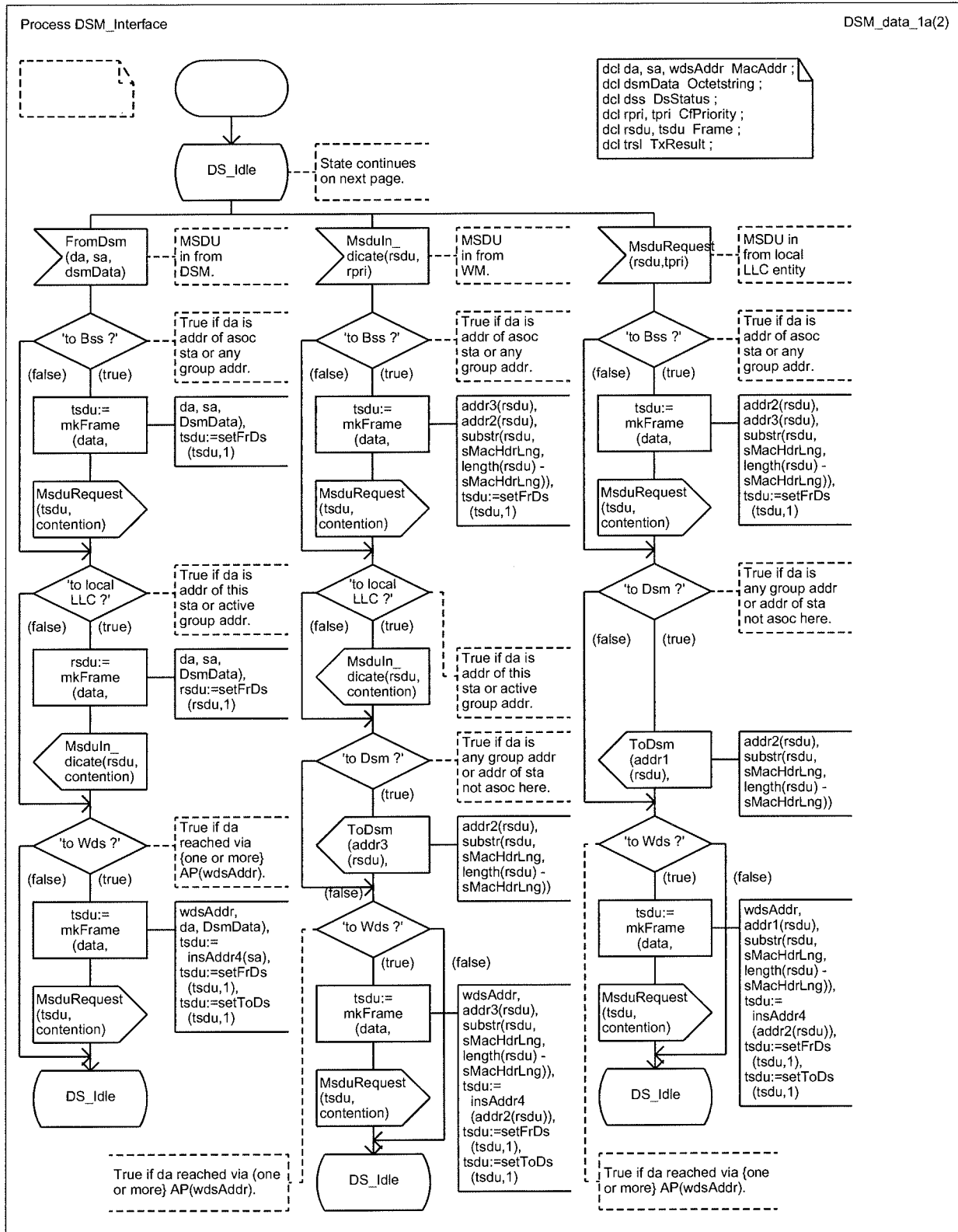
From source of the RSDU channel.
STA source is Protocol Control,
AP source is Distribution Service.

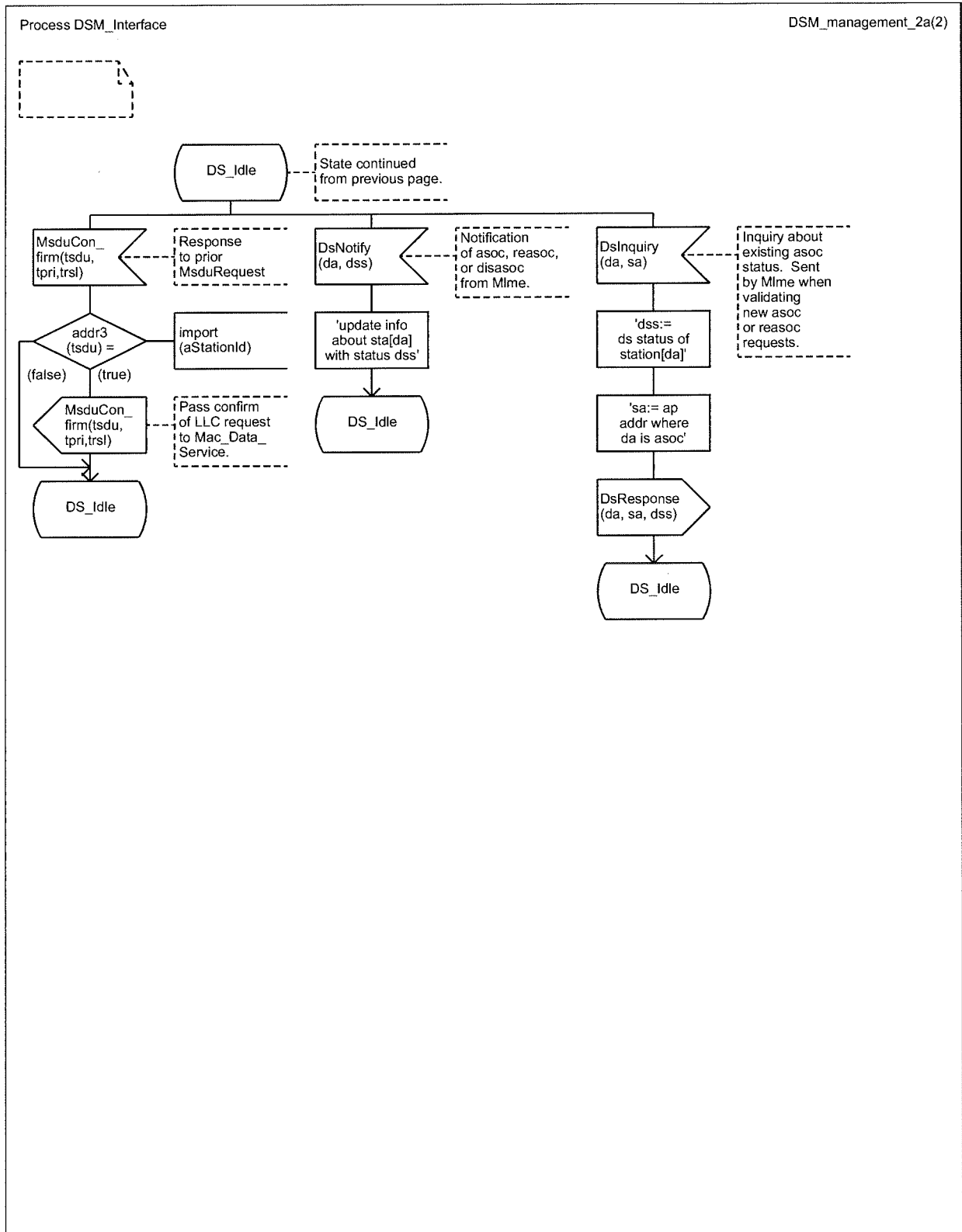
Remove MAC header from beginning of MSDU to obtain the LLC data octet string.

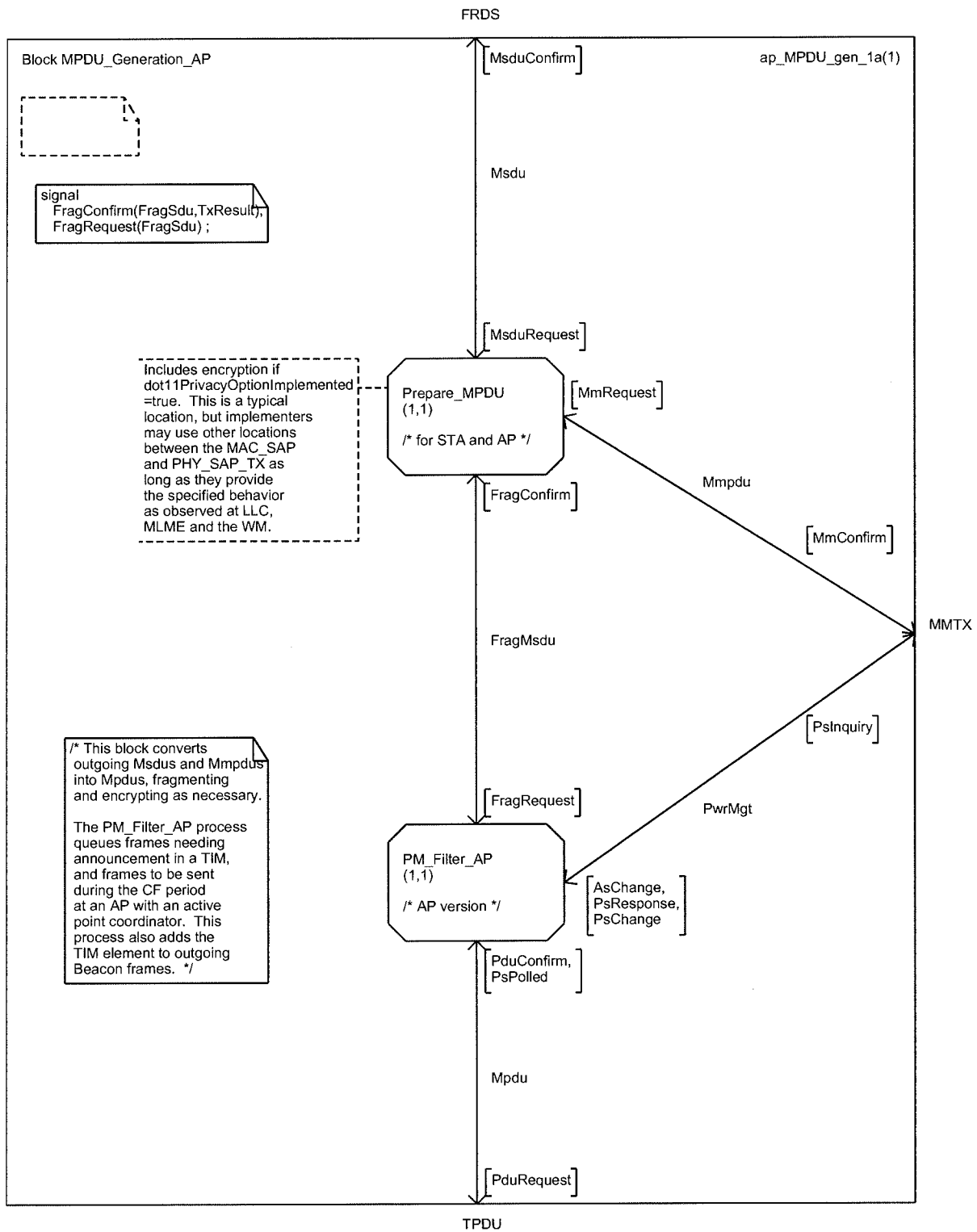
Reception status always successful because any error would prevent the MsdUIndicate from reaching this process.

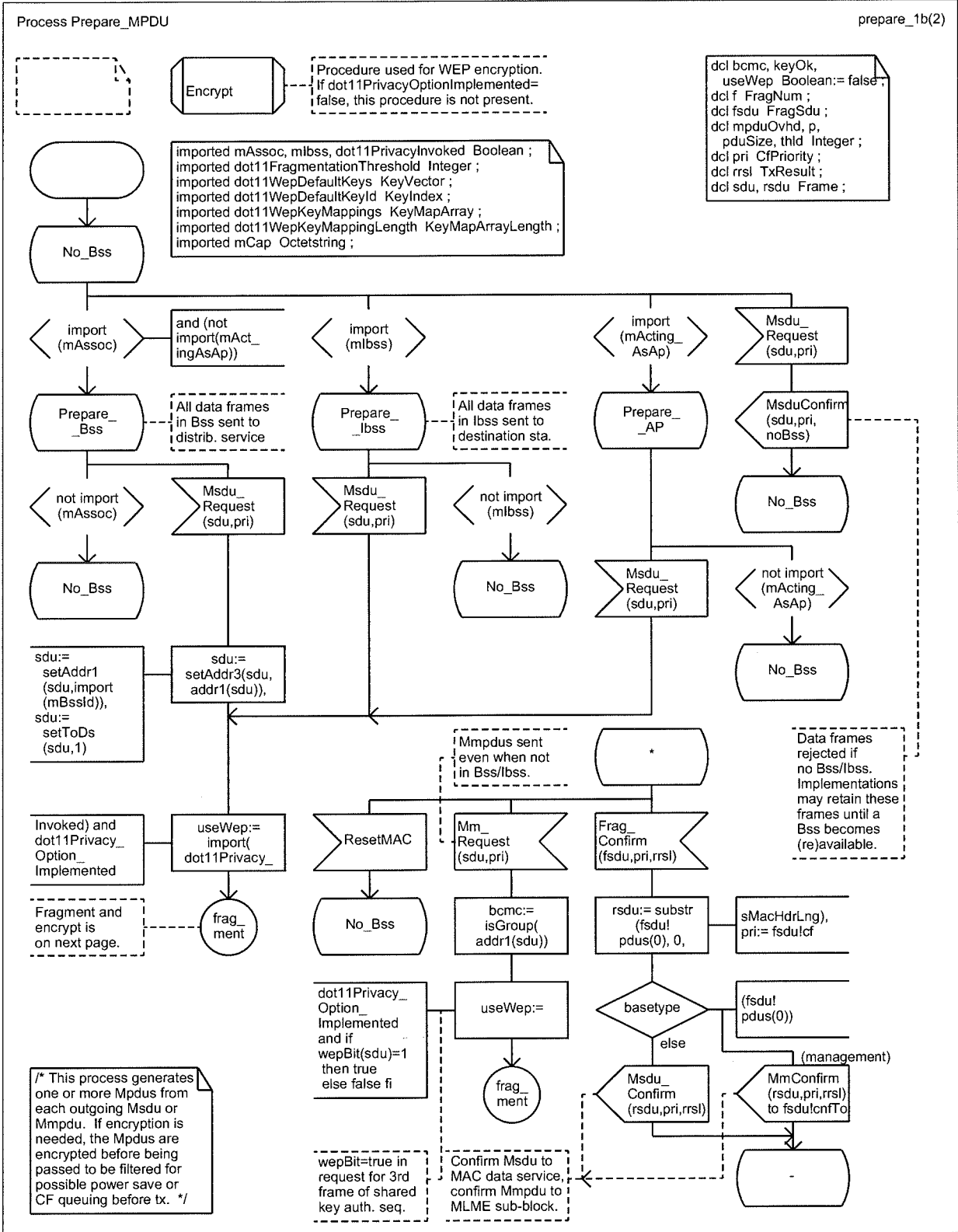


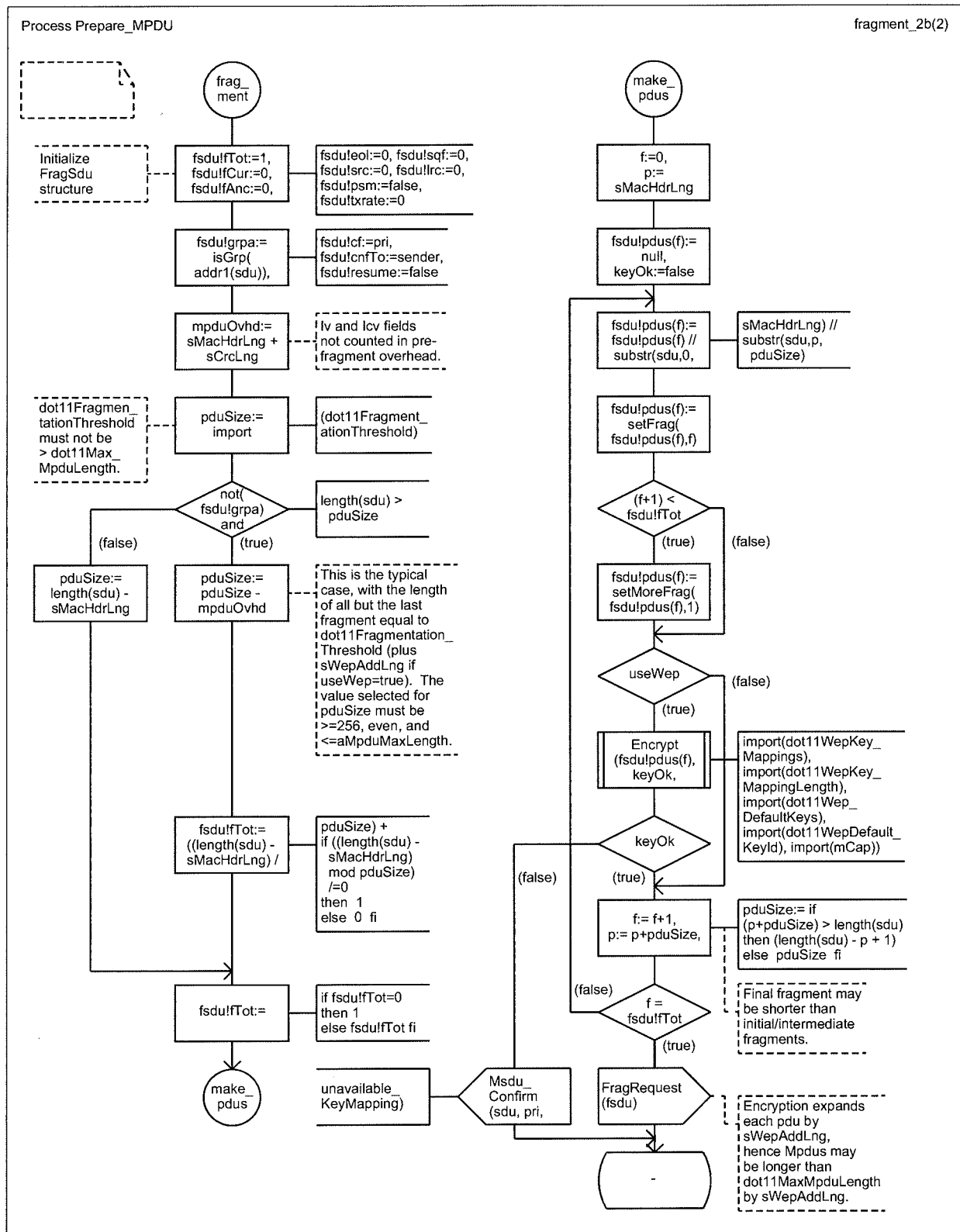


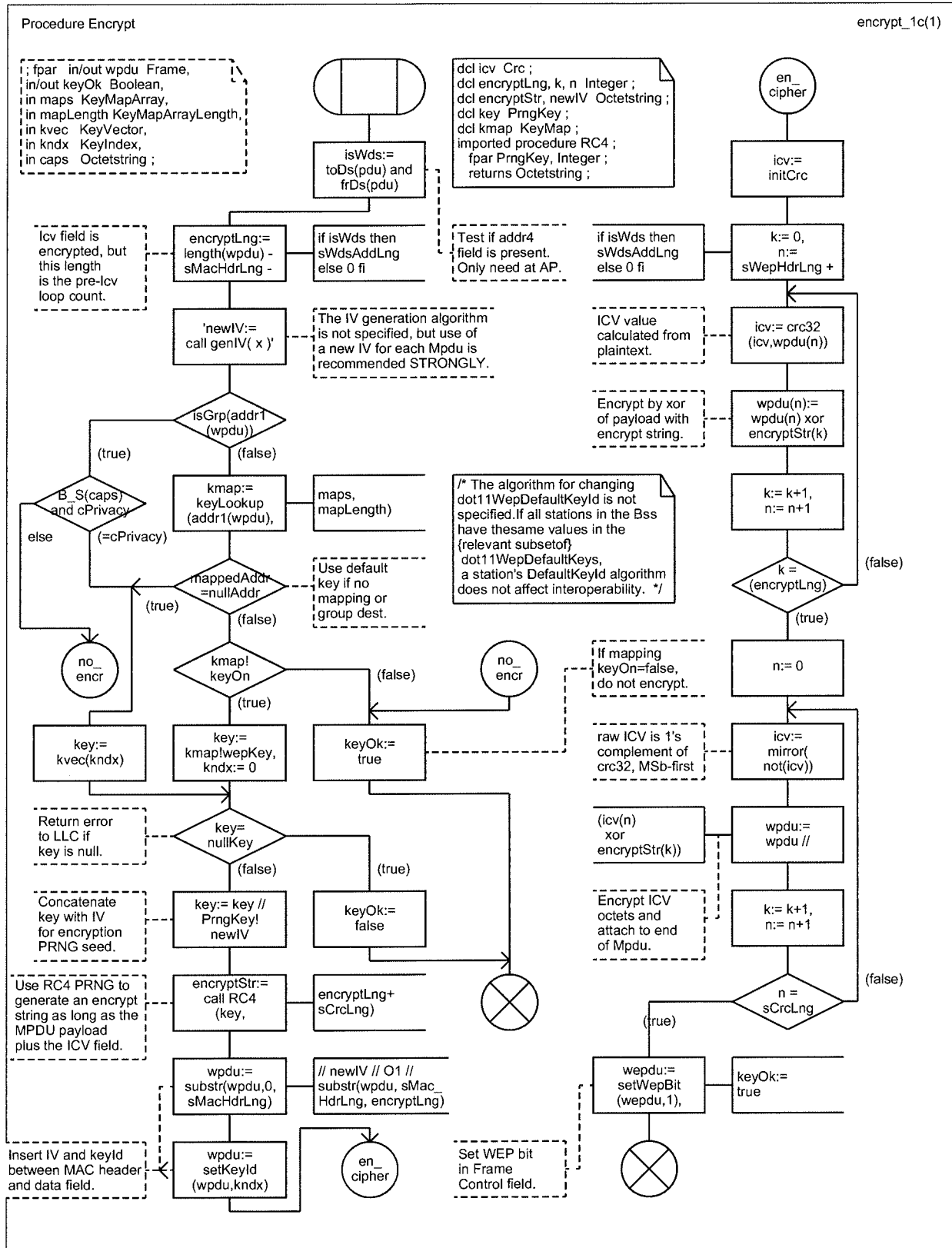


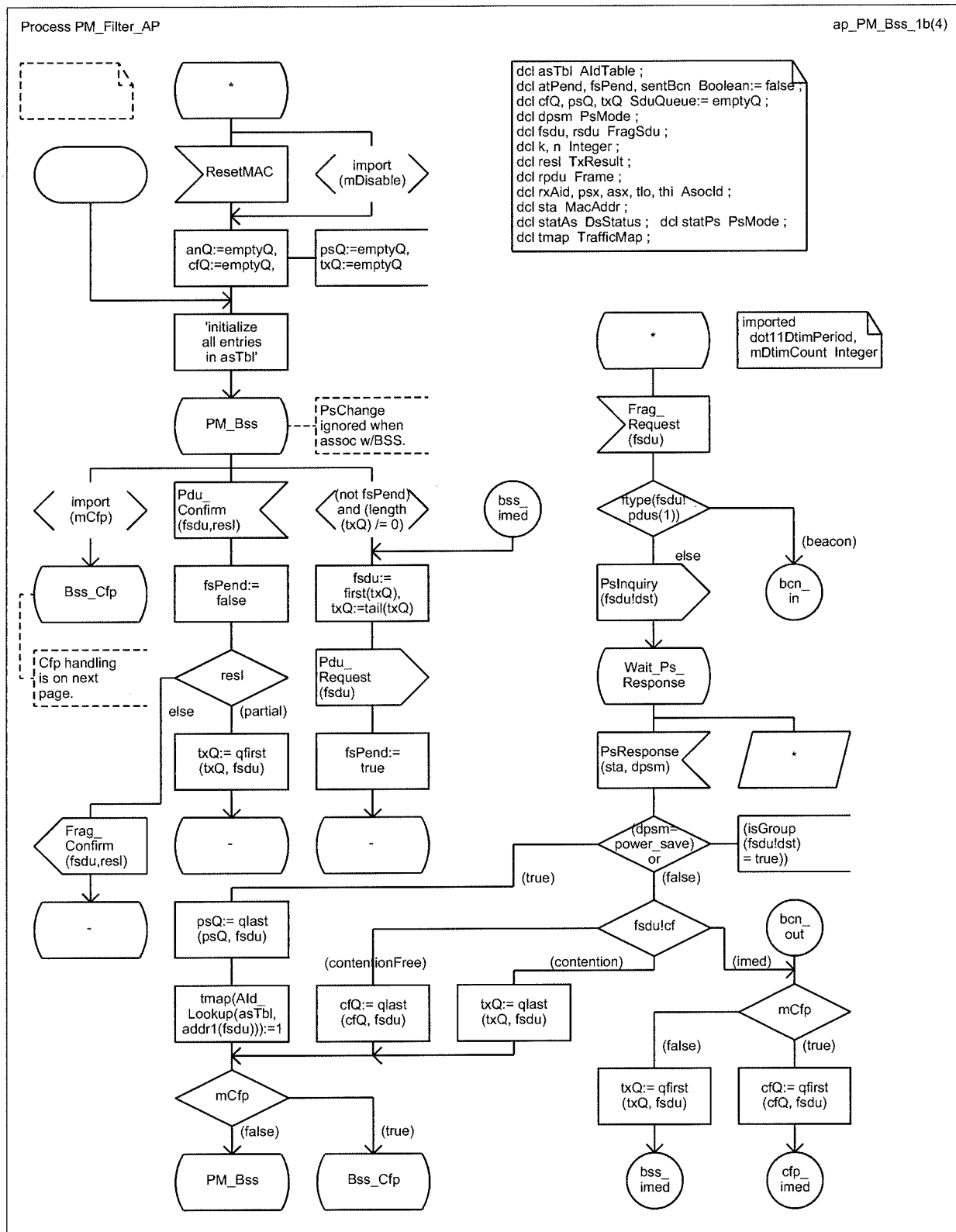


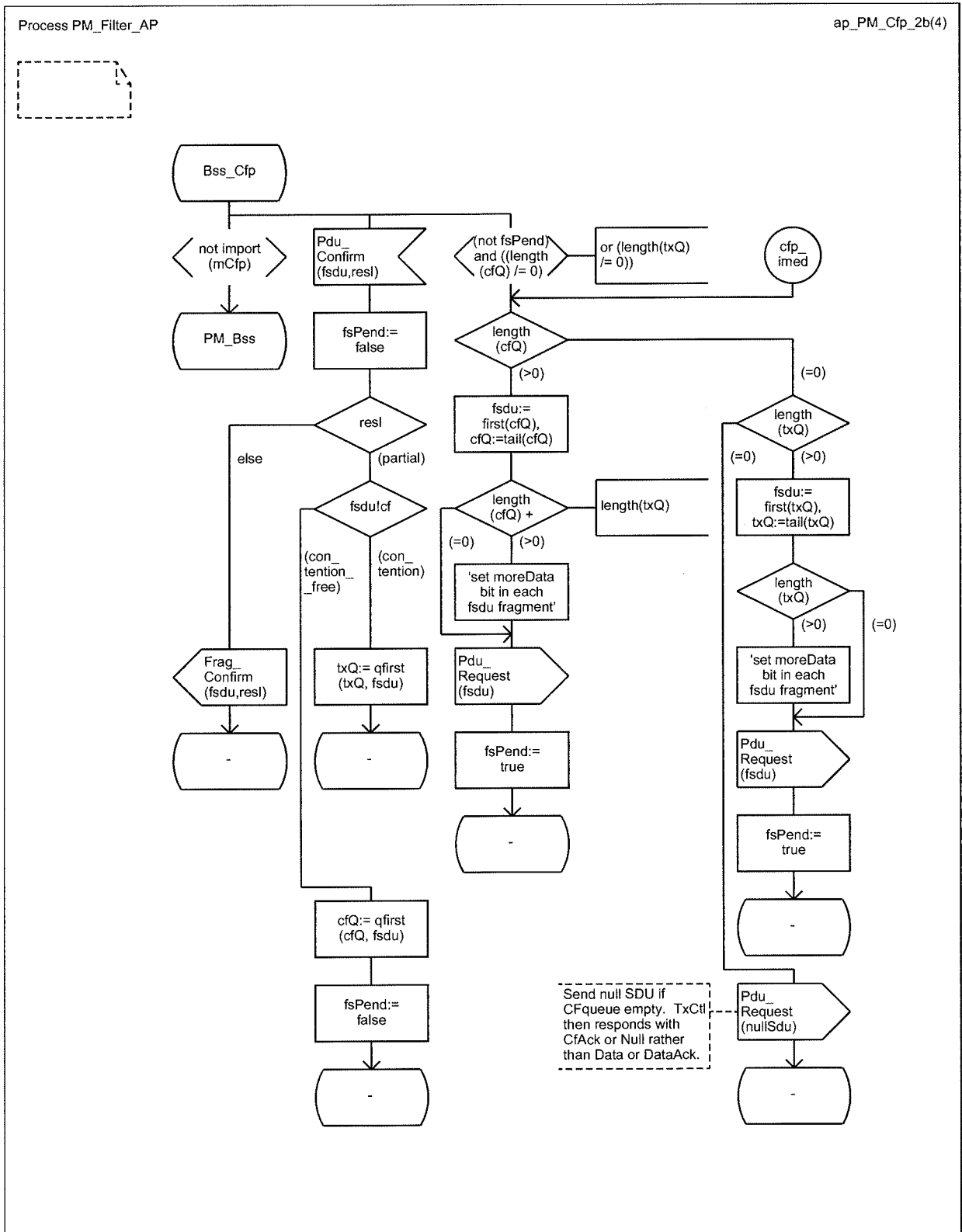


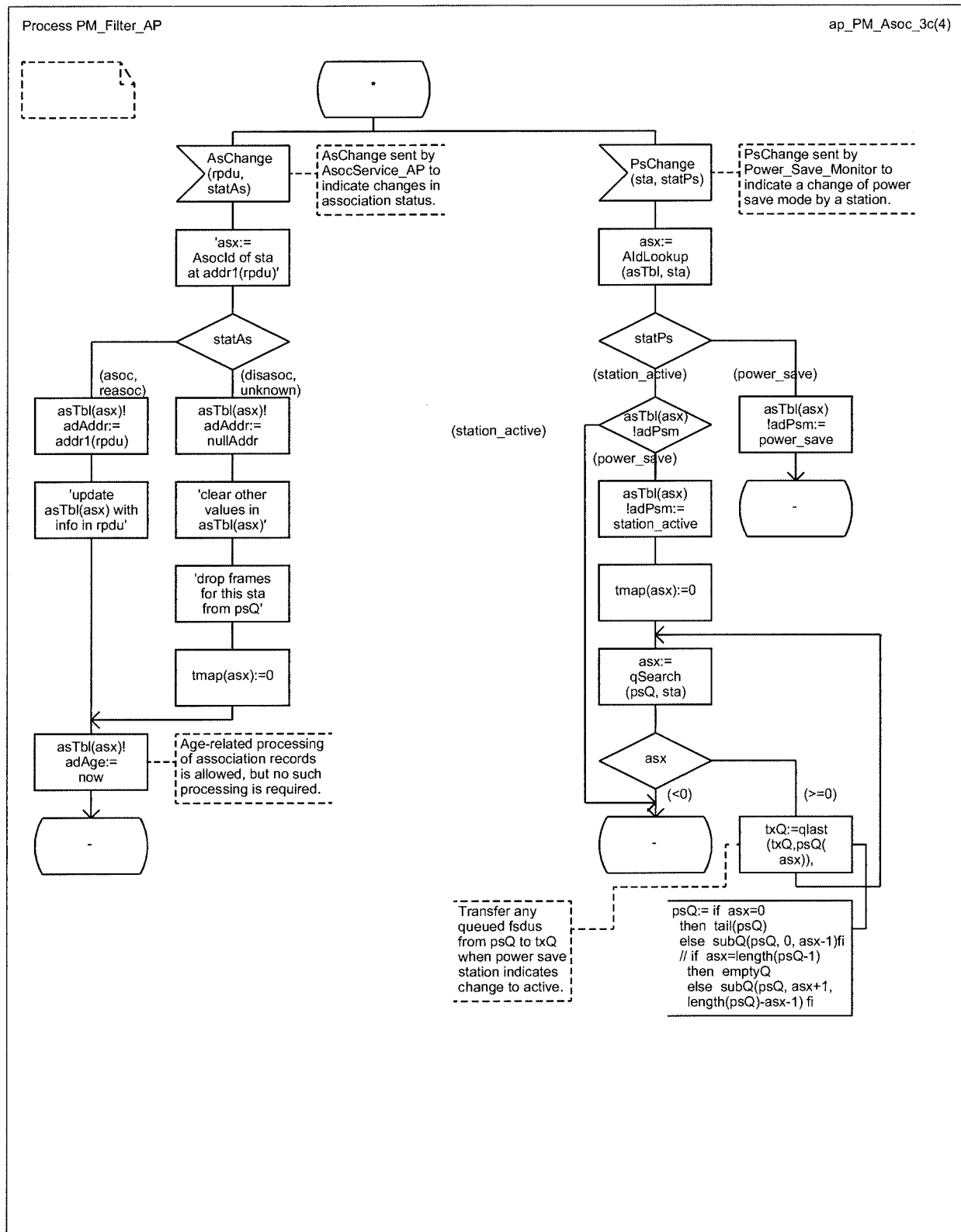


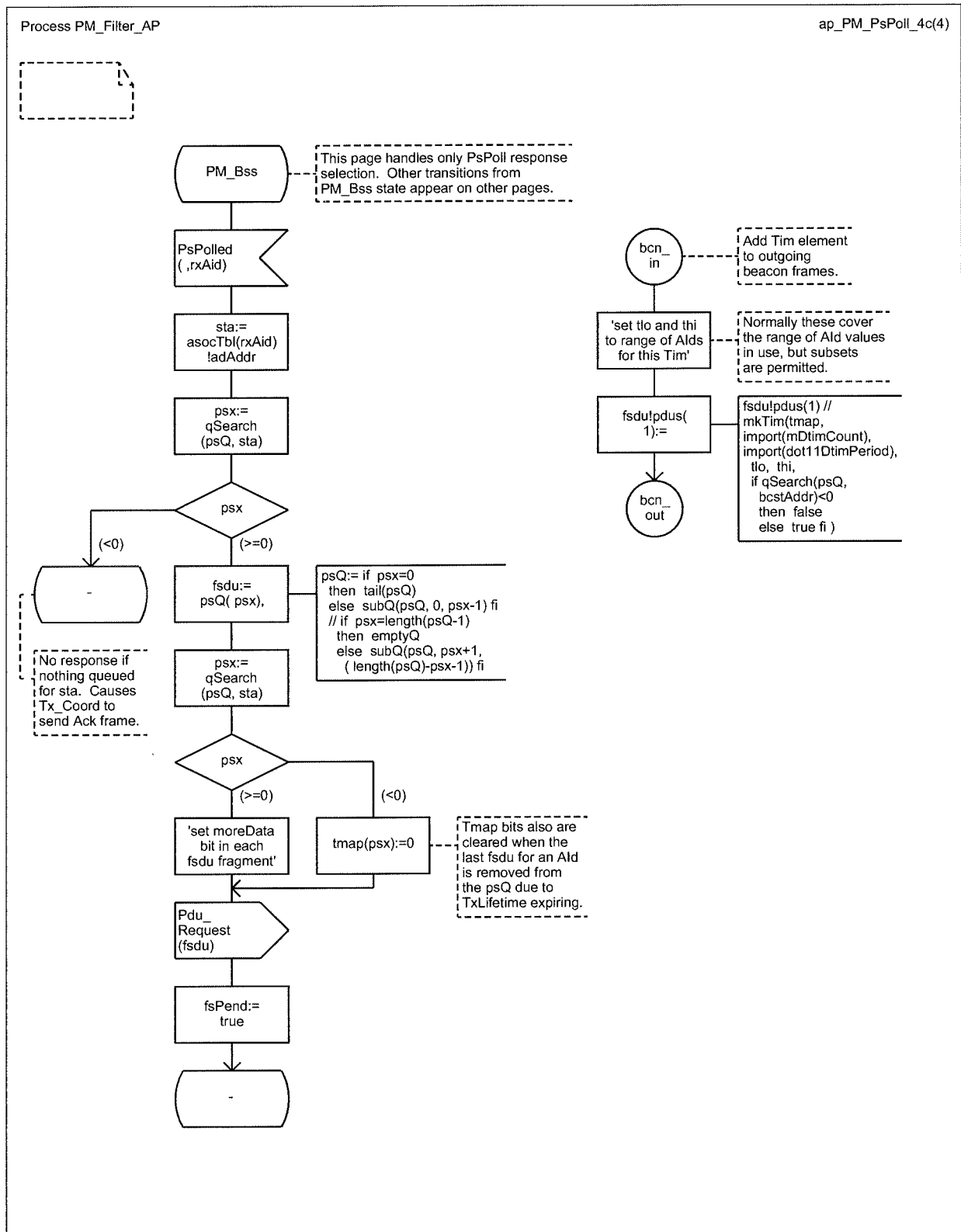


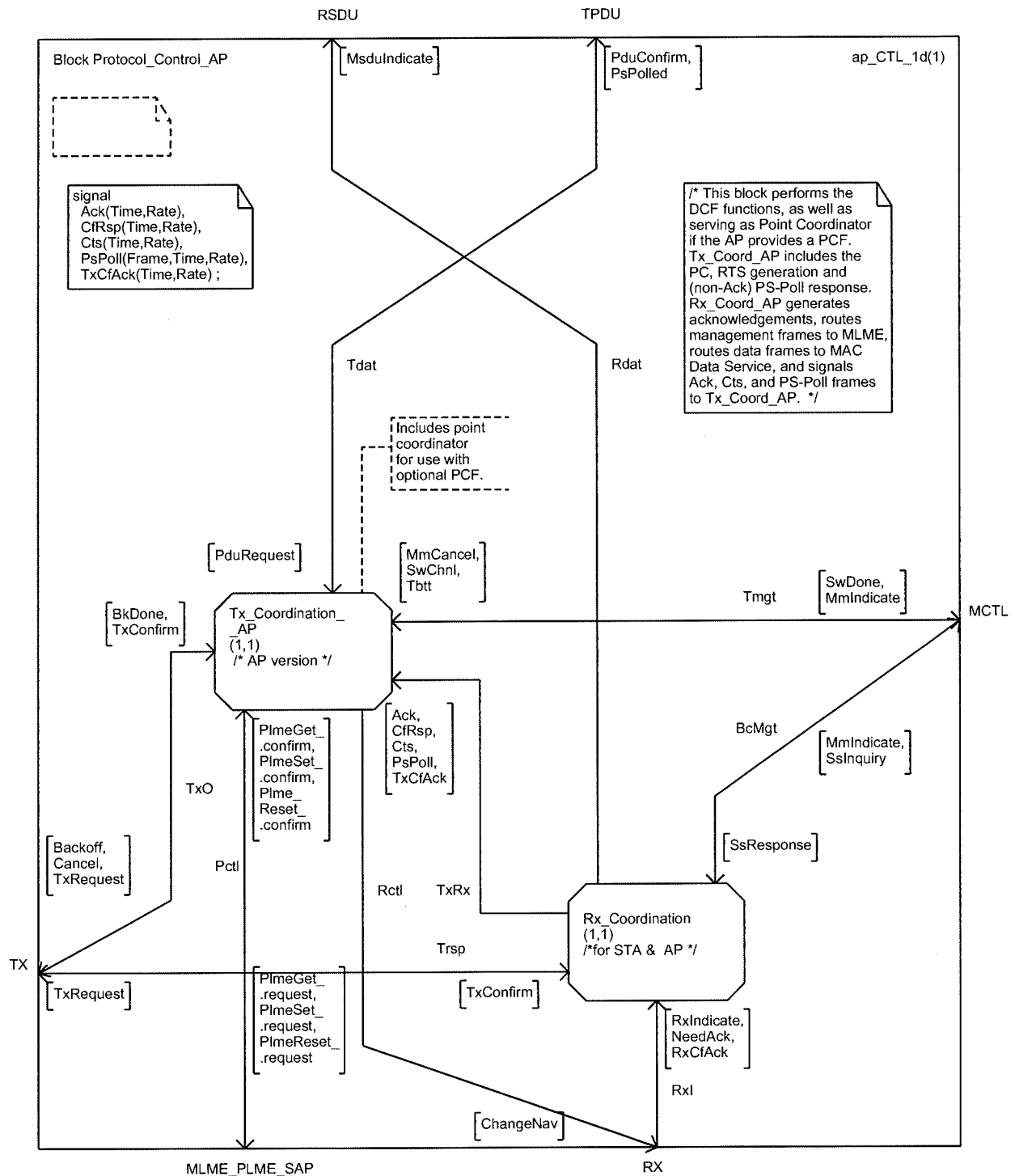


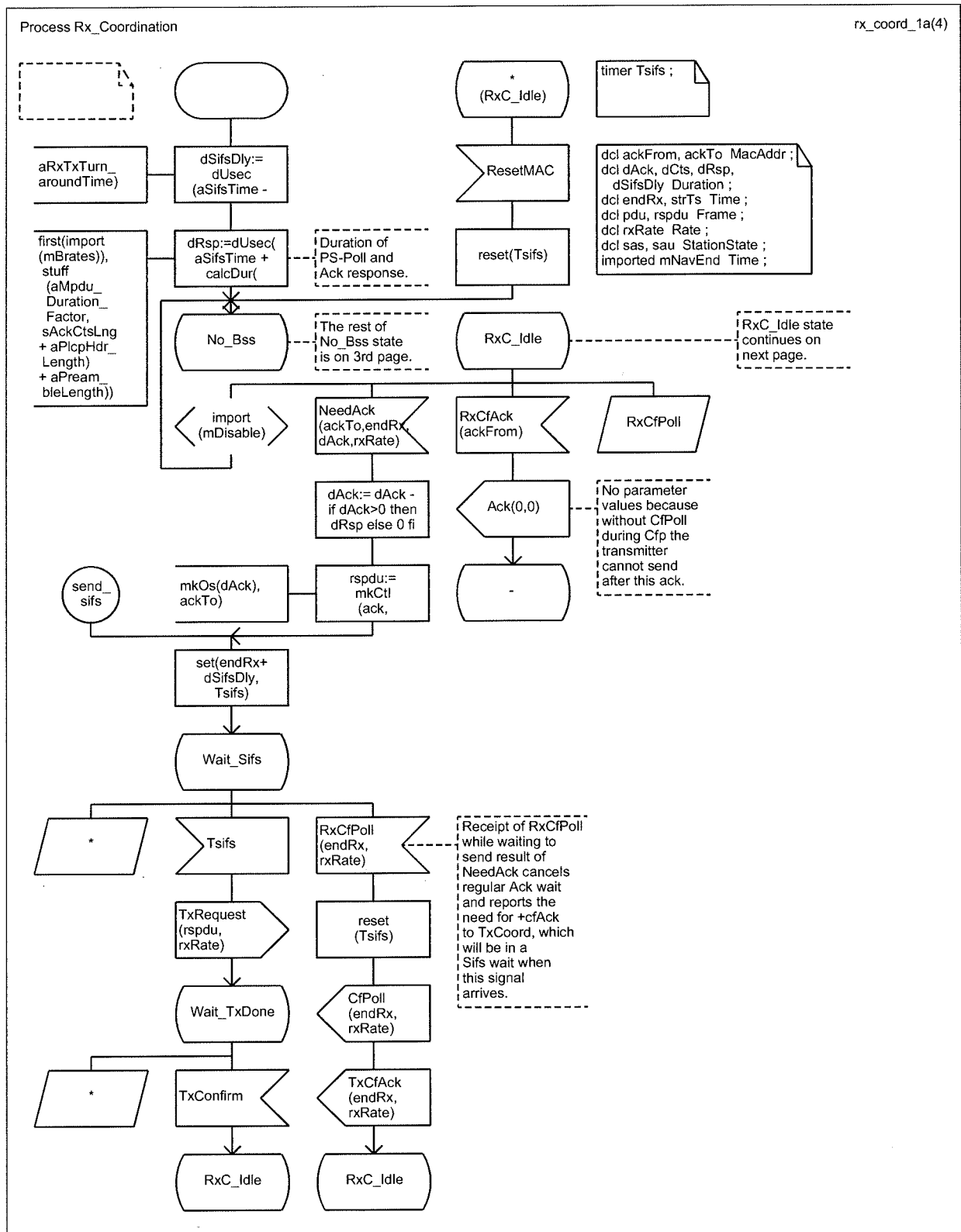


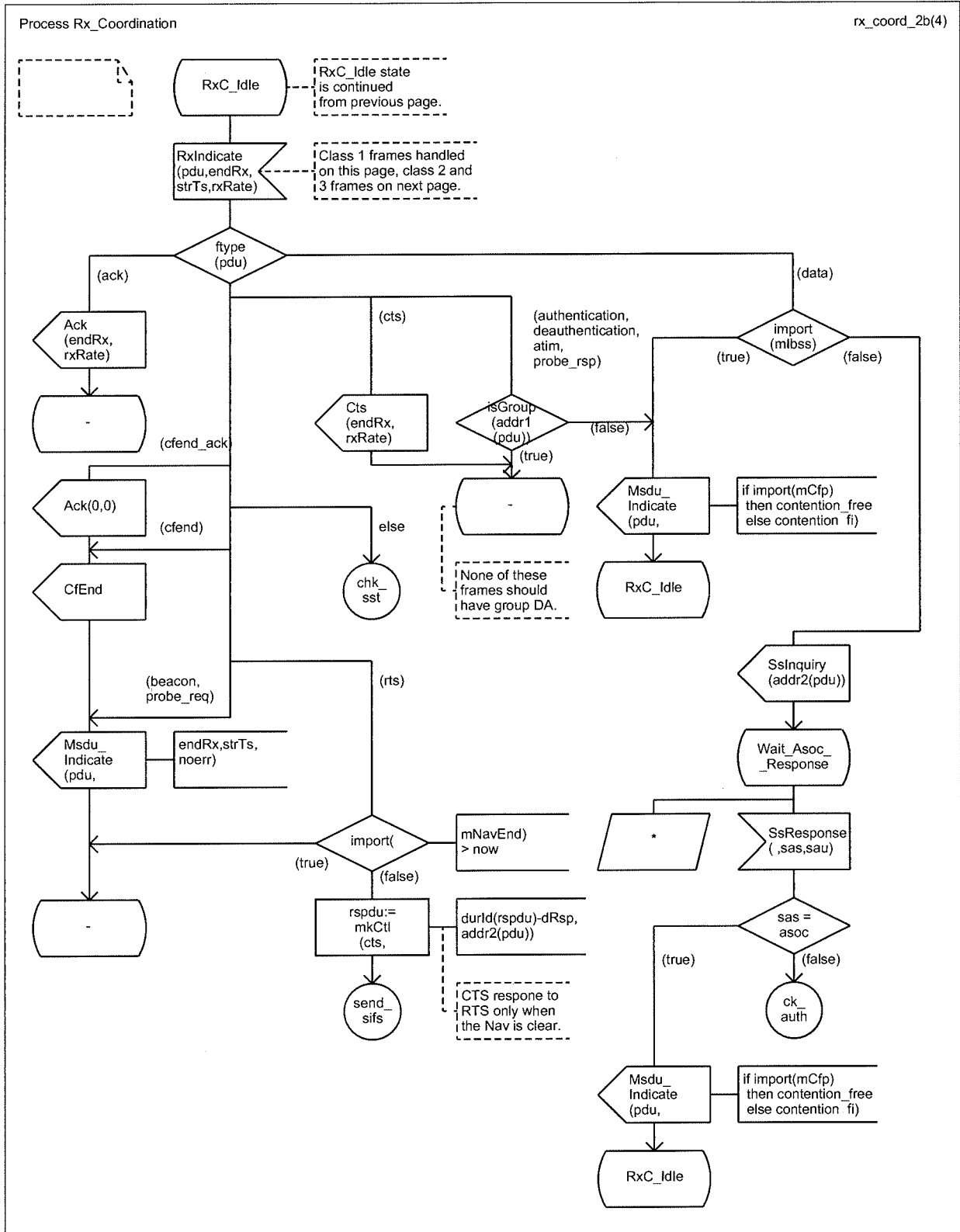


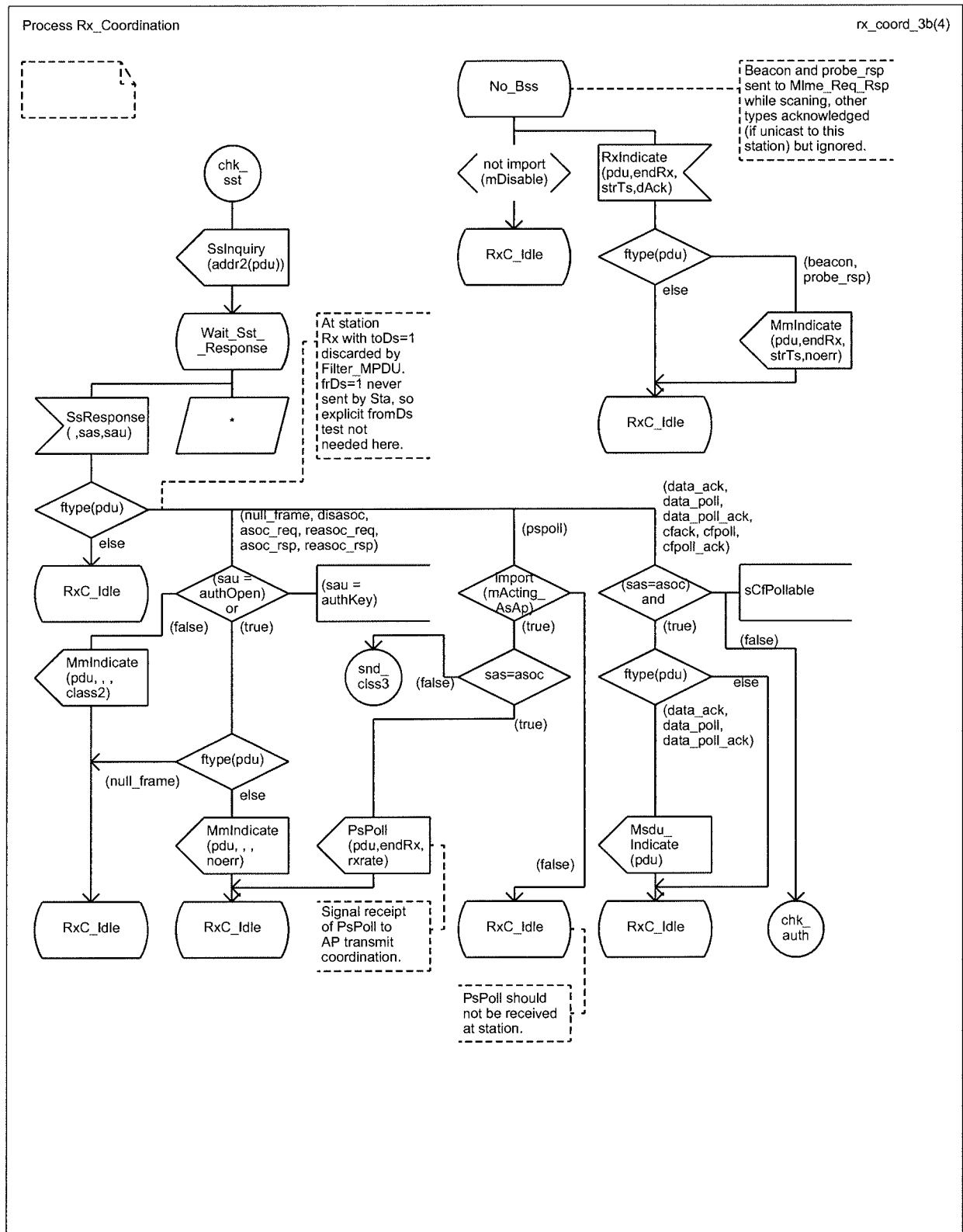






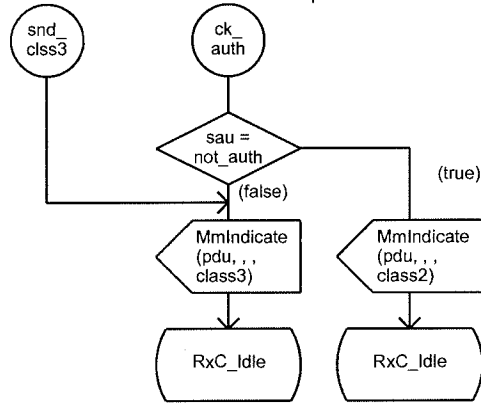


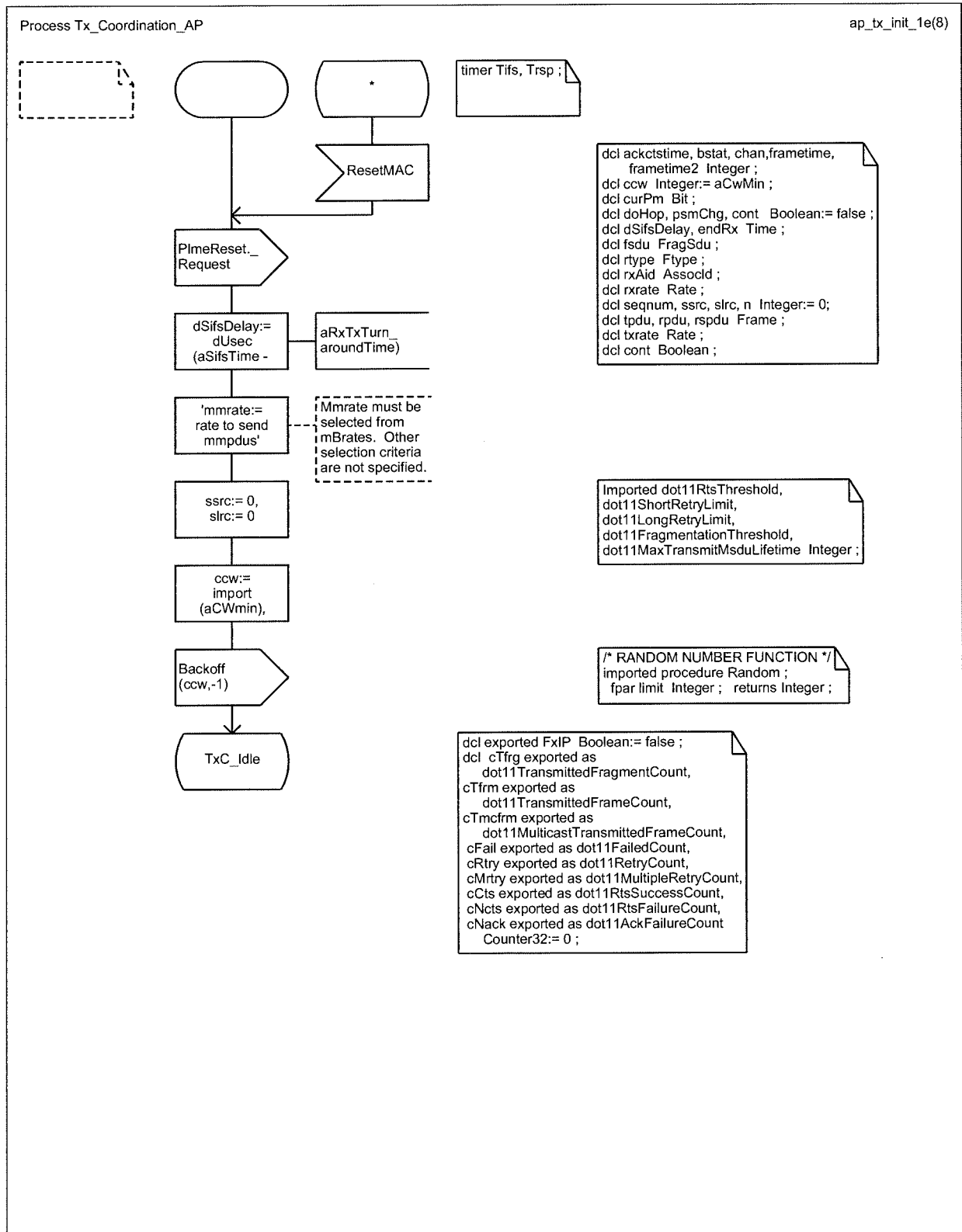


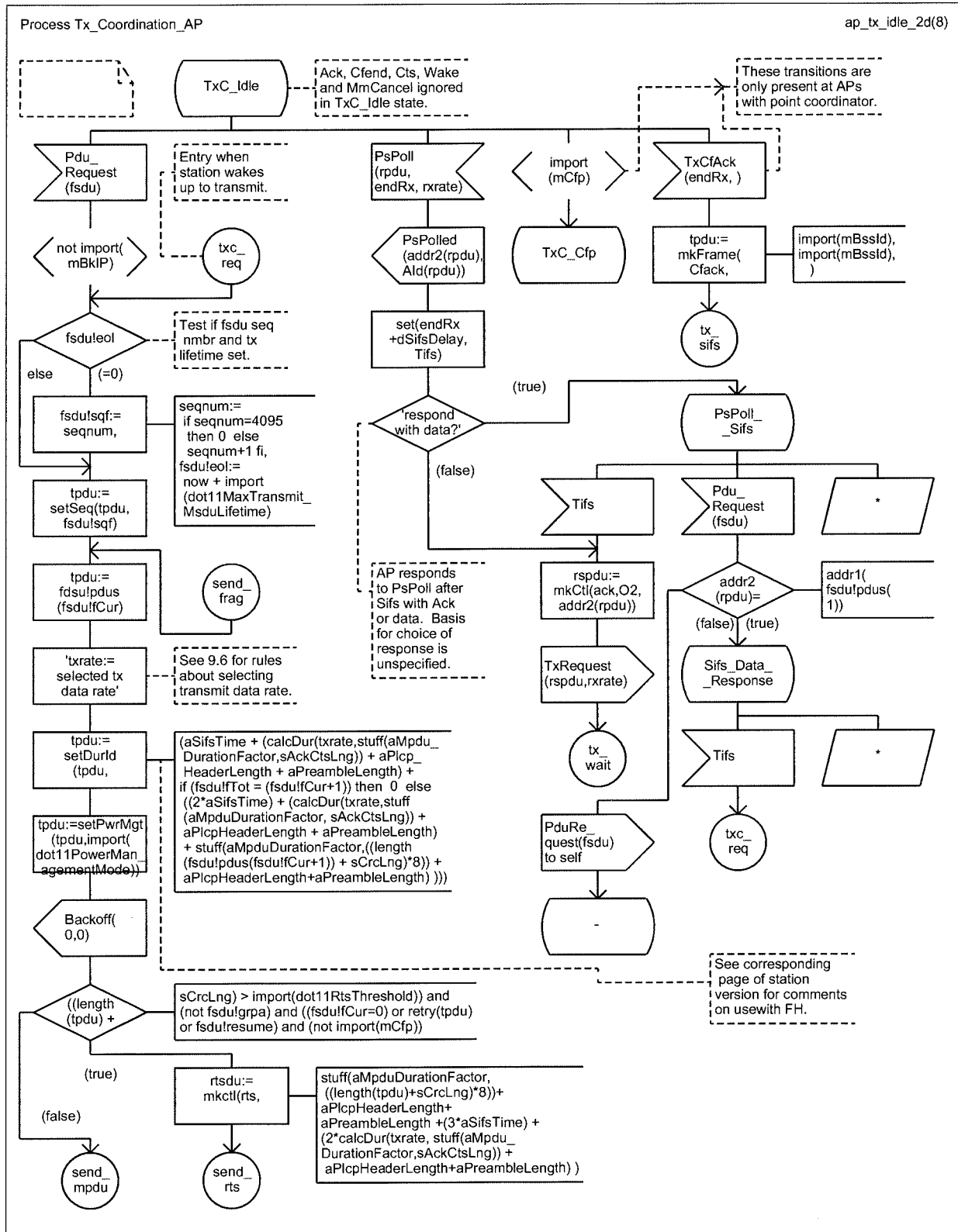


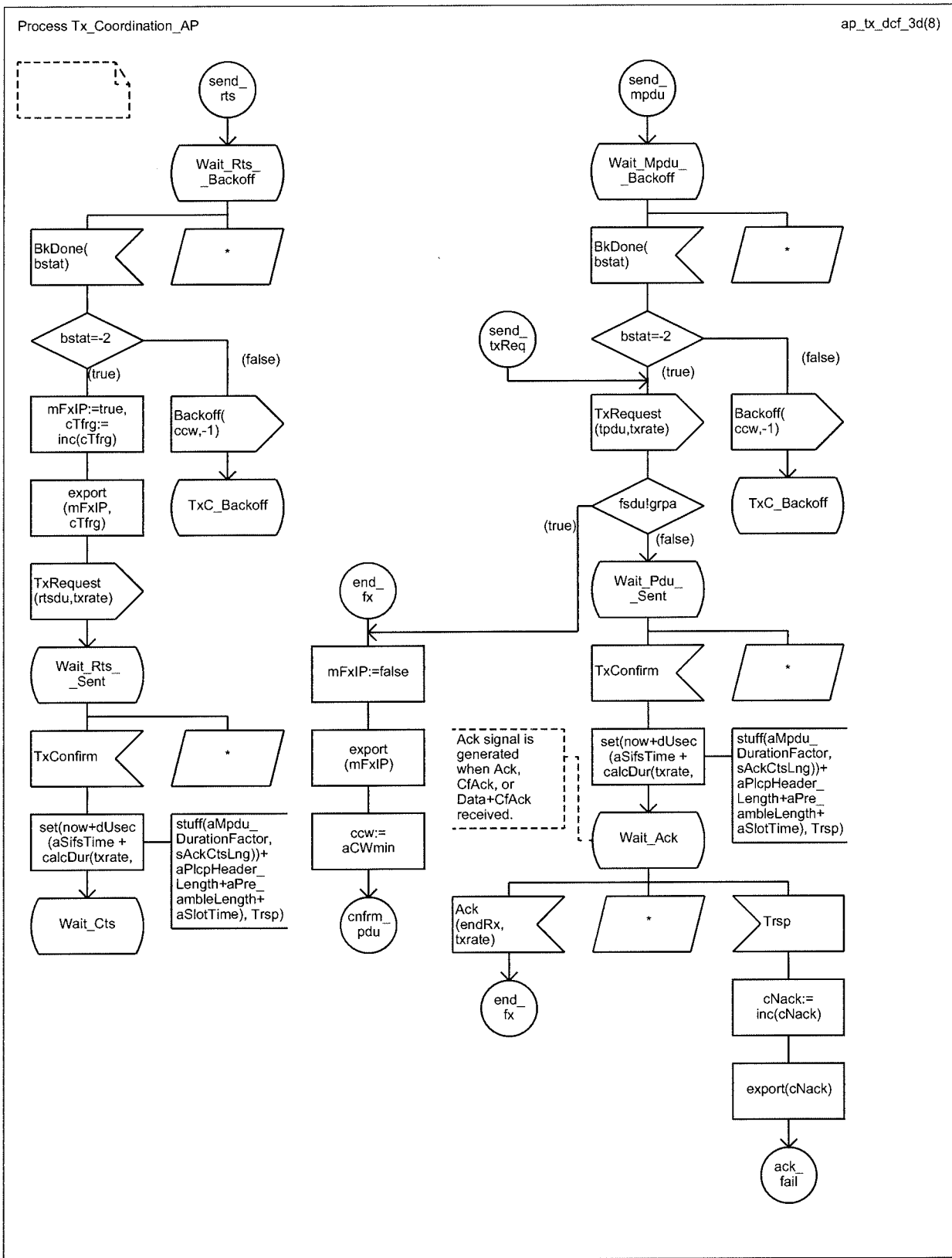
Process Rx_Coordination

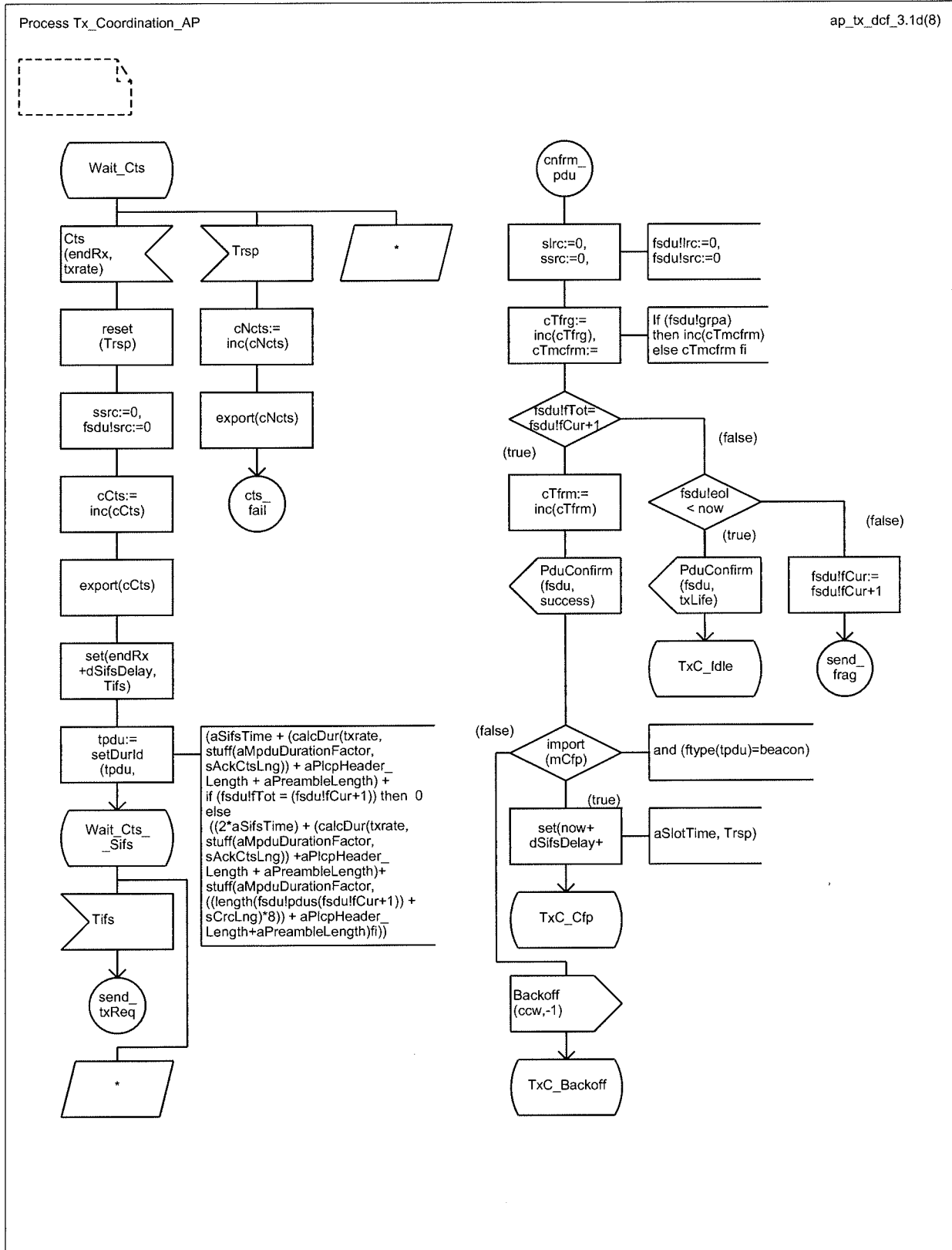
rx_coord_3.1a(4)

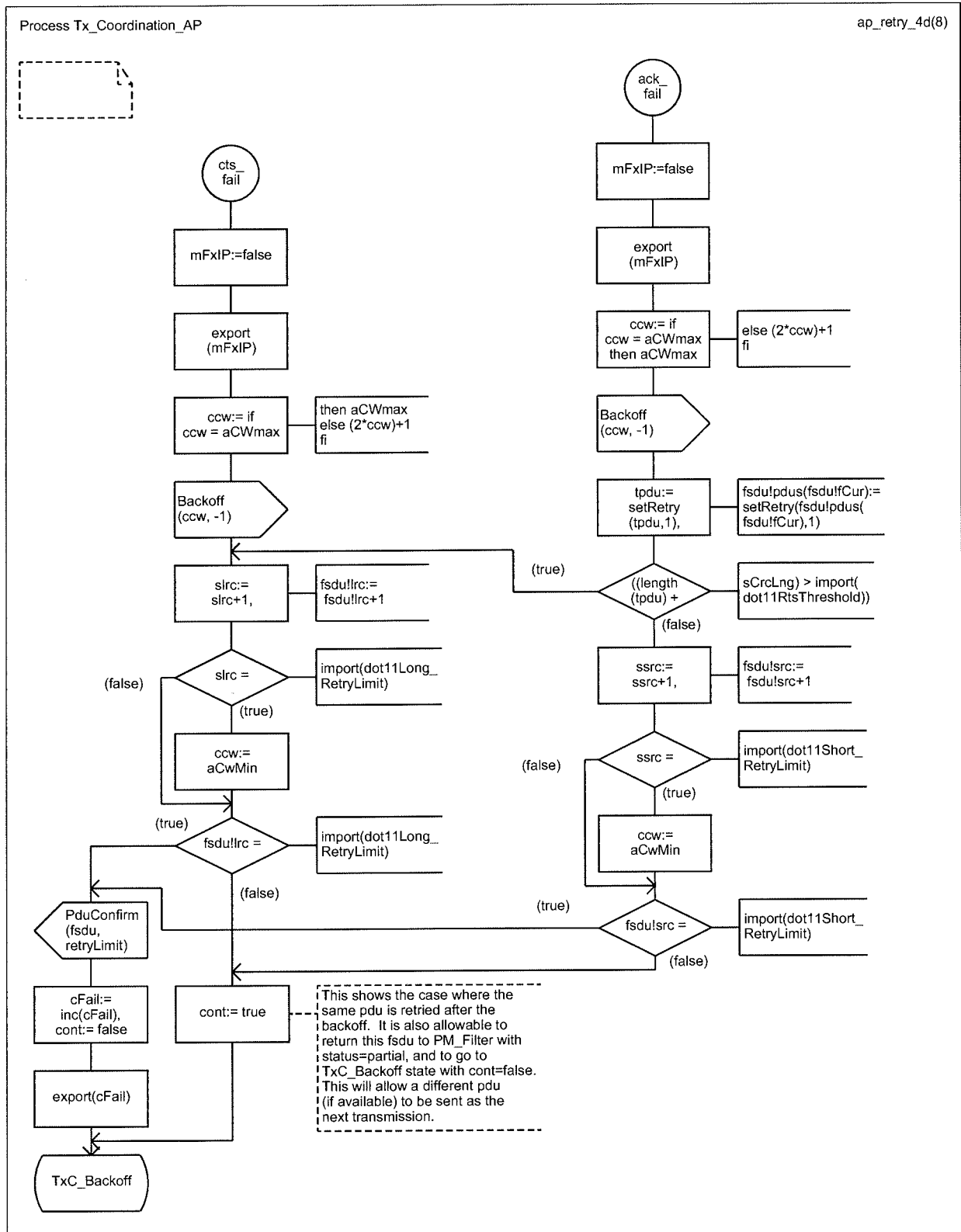


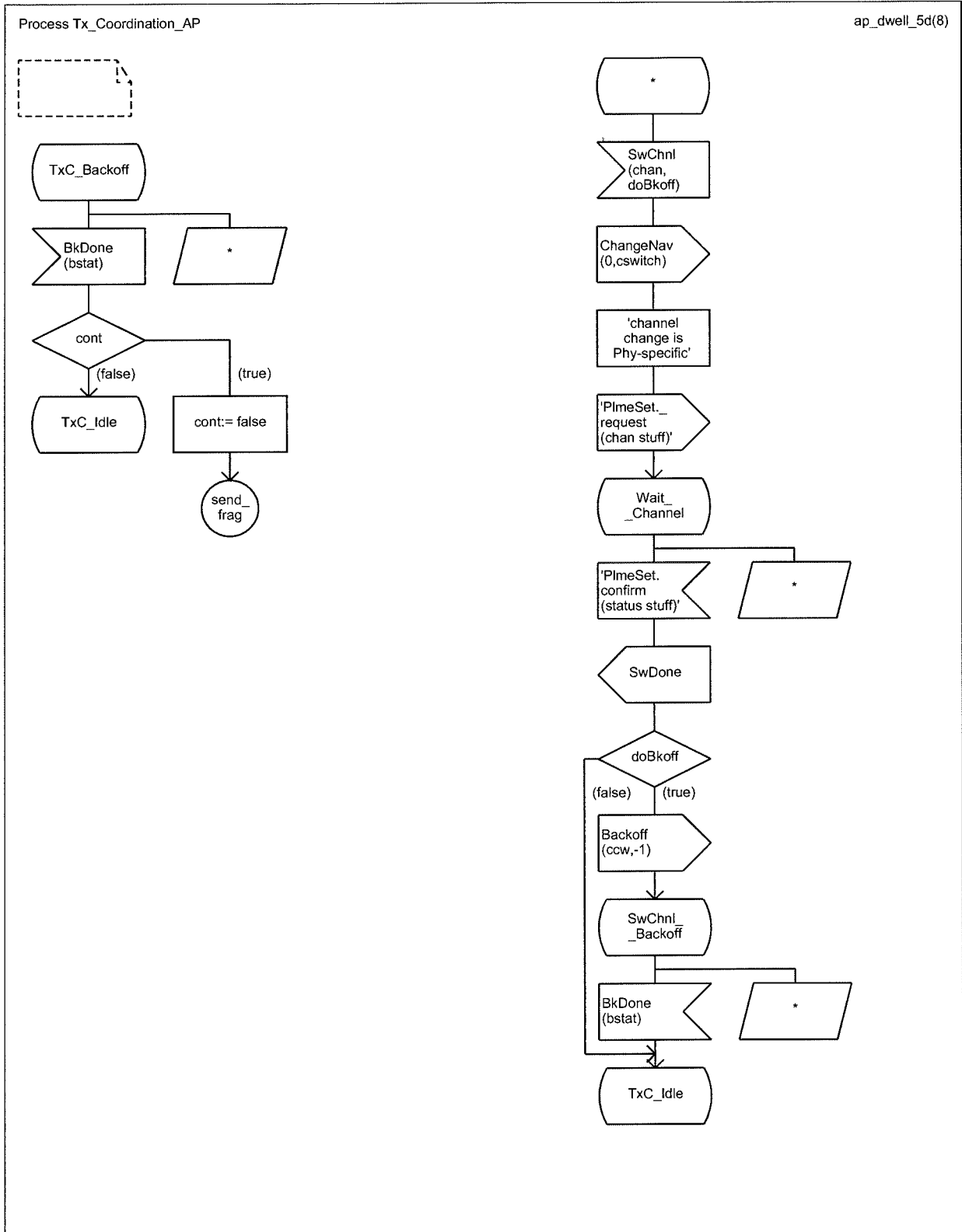


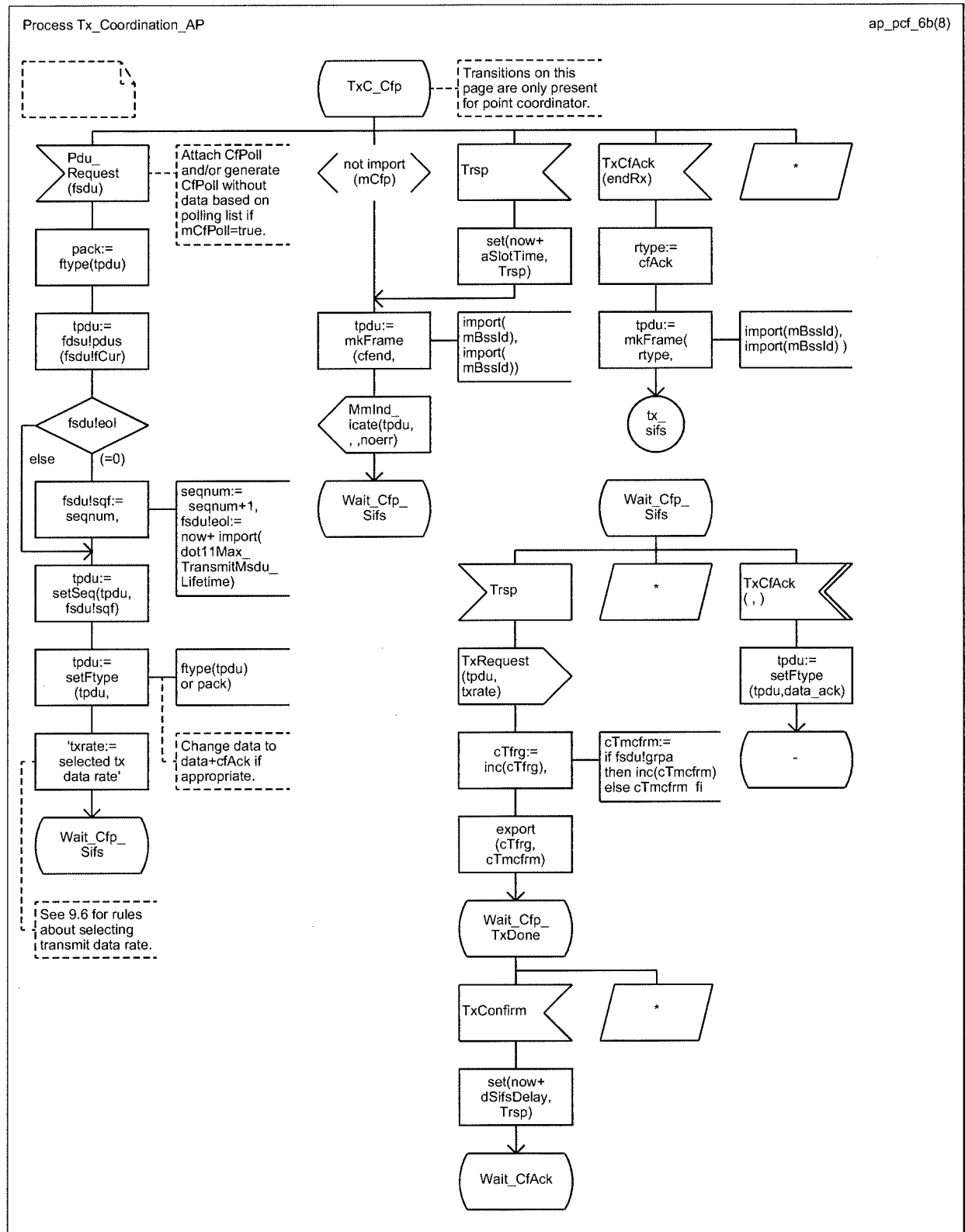


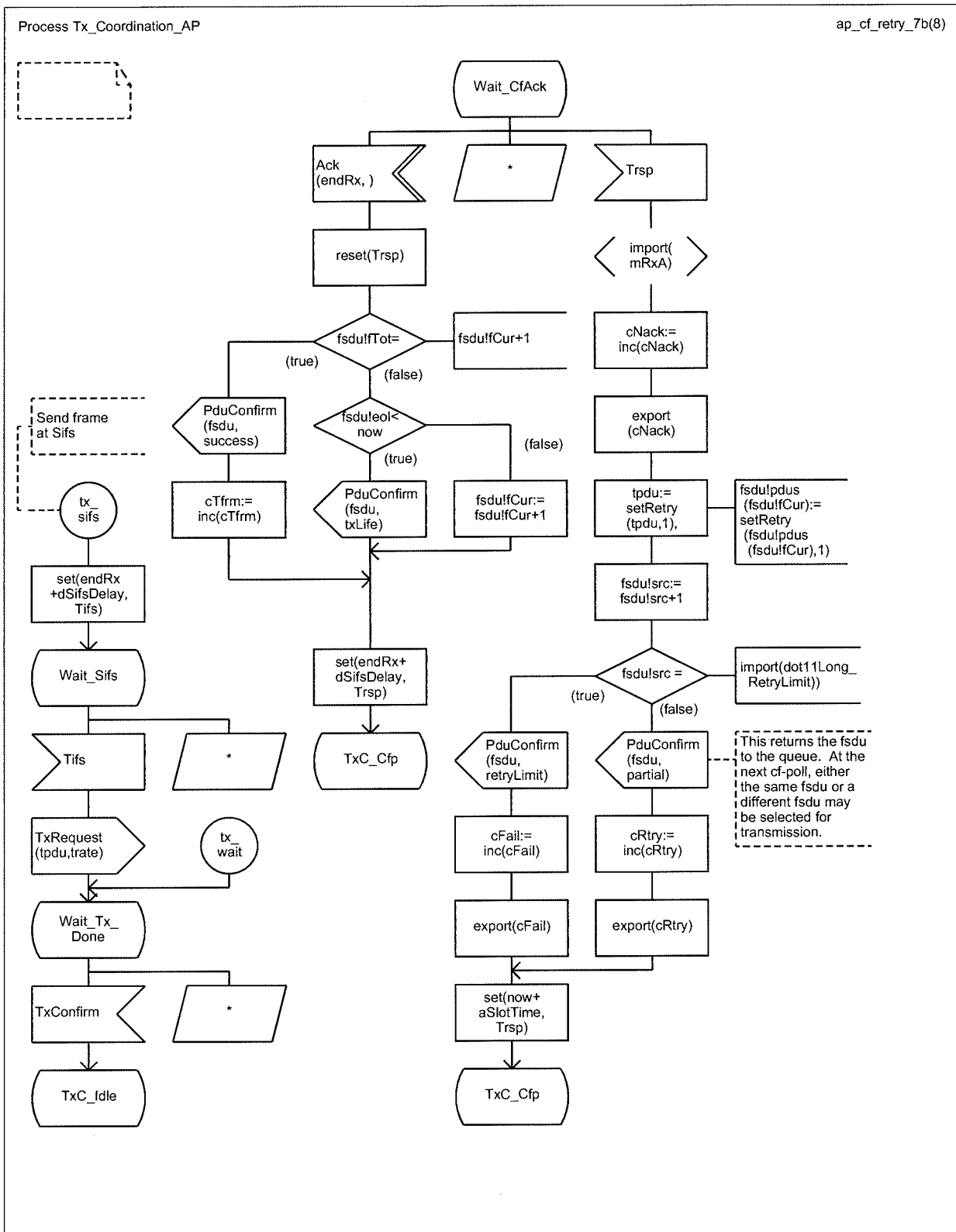


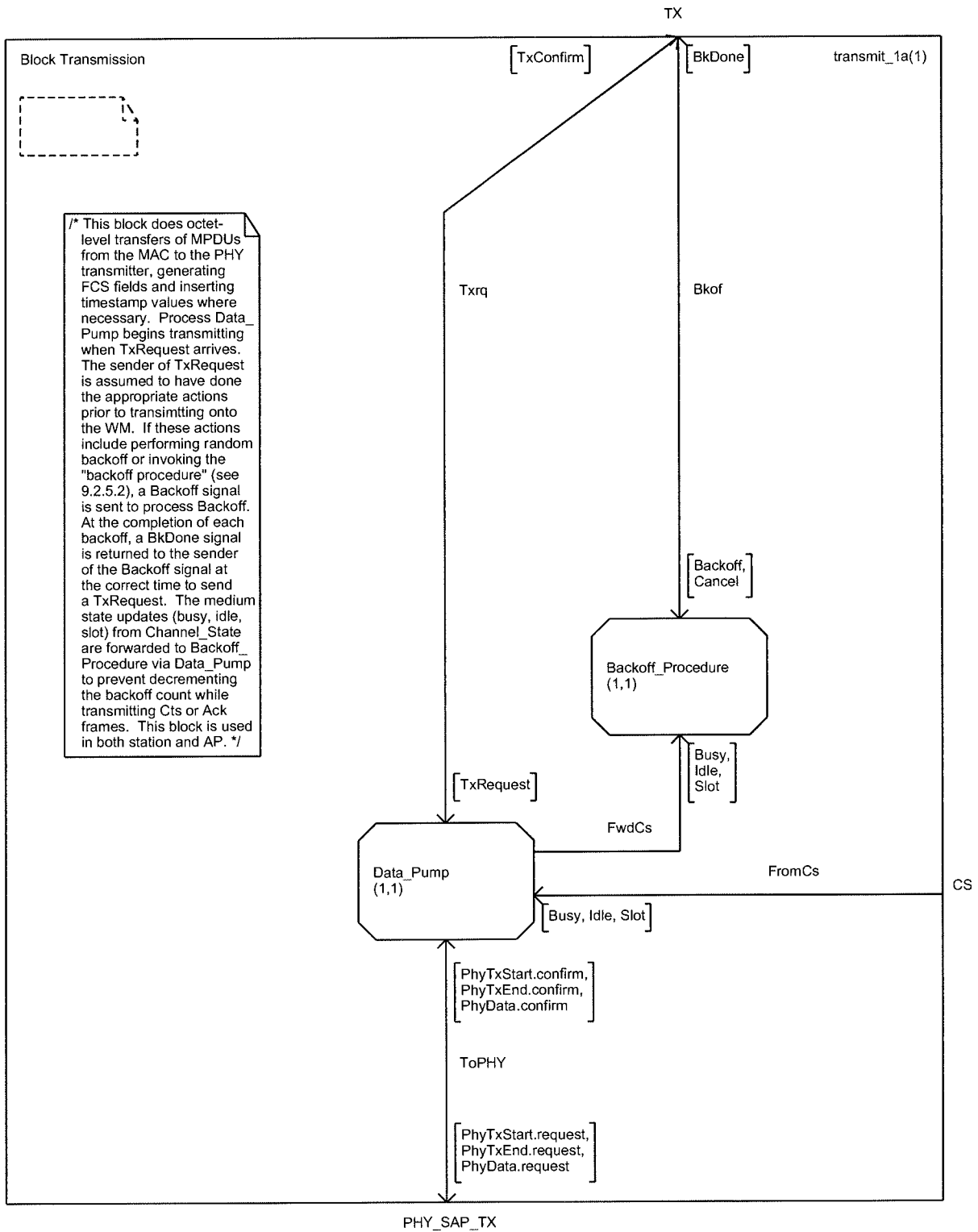


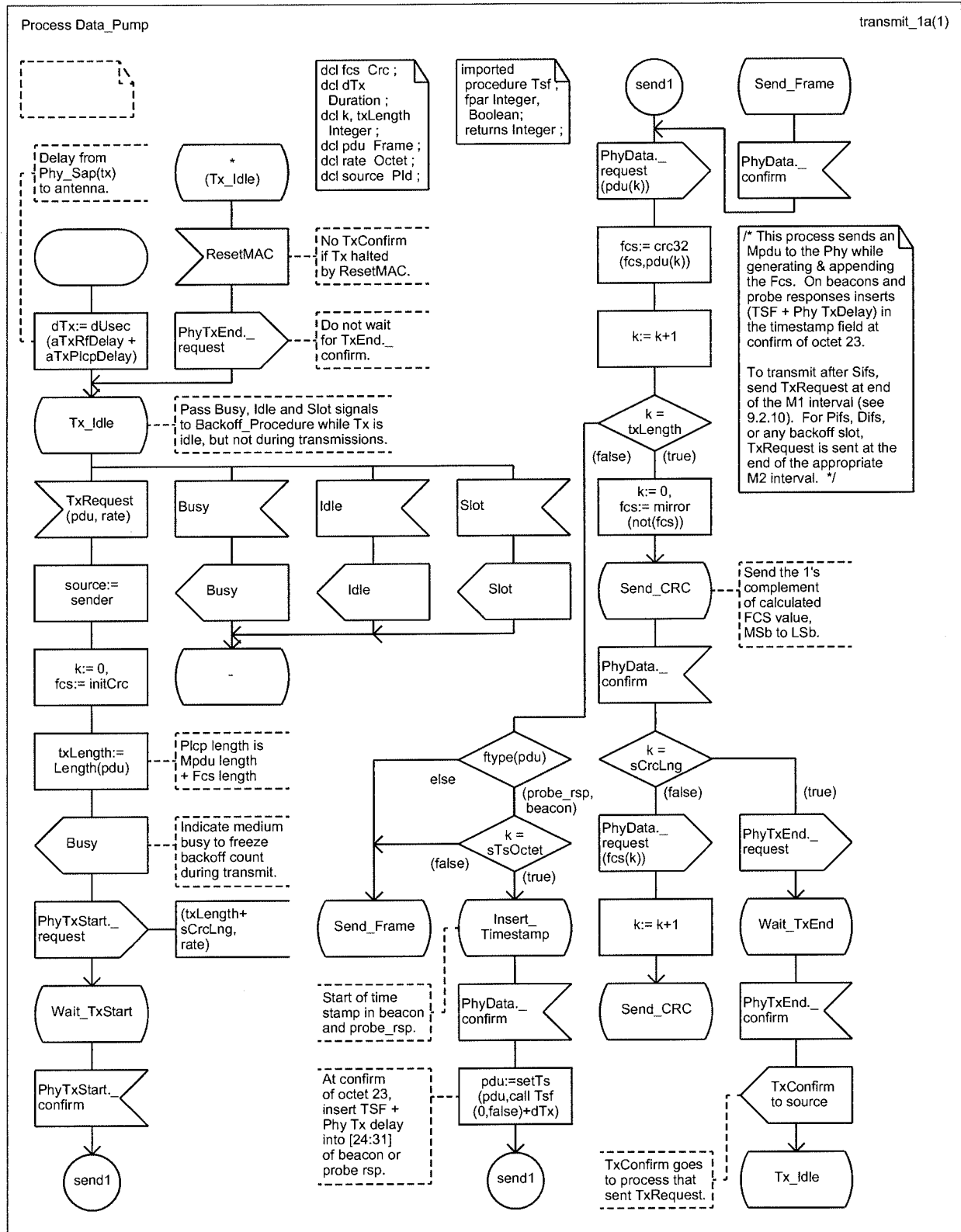


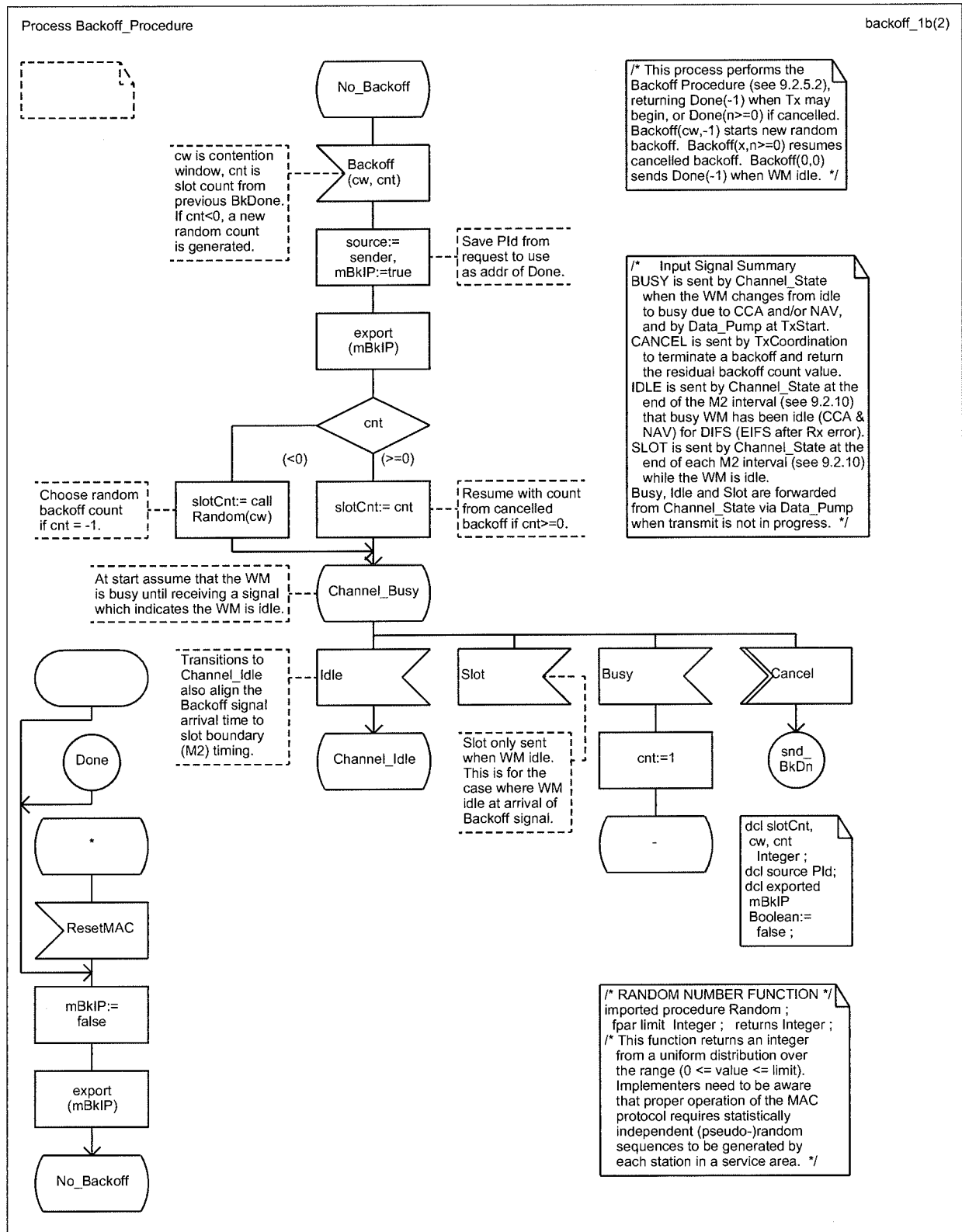


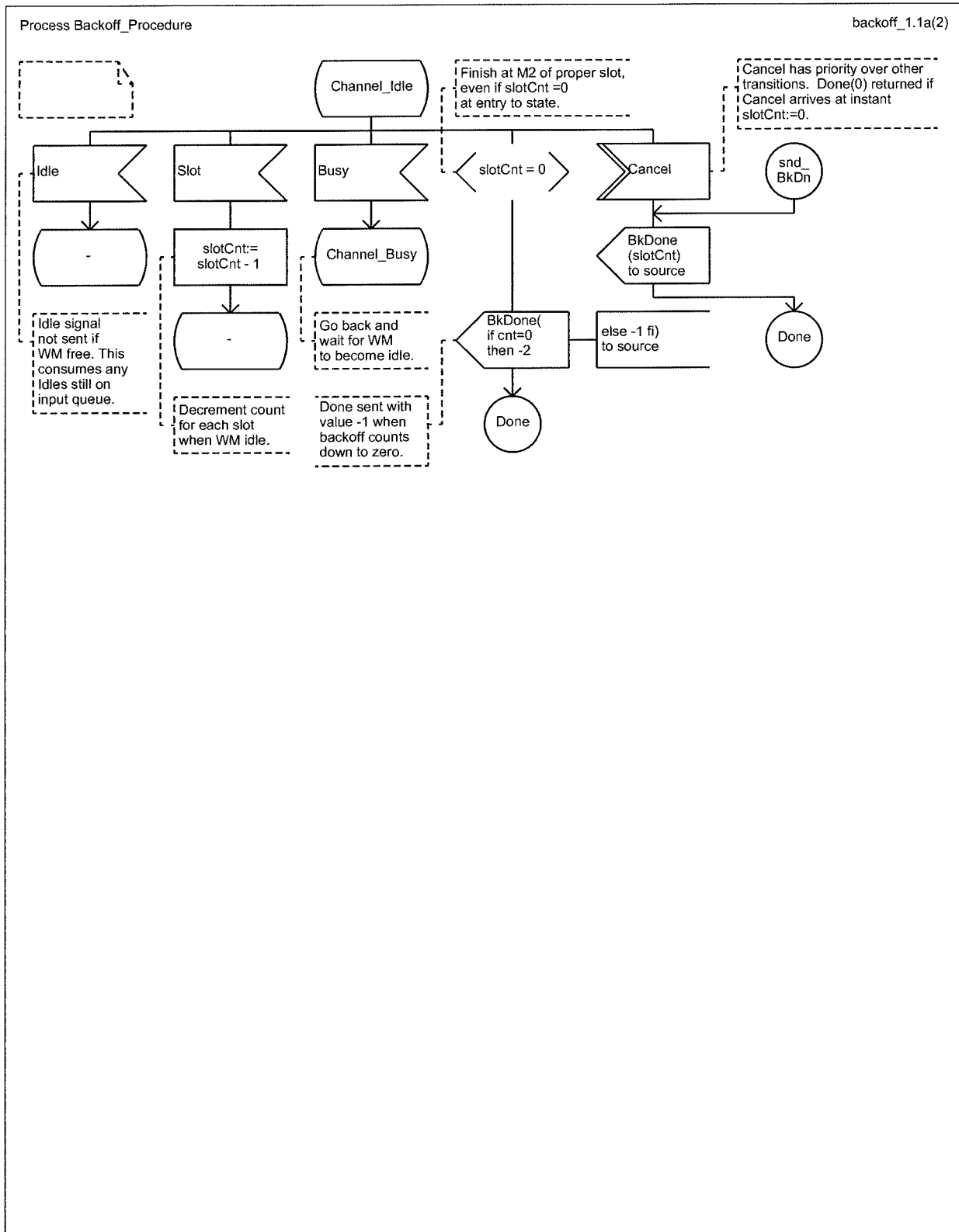


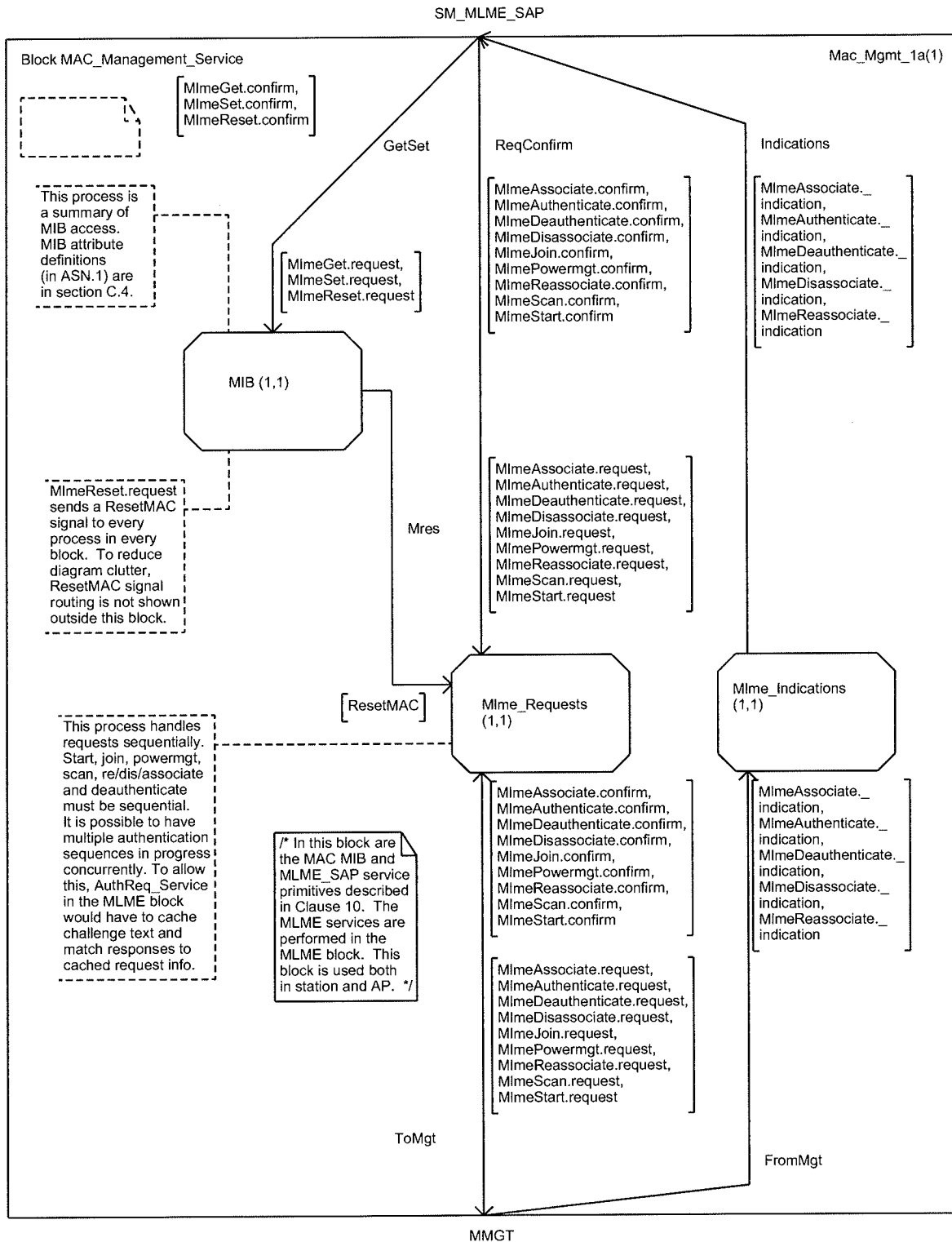


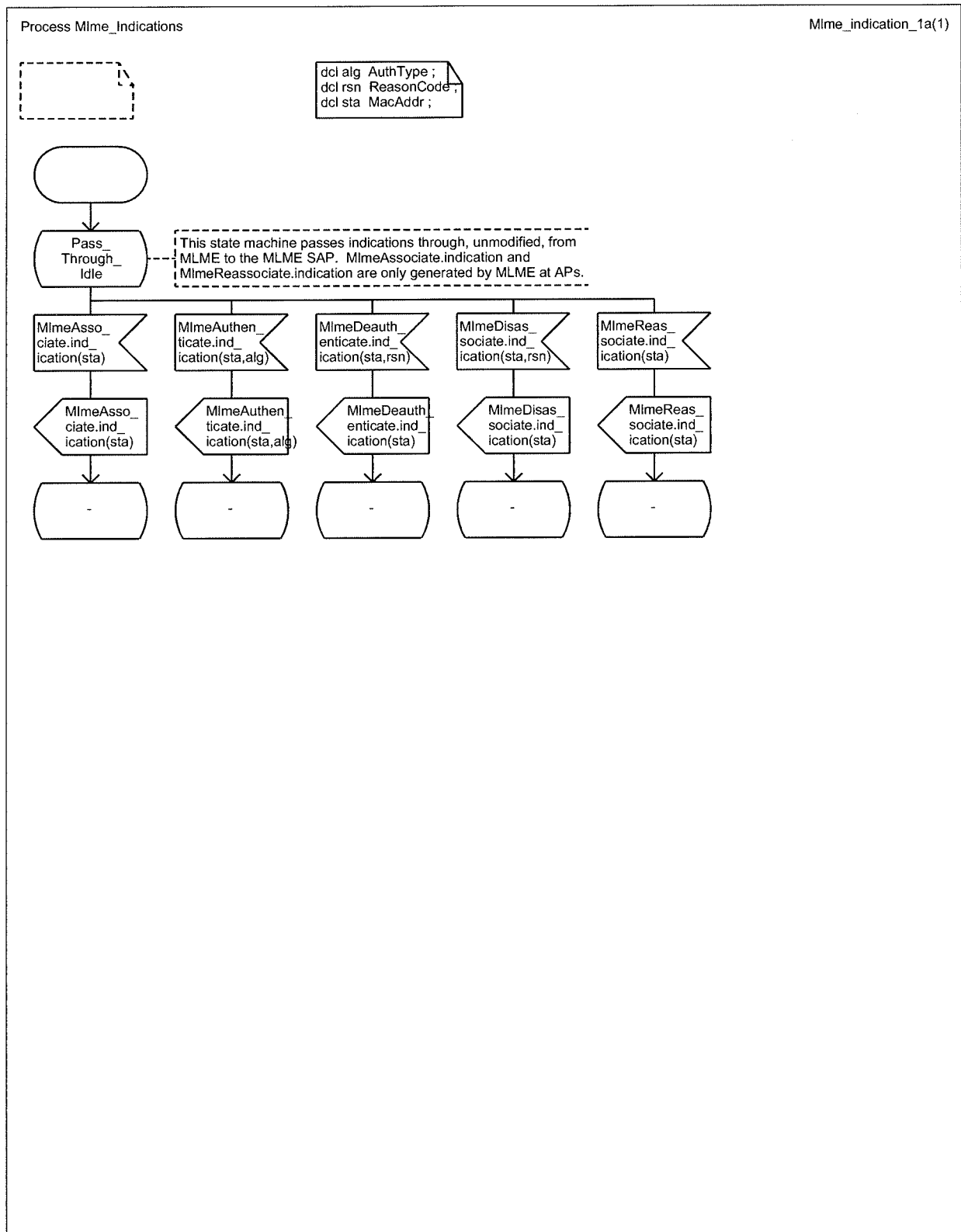


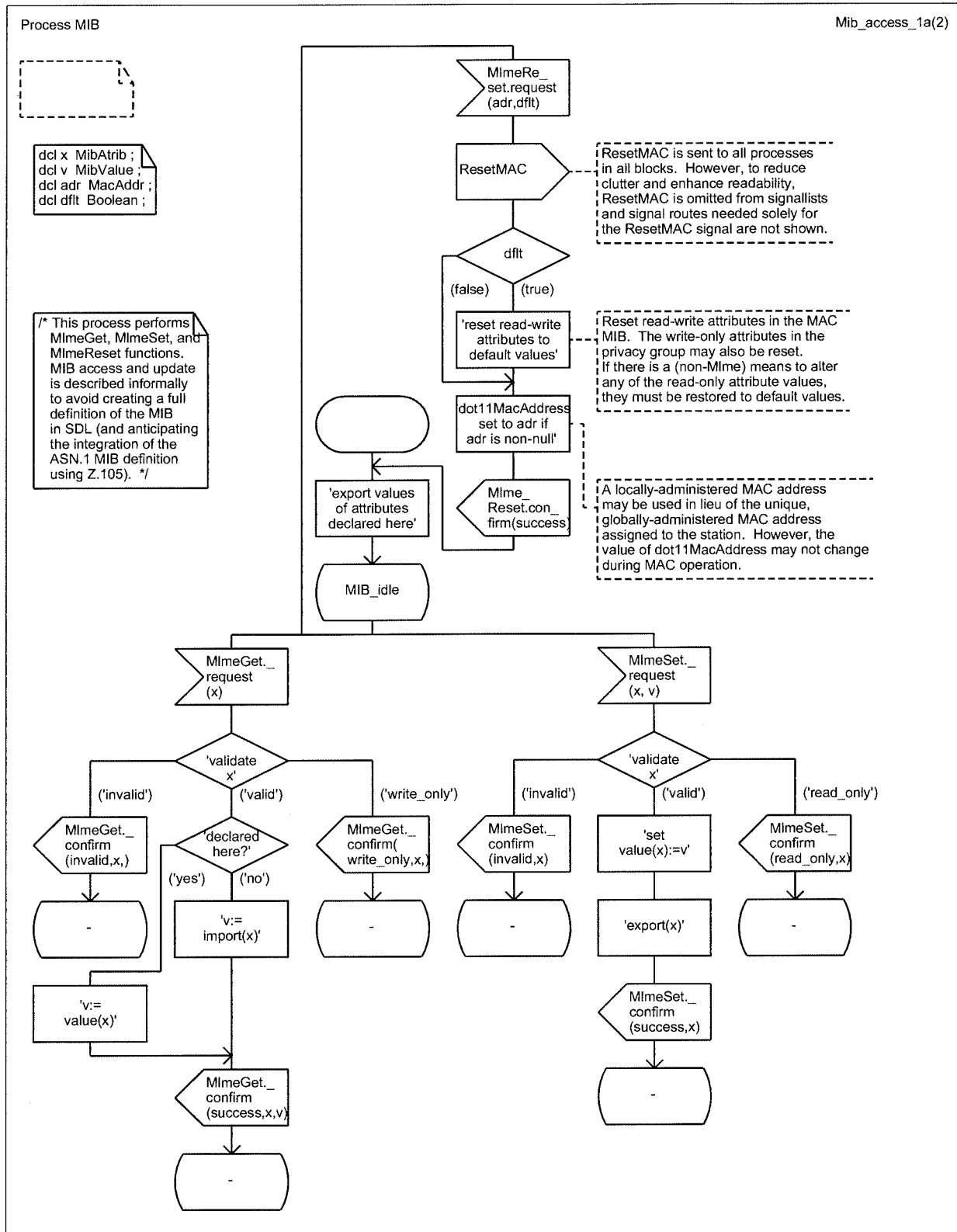












Process MIB

Mib_import_export_2b(2)



```

/* Import of {Read-Only} MIB counter
values exported from other processes */
imported
dot11AckFailureCount,
dot11FailedCount,
dot11FcsErrorCount,
dot11FrameDuplicateCount,
dot11MulticastReceivedFrameCount,
dot11MulticastTransmittedFrameCount,
dot11MultipleRetryCount,
dot11ReceivedFragmentCount,
dot11RetryCount,
dot11RtsFailureCount,
dot11RtsSuccessCount,
dot11TransmittedFragmentCount,
dot11WepExcludedCount,
dot11WepIcvErrorCount,
dot11WepUndecryptableCount Counter32 ;

```

```

/* Declarations of MIB attributes exported from
this process */

/* Read-Write attributes */
dcl exported
dot11AuthenticationAlgorithms AuthTypeSet:=
incl(open_system, shared_key),
dot11ExcludeUnencrypted Boolean:= false,
dot11FragmentationThreshold Integer:= 2346,
dot11GroupAddresses MacAddrSet:= empty,
dot11LongRetryLimit Integer:= 4,
dot11MaxReceiveLifetime Kusec:= 512,
dot11MaxTransmitMsduLifetime Kusec:= 512,
dot11MediumOccupancyLimit Kusec:= 100,
dot11PrivacyInvoked Boolean:= false,
mReceiveDTIMs Boolean:= true,
dot11CfpPeriod Integer:= 1,
dot11CfpMaxDuration Kusec:= 200,
dot11AuthenticationResponseTimeout Kusec:= 512,
dot11RtsThreshold Integer:= 3000,
dot11ShortRetryLimit Integer:= 7,
dot11WepDefaultKeyId KeyIndex:= 0,
dot11CurrentChannelNumber Integer:= 0,
dot11CurrentSet Integer:= 0,
dot11CurrentPattern Integer:= 0,
dot11CurrentIndex Integer:= 0 ;

/* Write-Only attributes */
dcl exported
dot11WepDefaultKeys KeyVector:= nullKey,
dot11WepKeyMappings
KeyMapArray:= (. nullAddr, false, nullKey .) ;

```

```

/* The following Read-Only attributes in the
MAC MIB are defined as synonyms (named
constants) rather than remote variables
because they describe properties of the
station which are static, at least during
any single instance of MAC operation:
dot11AuthenticationAlgorithms AuthTypeSet,
dot11CfpPollable Boolean,
dot11MacAddress MacAddr,
dot11ManufacturerID Octetstring,
dot11PrivacyOptionImplemented Boolean,
dot11ProductID Octetstring,
aStationID MacAddr,
dot11WepKeyMappingLength Integer ;

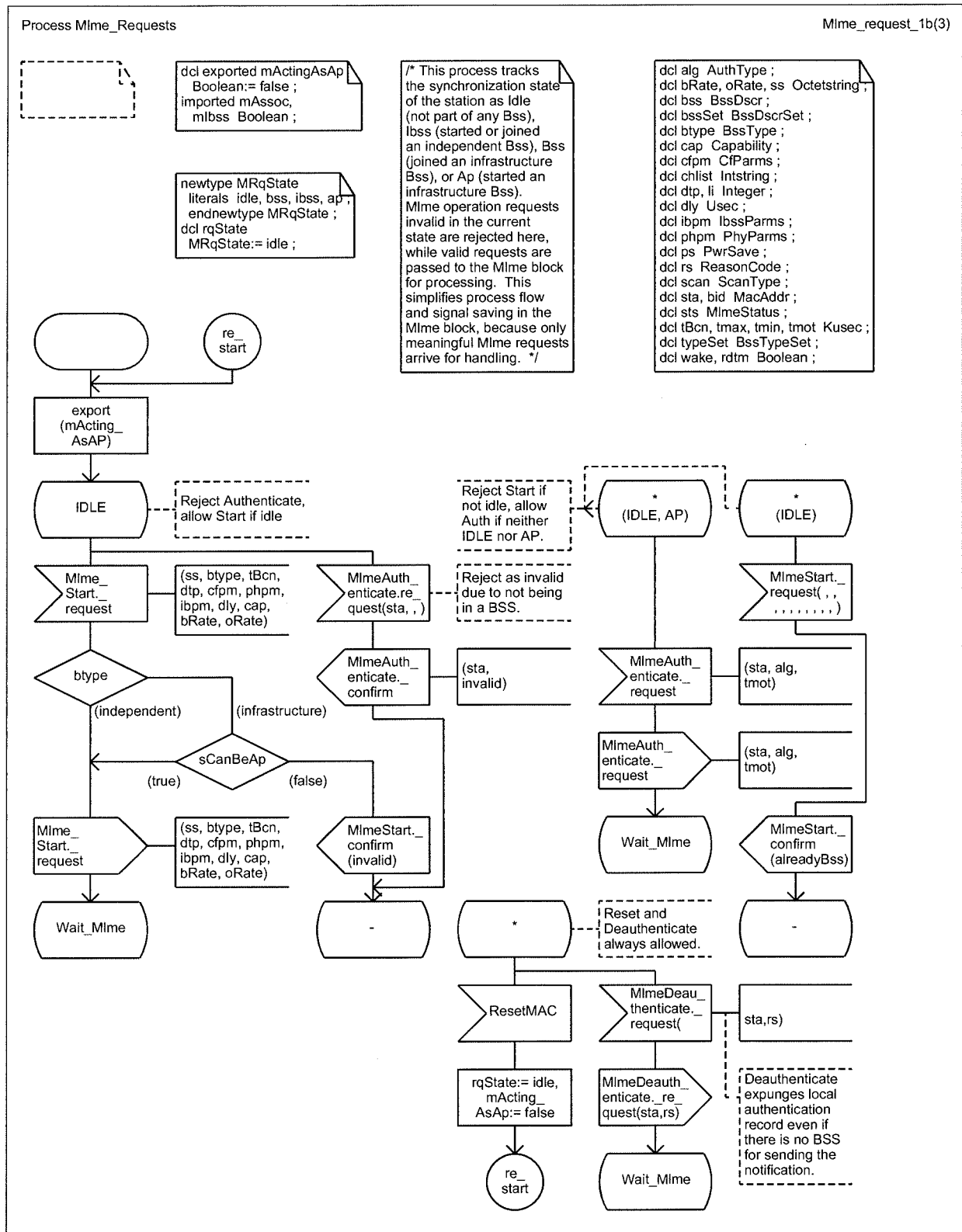
In addition, all Read-Only attributes in the
PHY MIB which are accessed by the MAC
are defined as synonyms.
*/

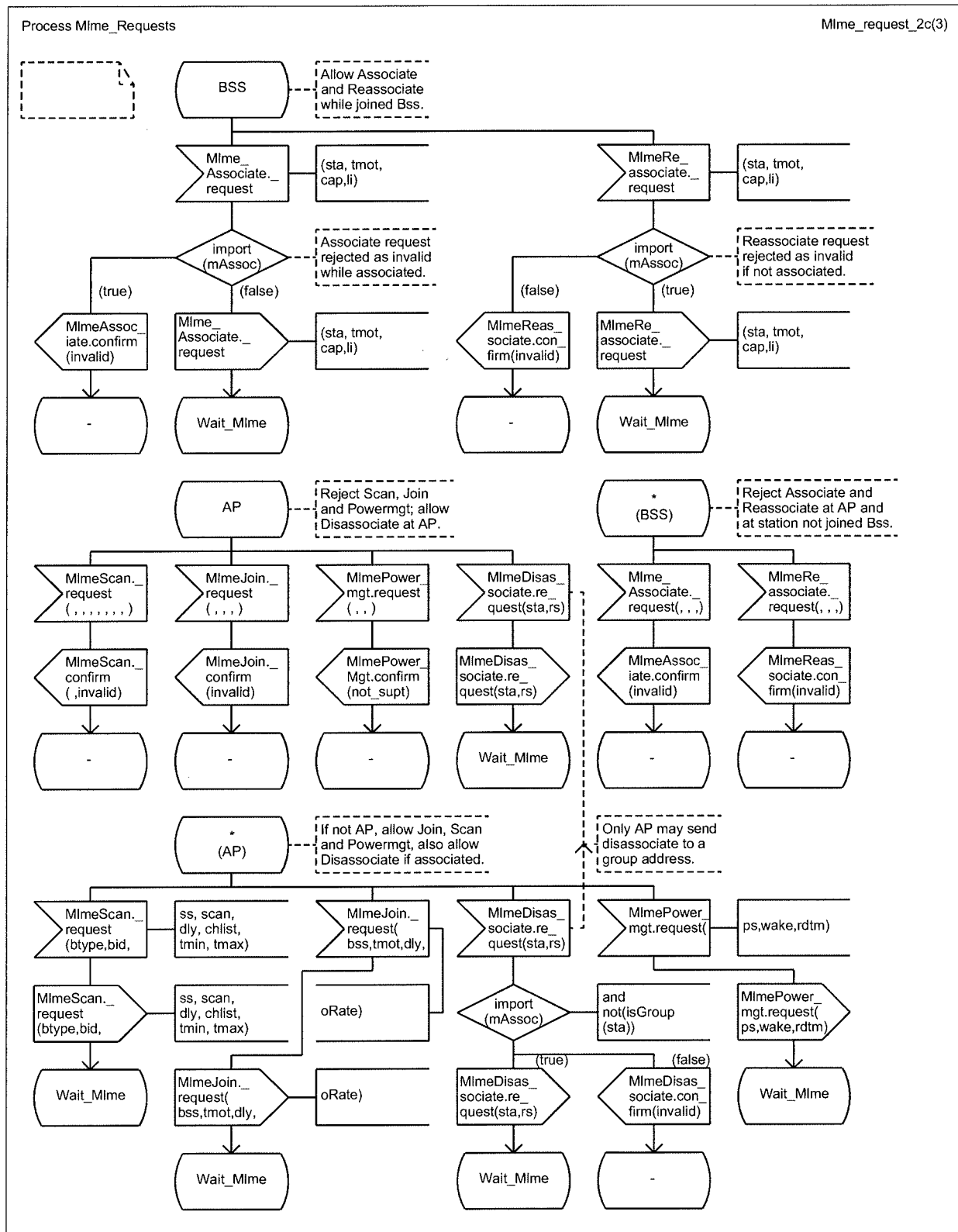
```

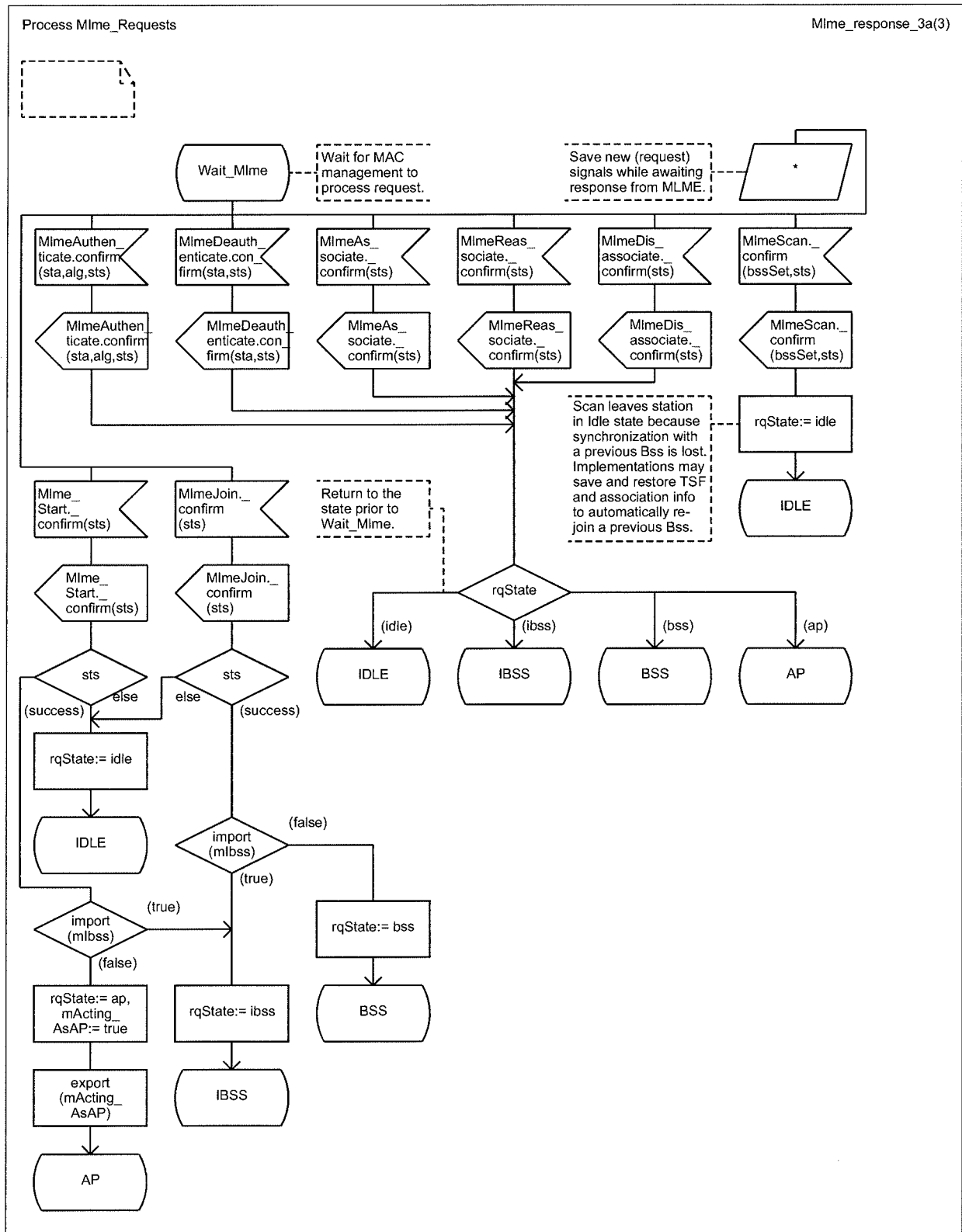
```

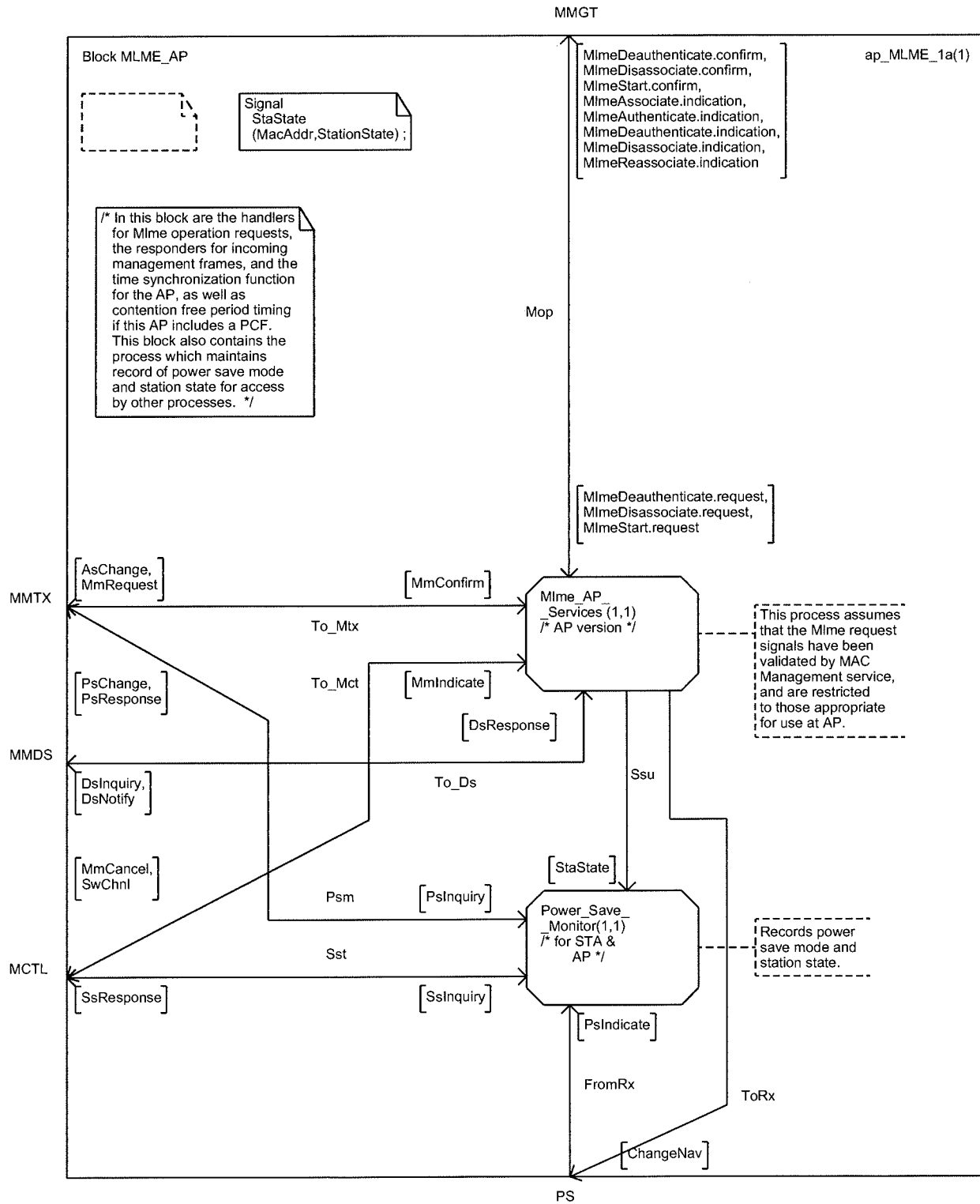
/* NOTE:
The values listed for MAC MIB attributes are the
specified default values for those attributes.
The values listed for PHY MIB attributes are either
the default values for the FH PHY, or arbitrary
values within the specified range. The specific
values for PHY attributes in this SDL description
of the MAC do not have normative significance.
*/

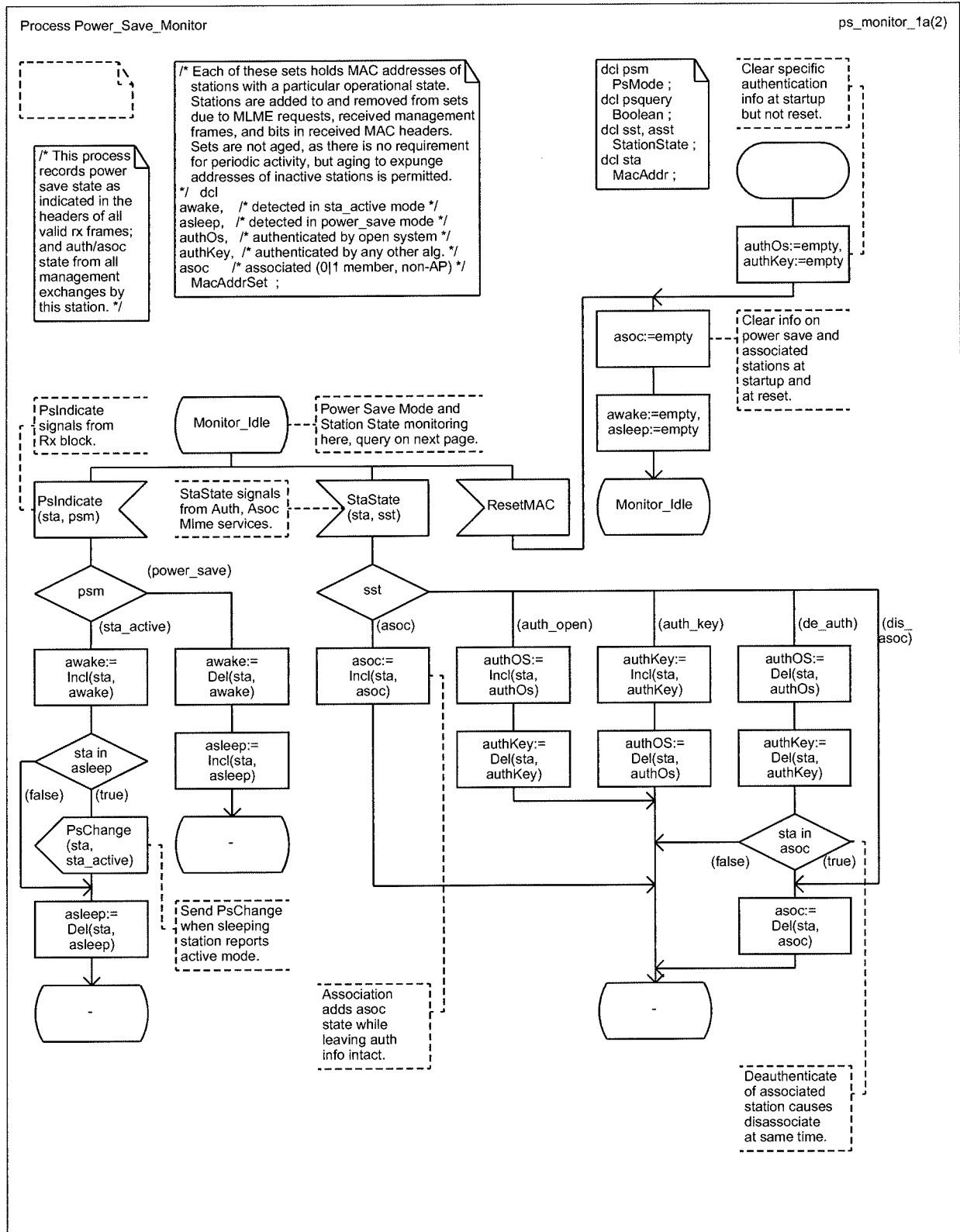
```

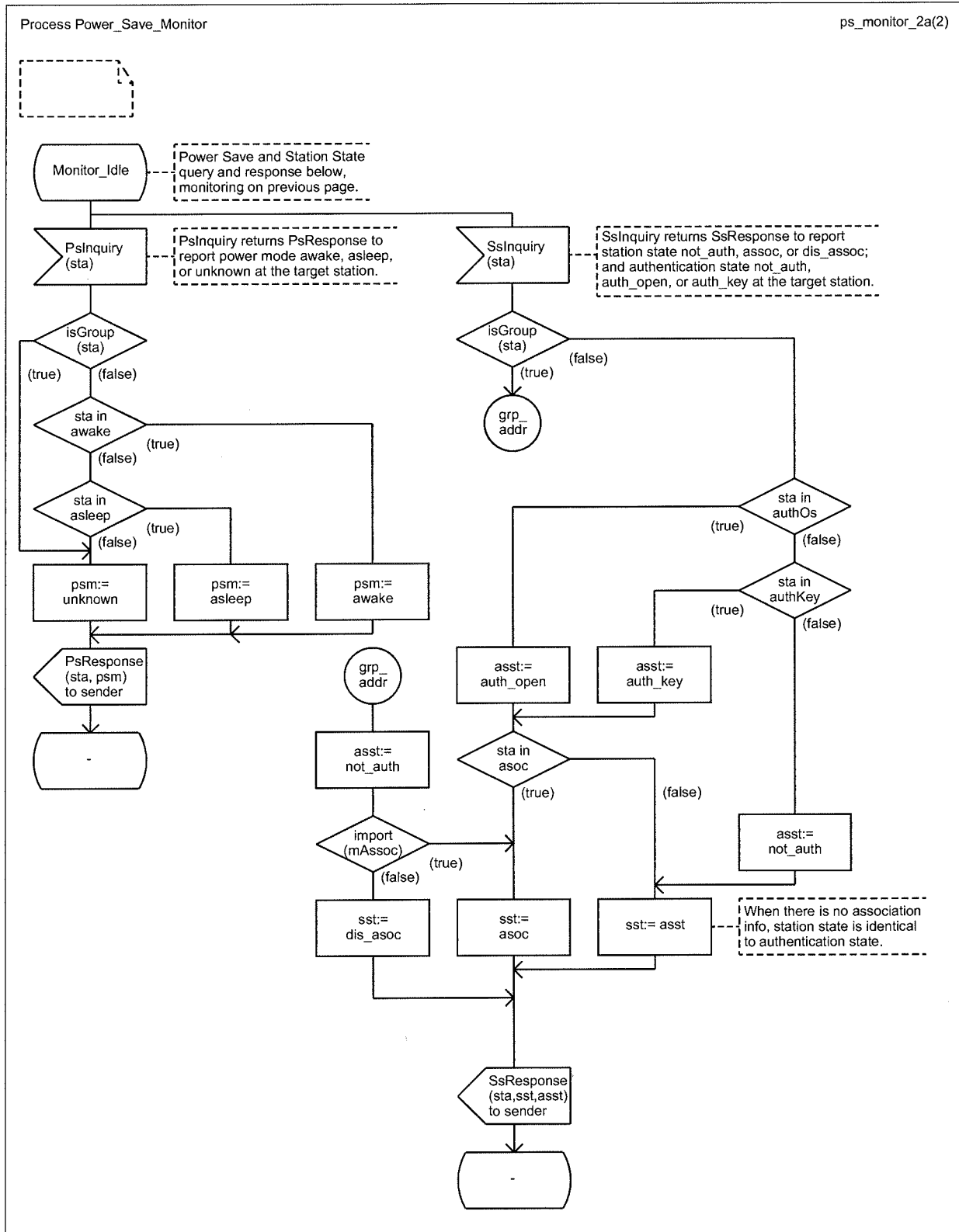


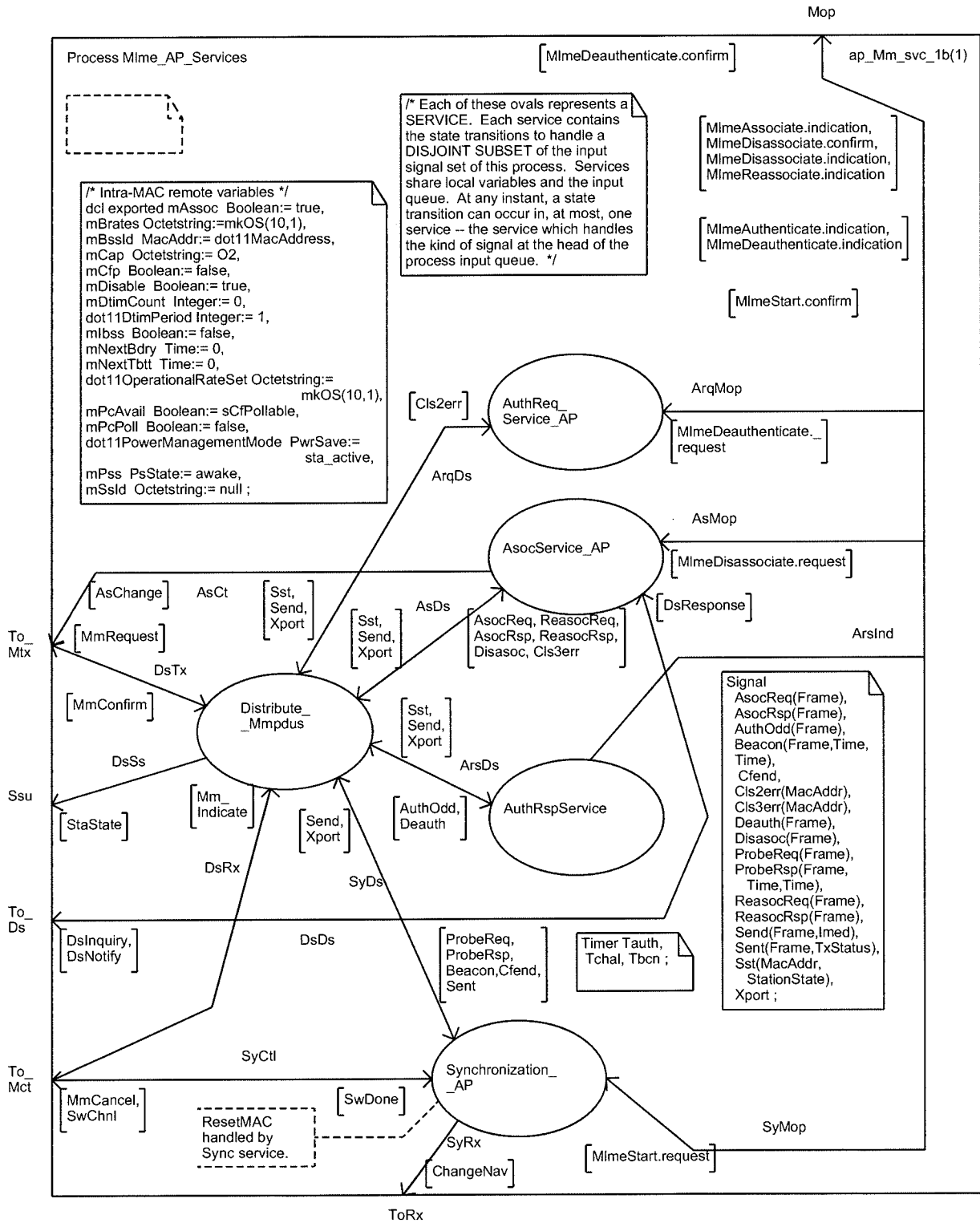


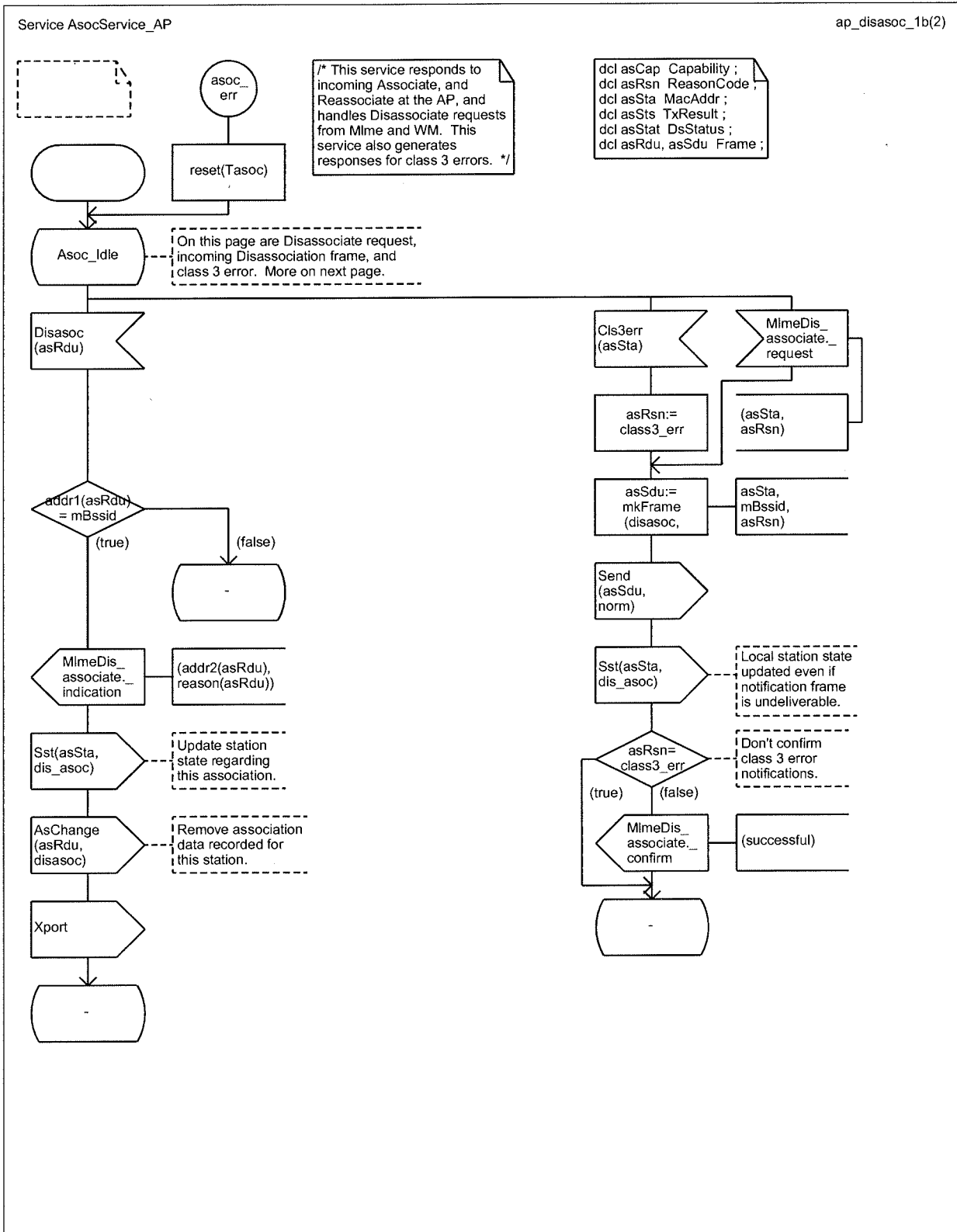


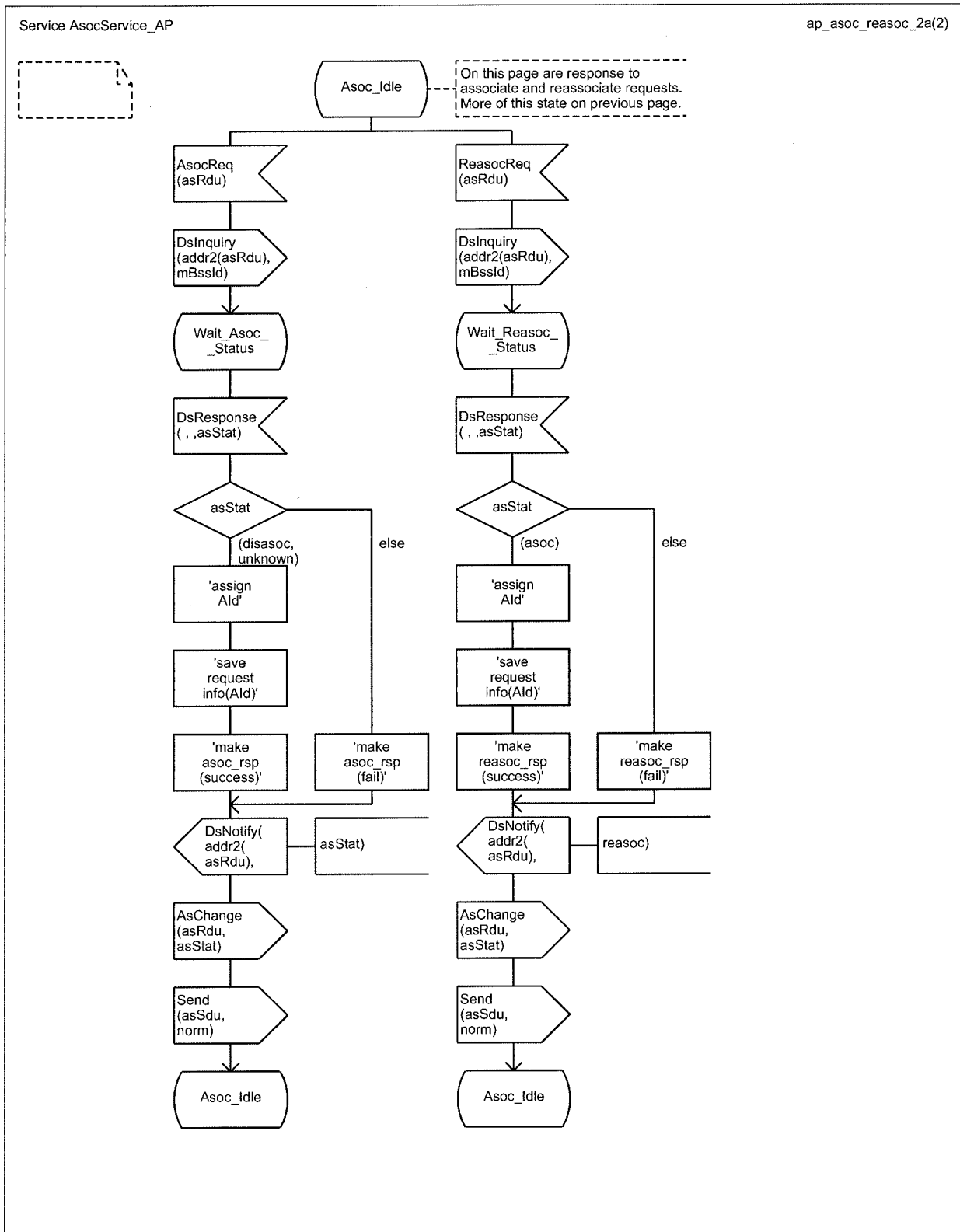


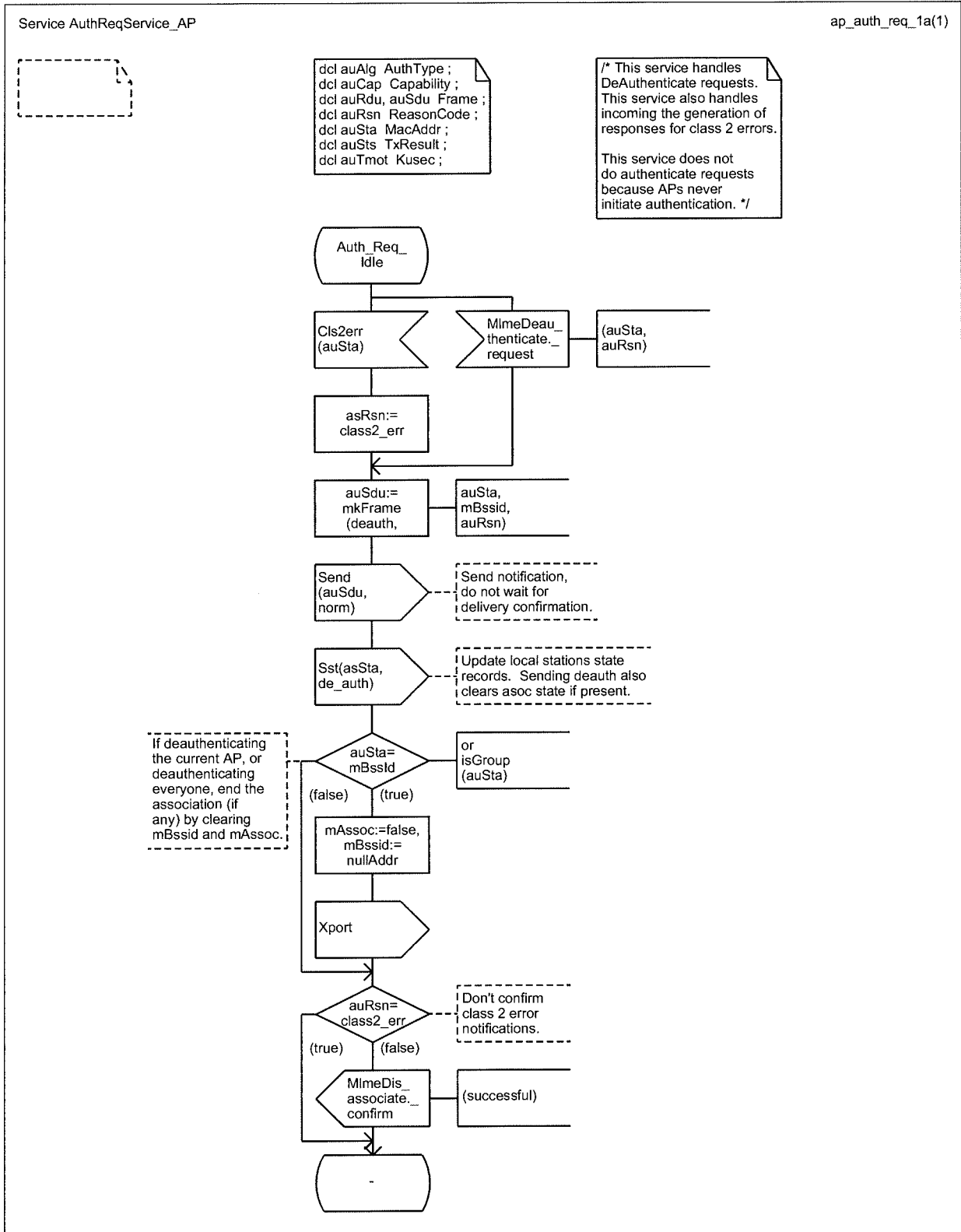


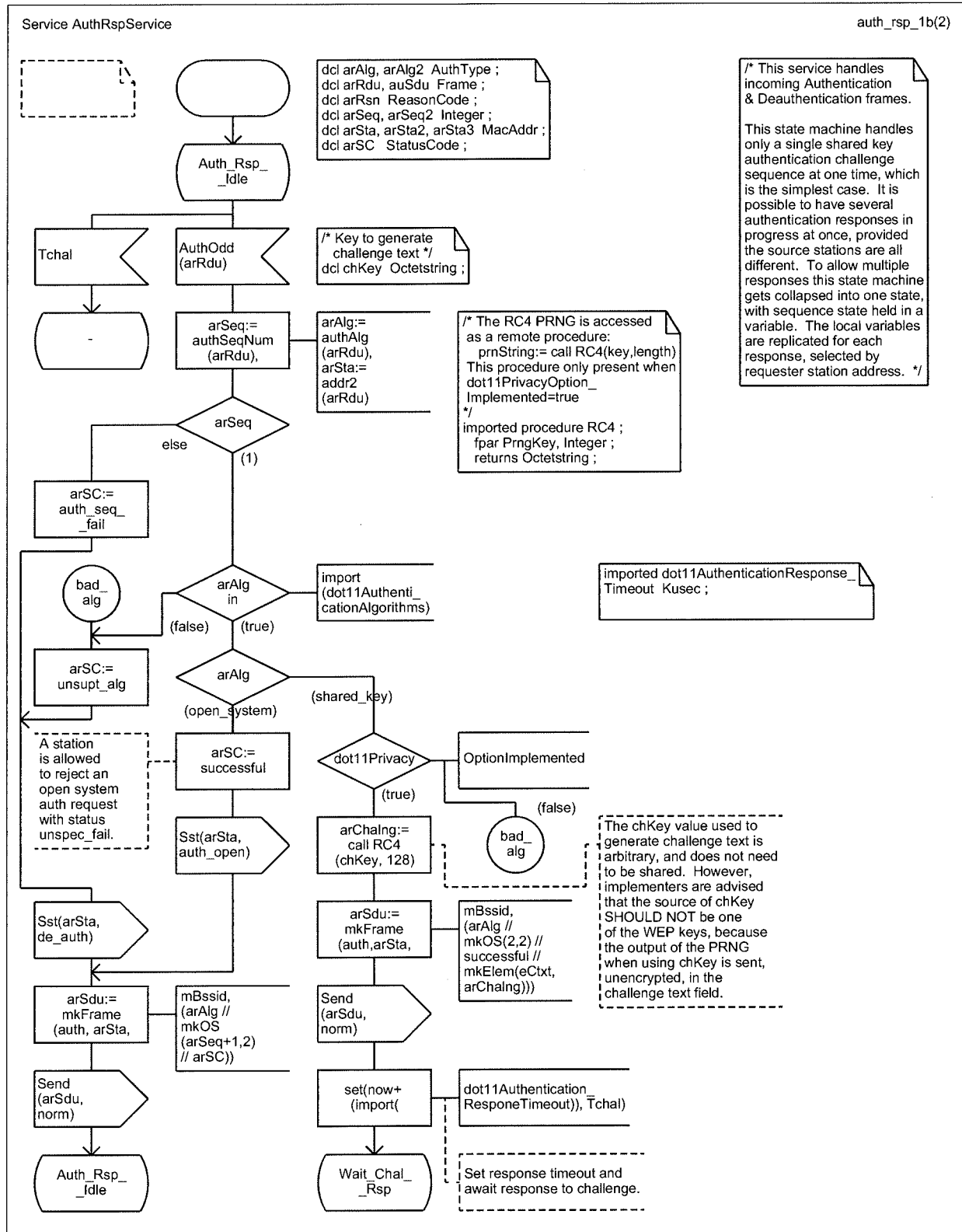


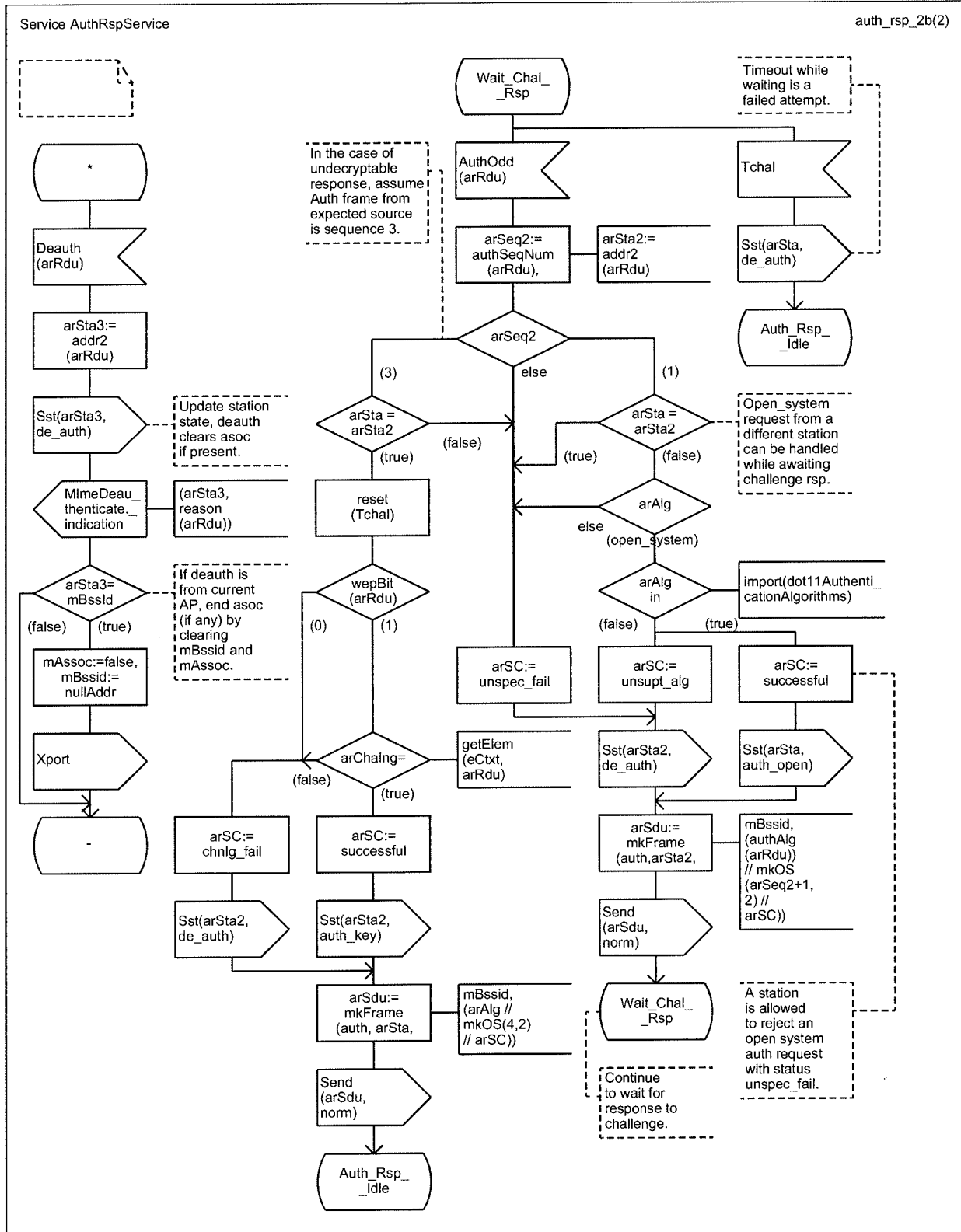


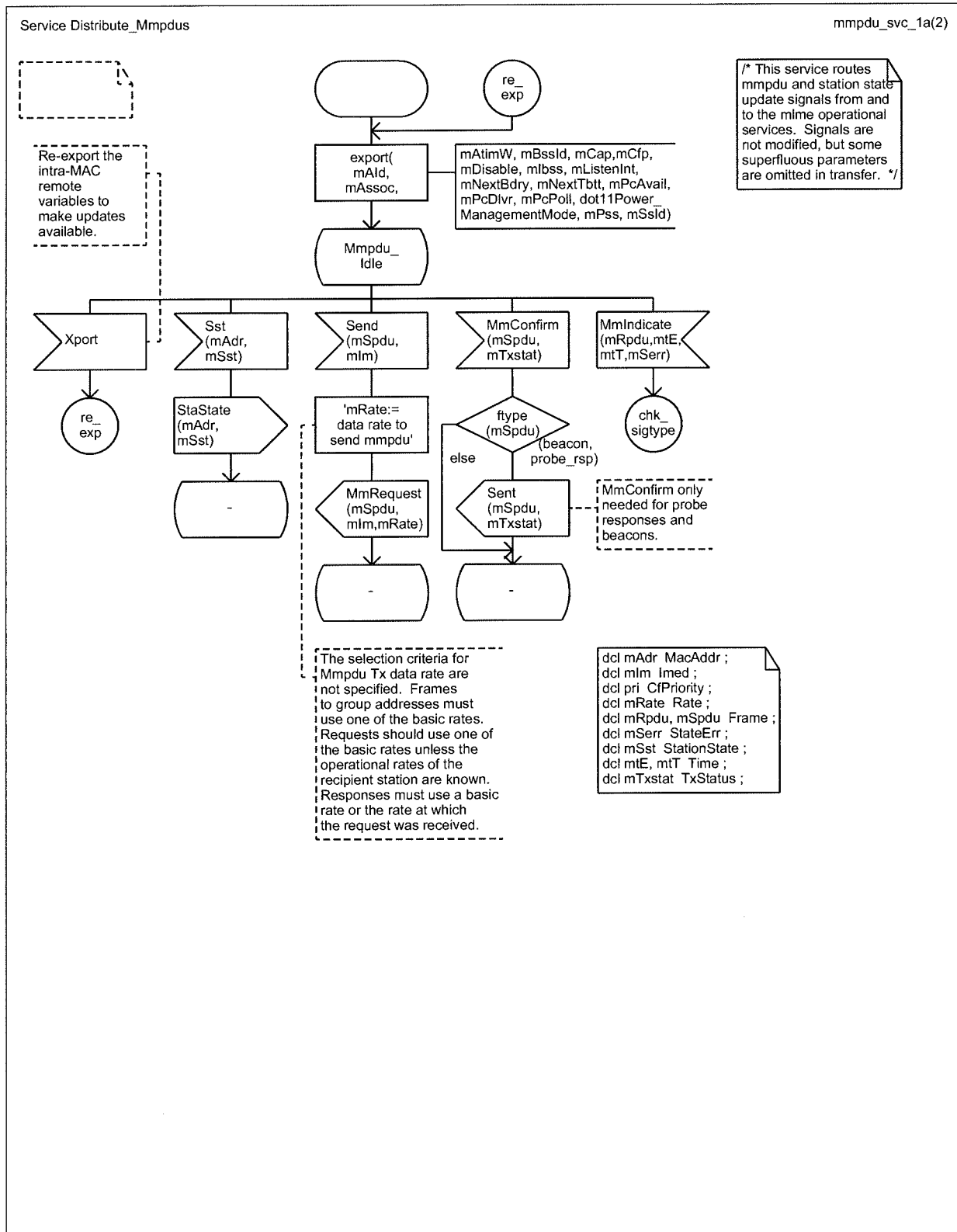


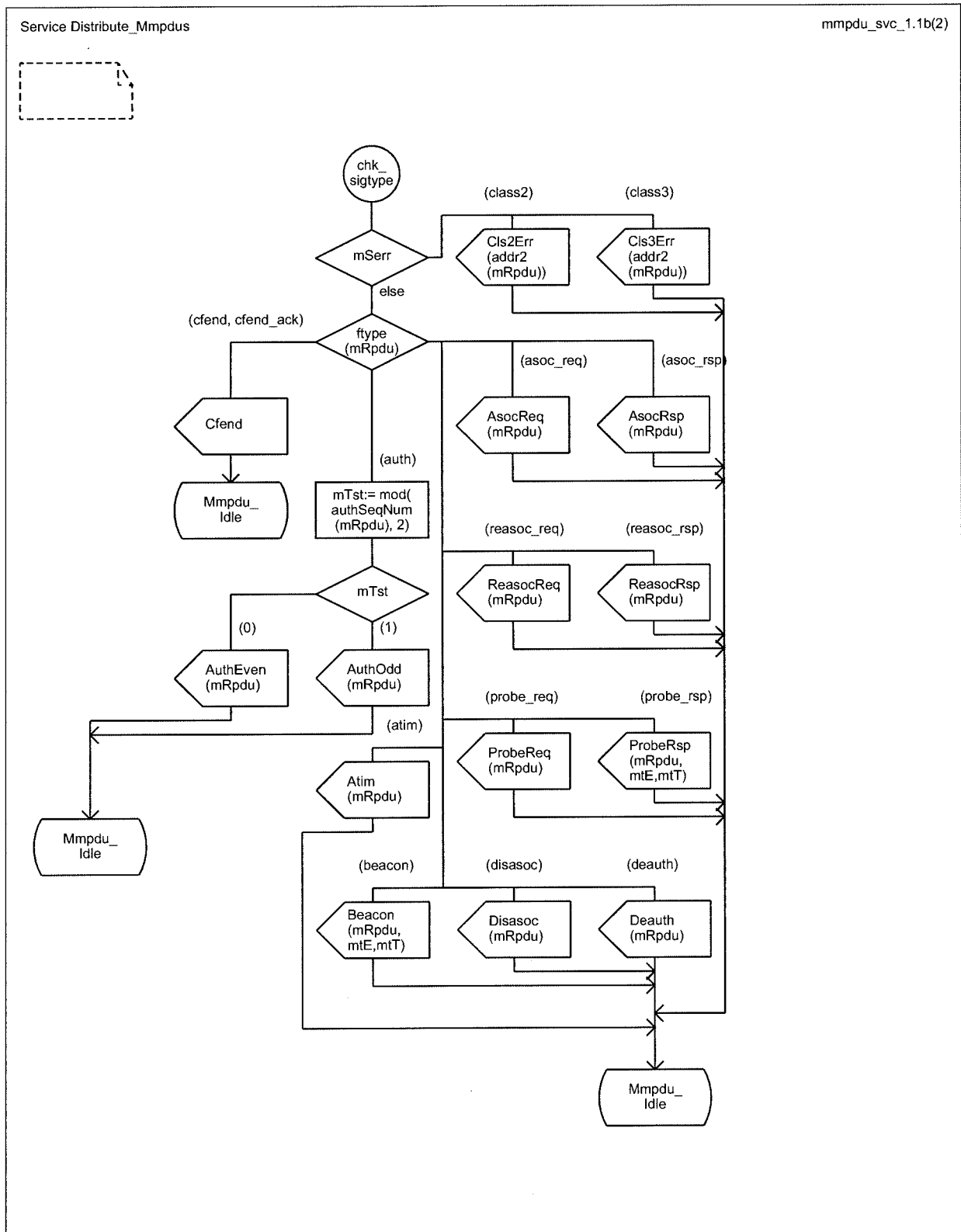












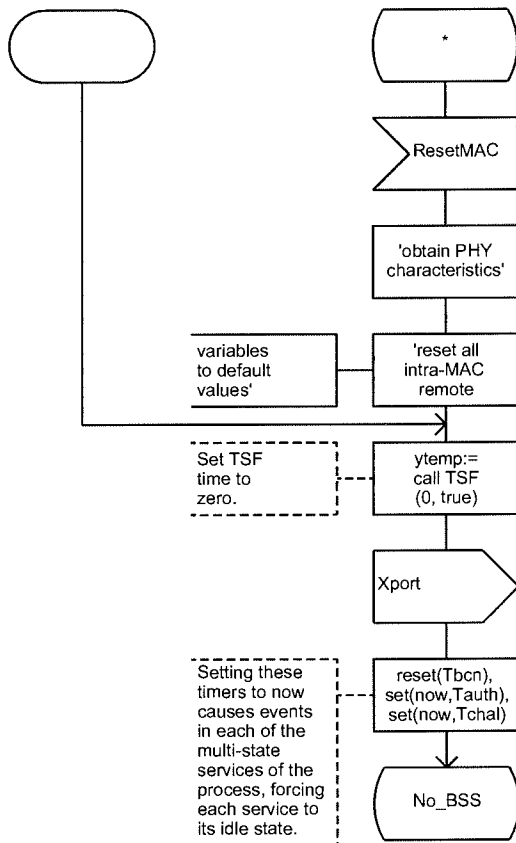
Service Synchronization_AP

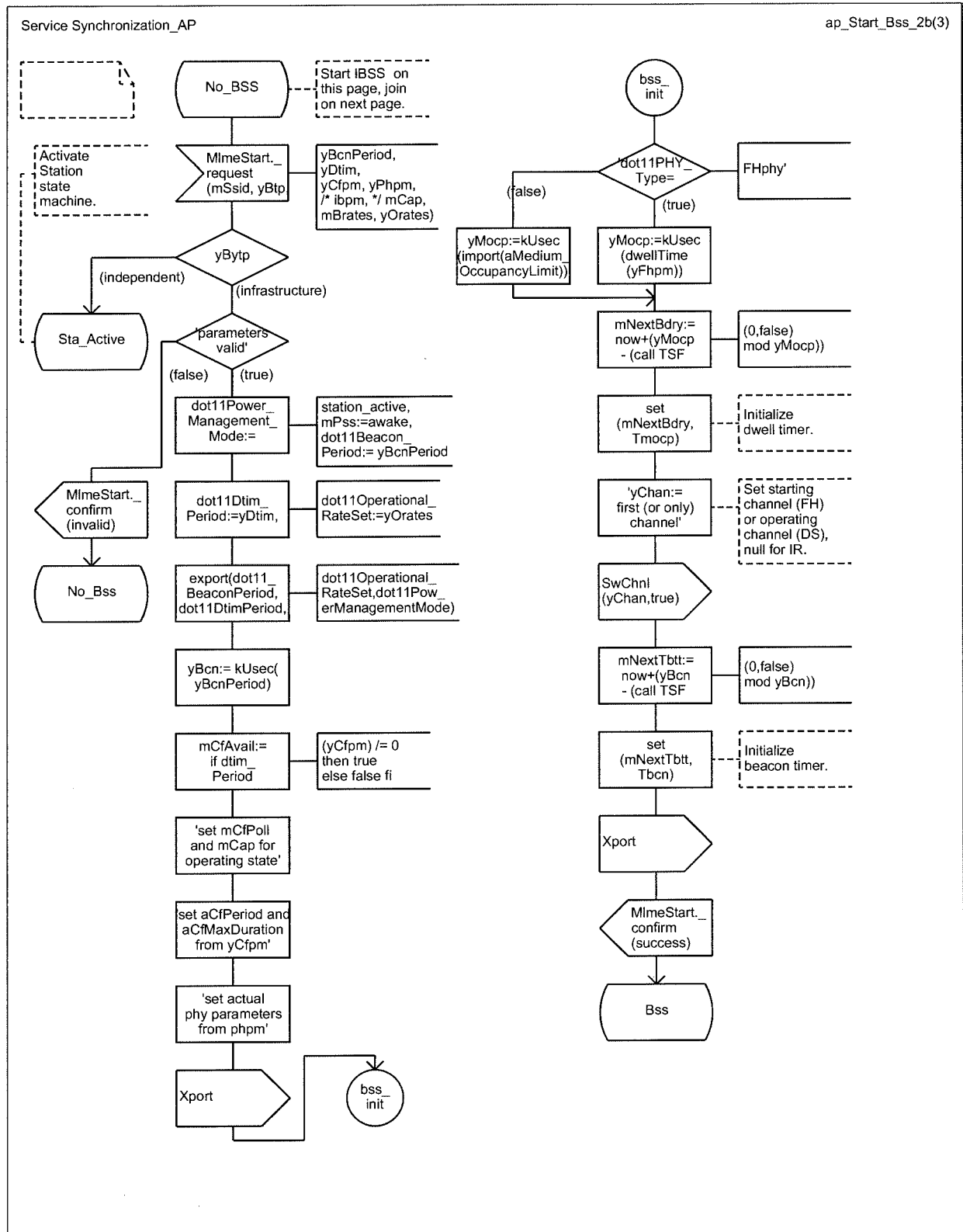
ap_Init_1b(3)

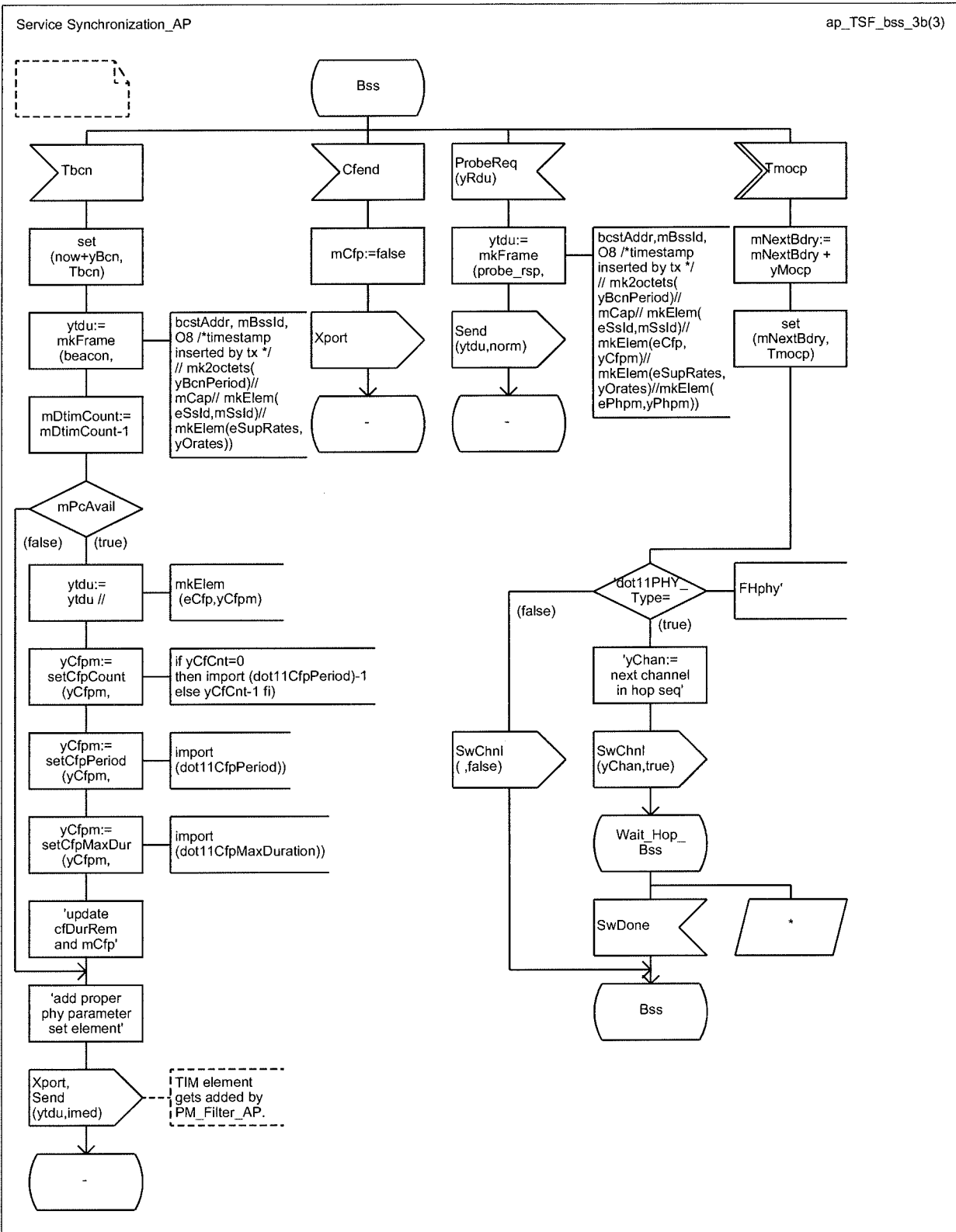
```
dcl yAtimRx, yPsm, yRdtim, yWake Boolean ;
dcl yAtw, yBcn, yMocp Time ;
dcl yBcnPeriod, yDtlim, yCmax, yCmin Kusec ;
dcl ybd BssDscr ;
dcl ybdset BssDscrSet ;
dcl ybtp BssType ;
dcl ybsid MacAddr ;
dcl yclist Intstring ;
dcl ycx, yJto, ytemp Integer ;
dcl yDspm DsParms ;
dcl yFhpm FhParms ;
dcl ylbpm lbssParms ;
dcl ypdly Usec ;
```

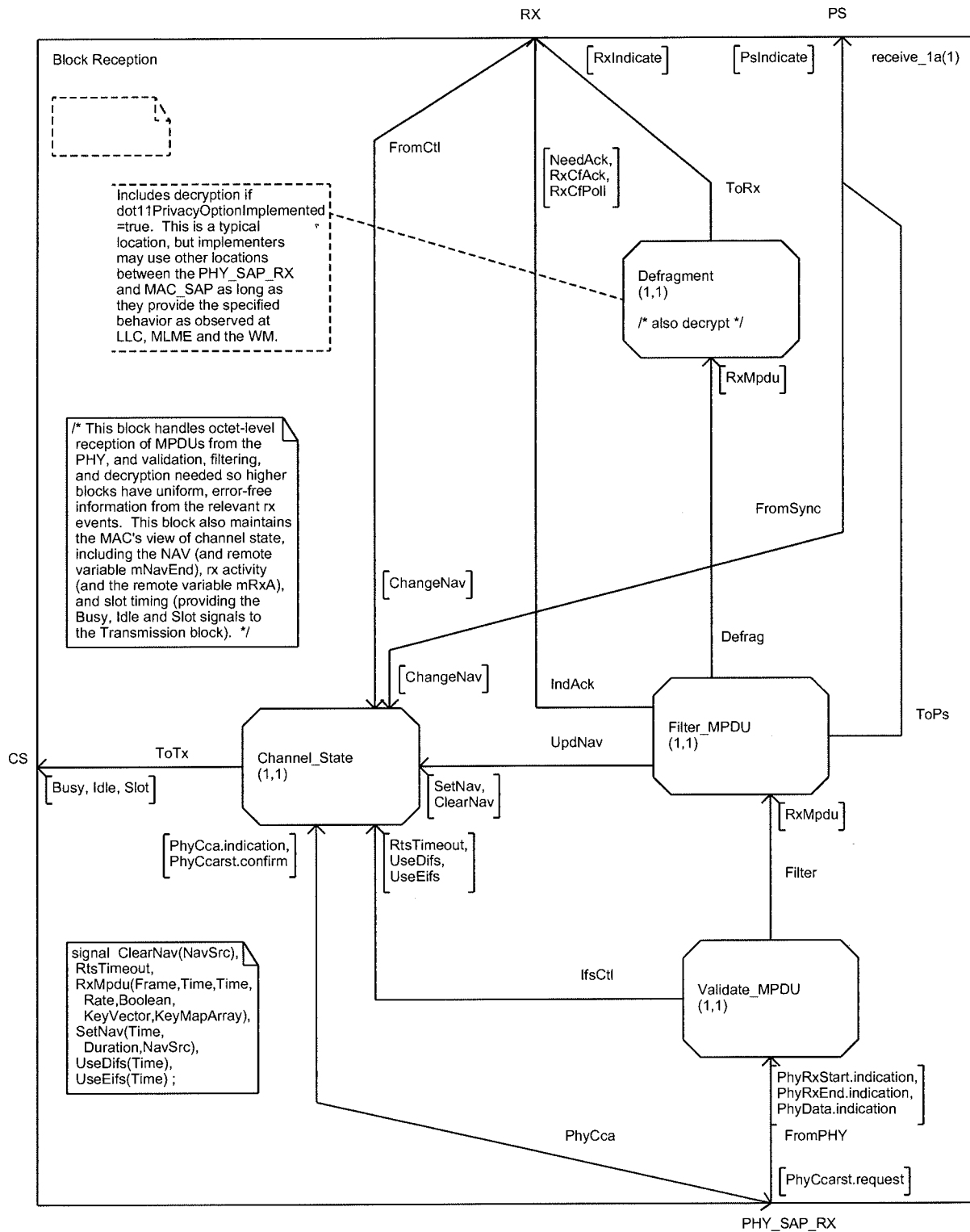
```
dcl yPhpm PhyParms ;
dcl yRdu, yTdu Frame ;
dcl yssid Octetstring ;
dcl yOrates Ratestring ;
dcl ystp ScanType ;
dcl ytrsl TxResult ;
```

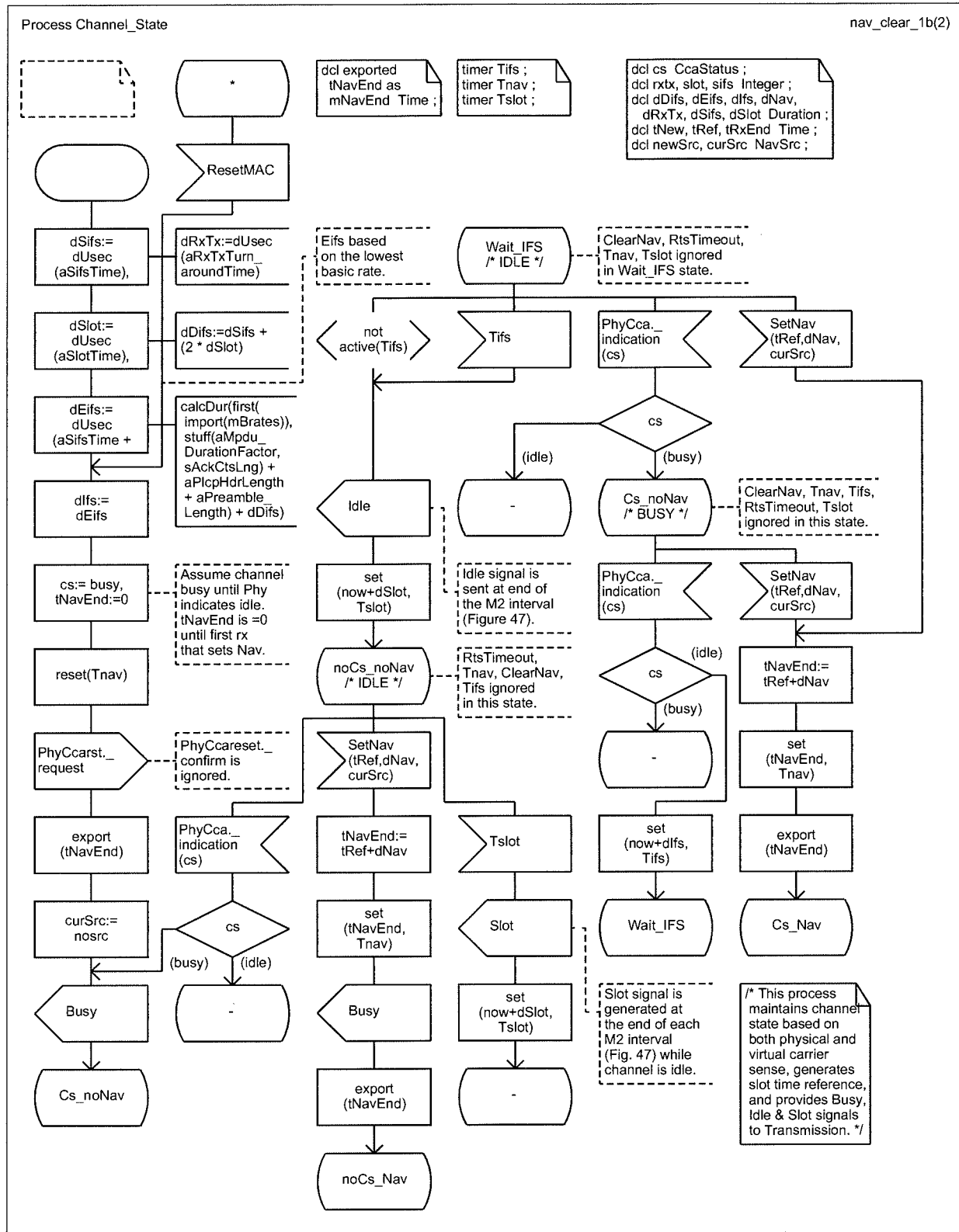
```
timer Tscan,
Tmocp ;
```

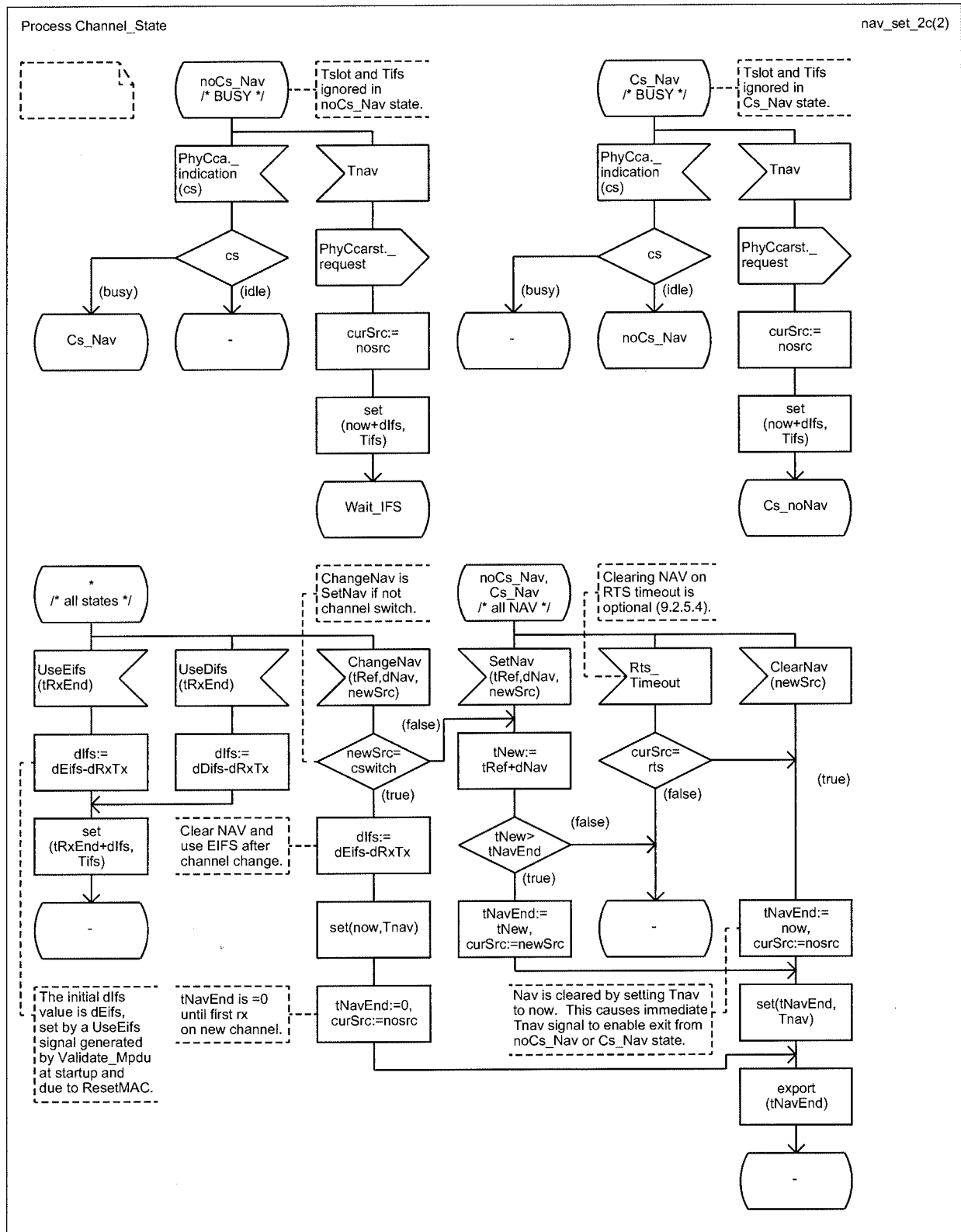


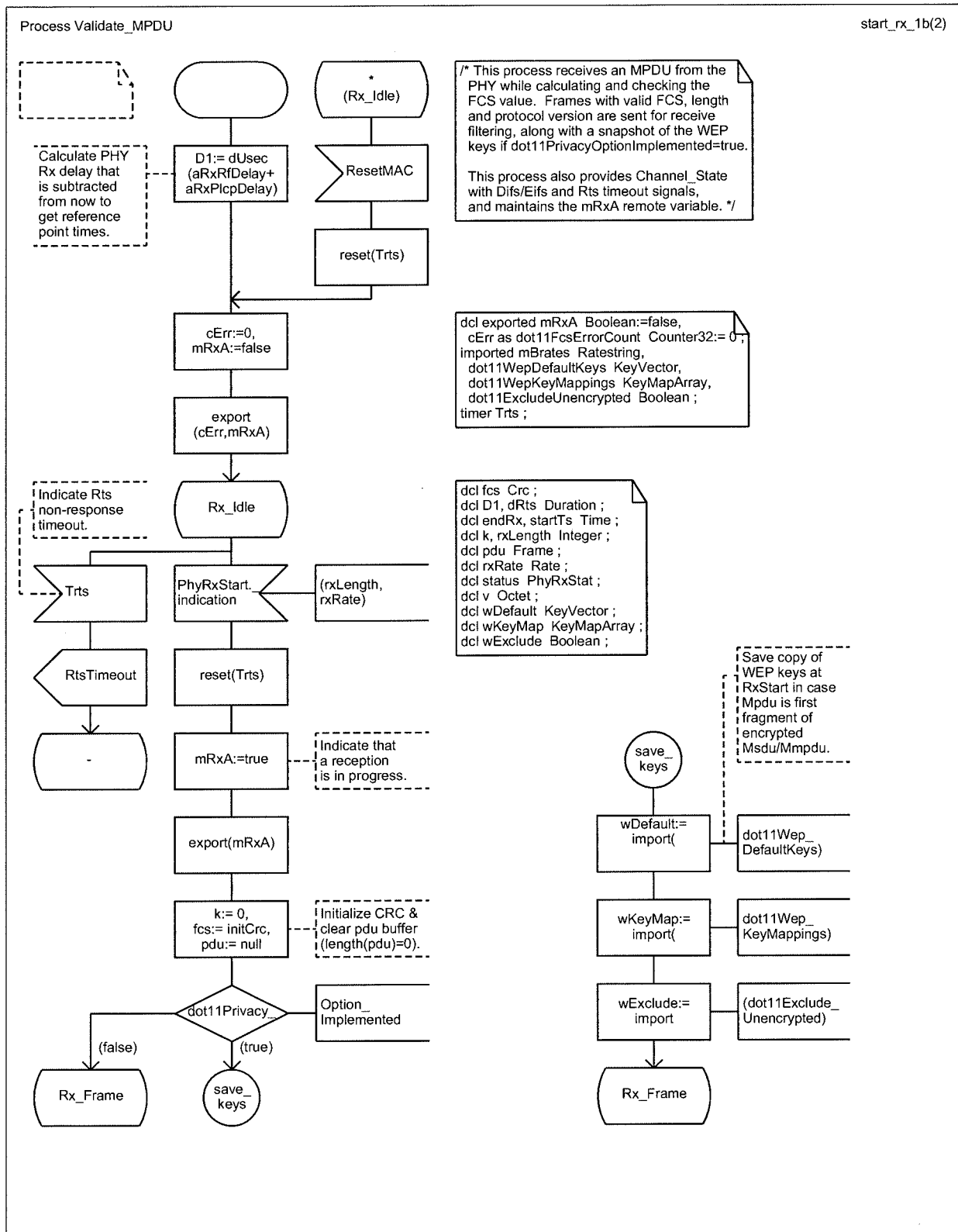


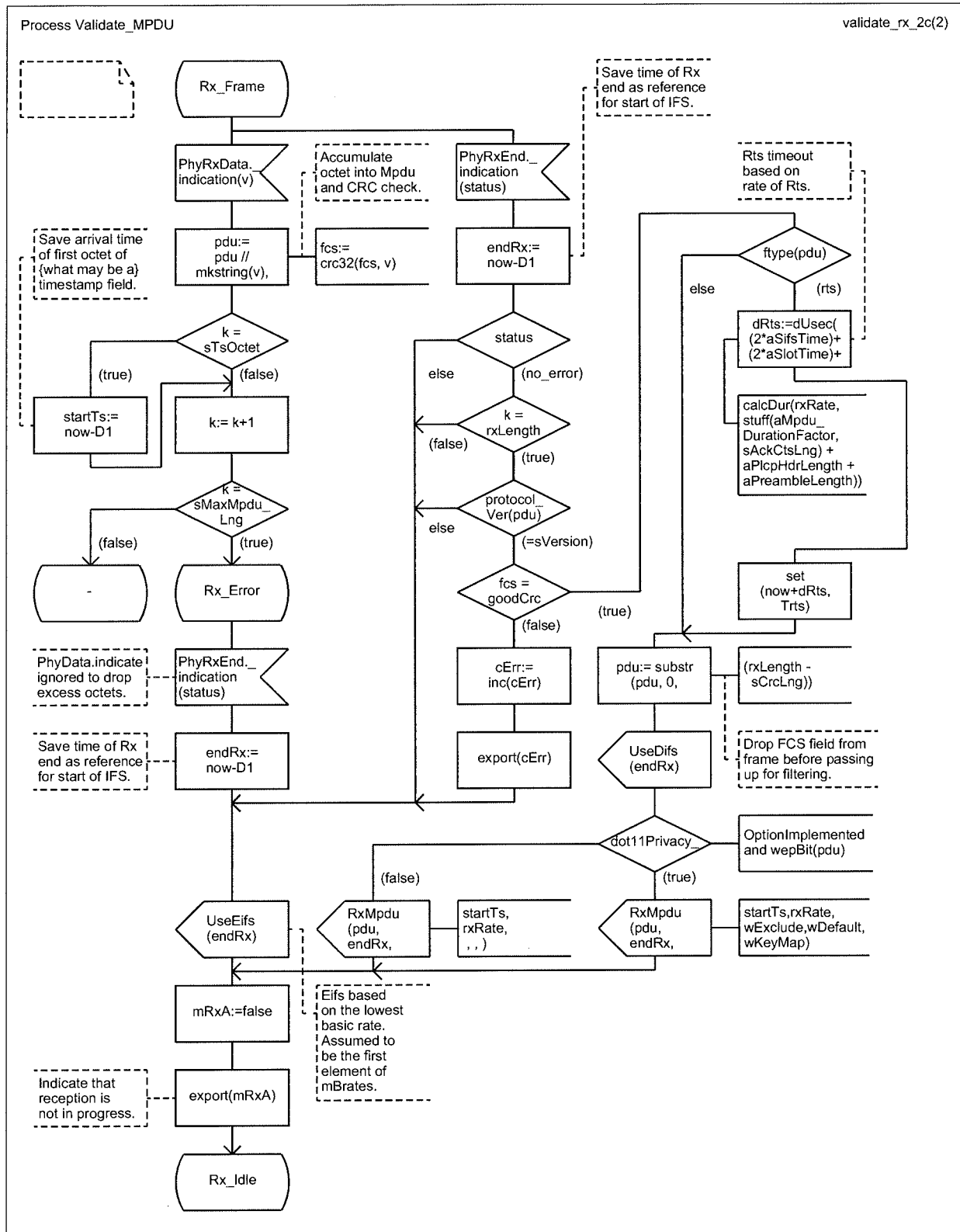


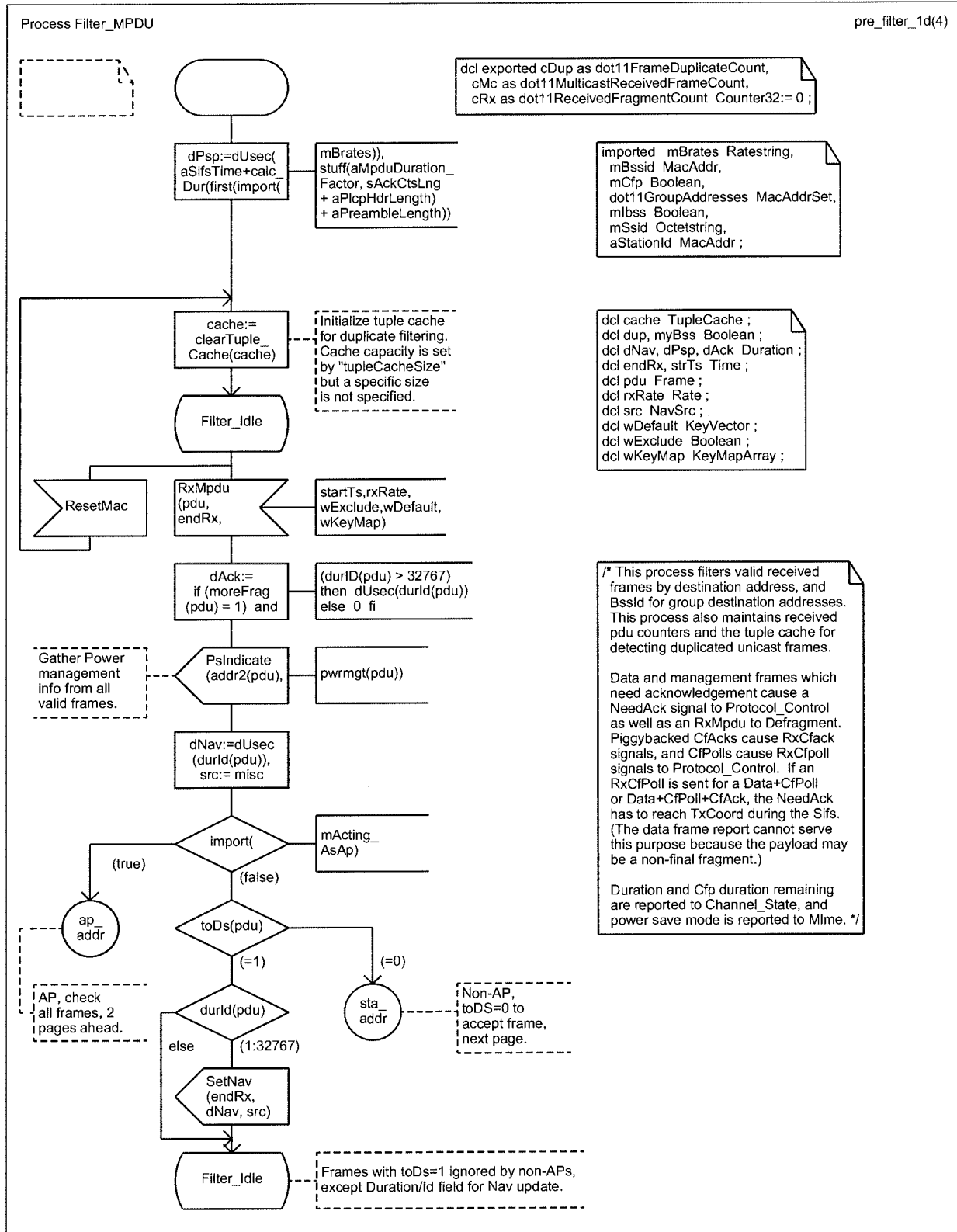


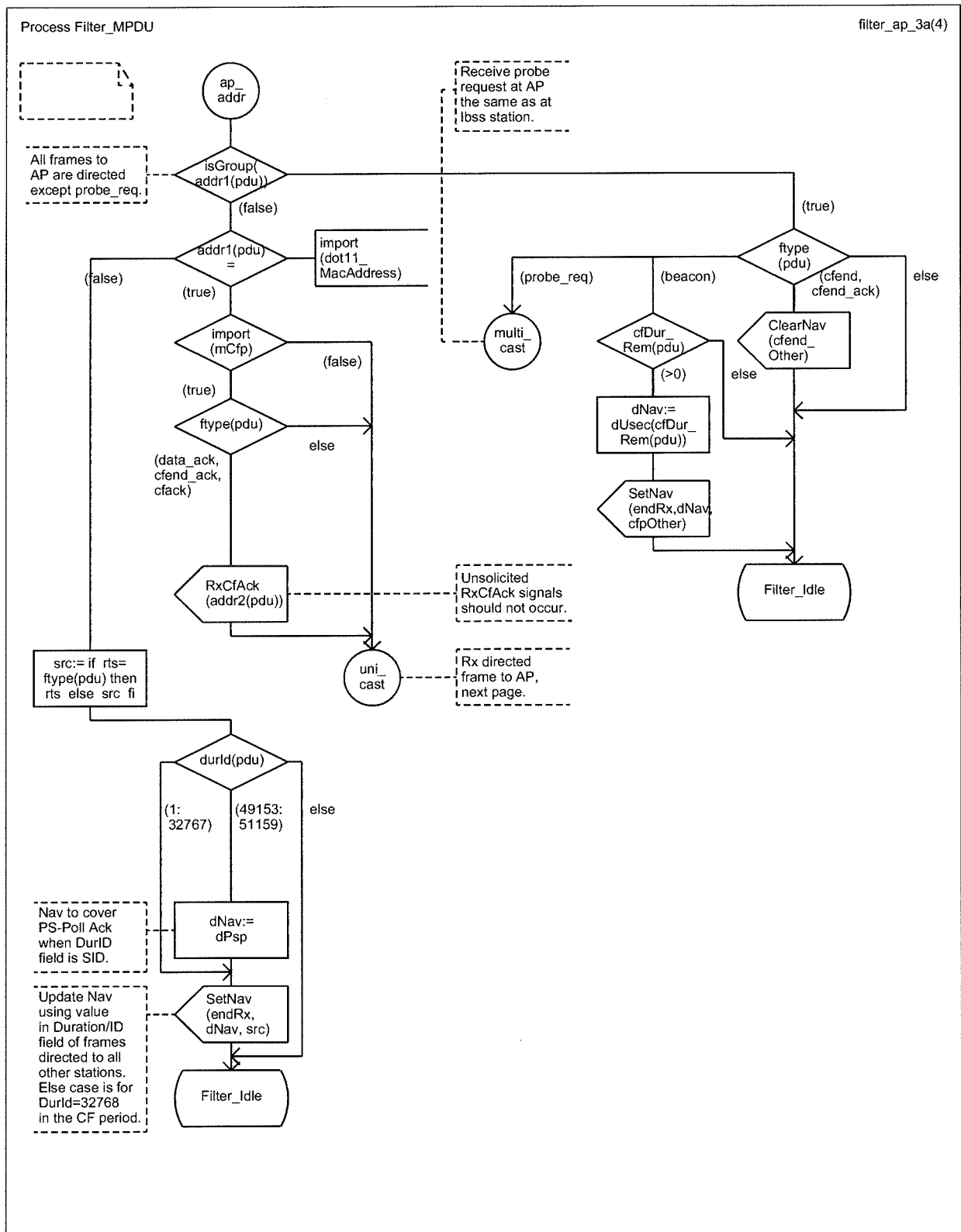


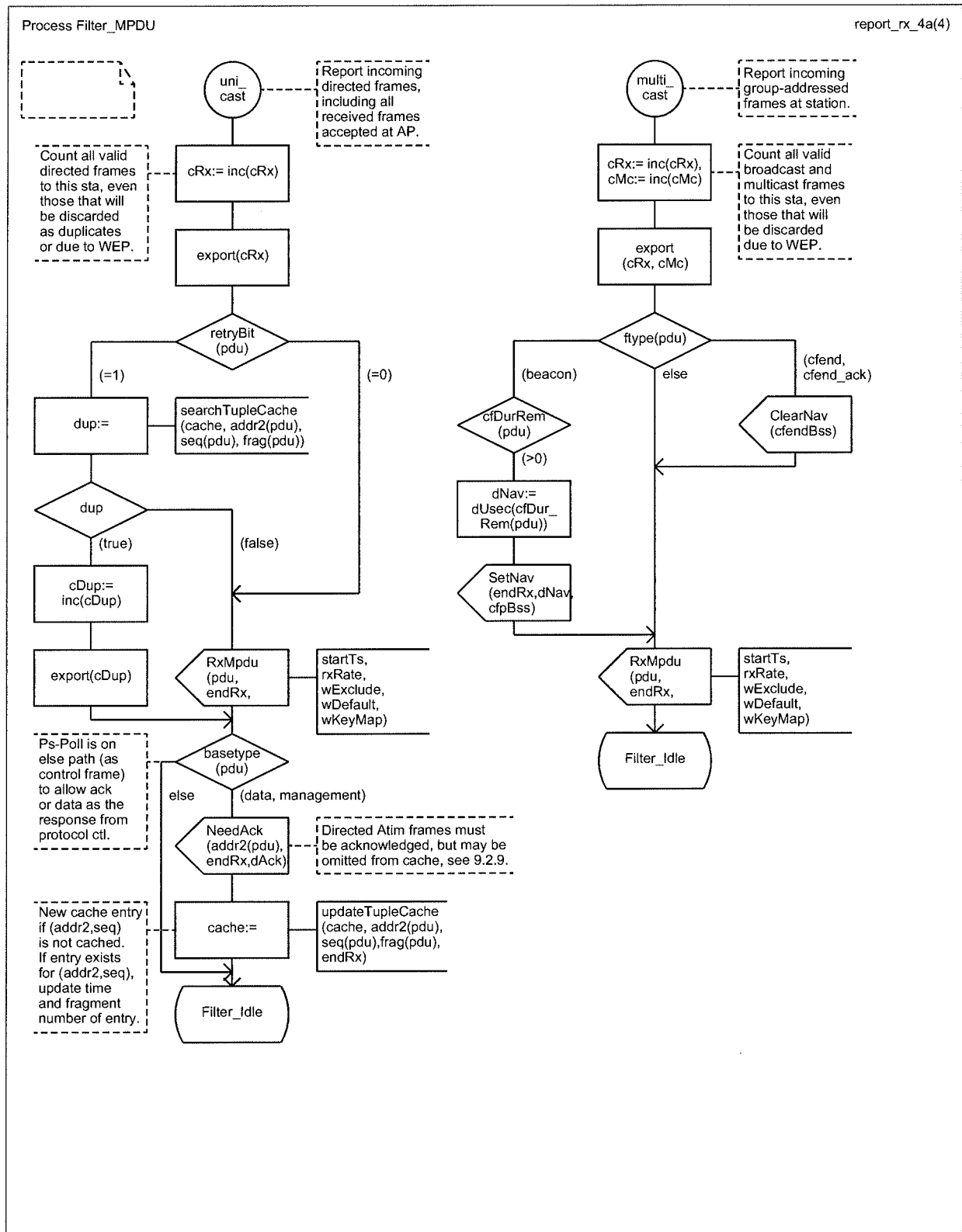


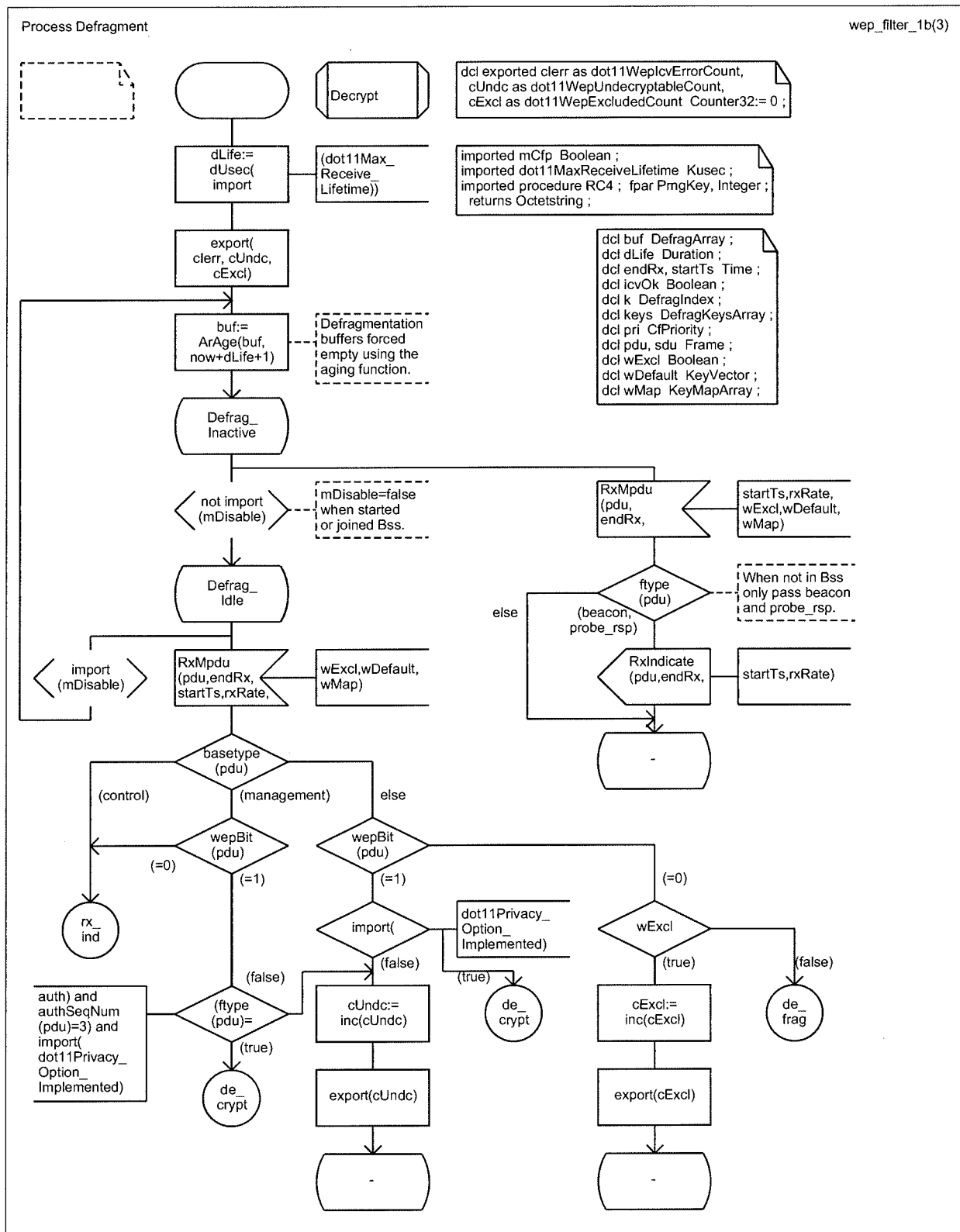






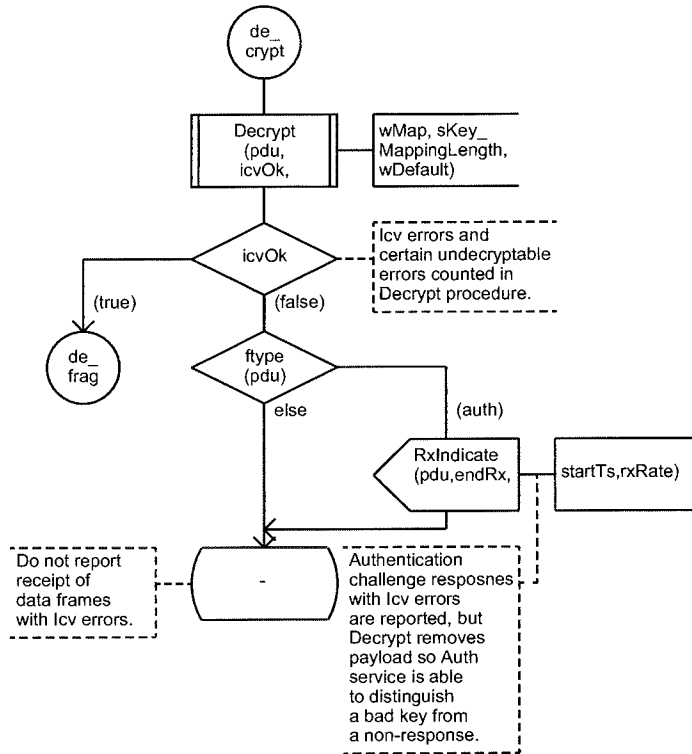


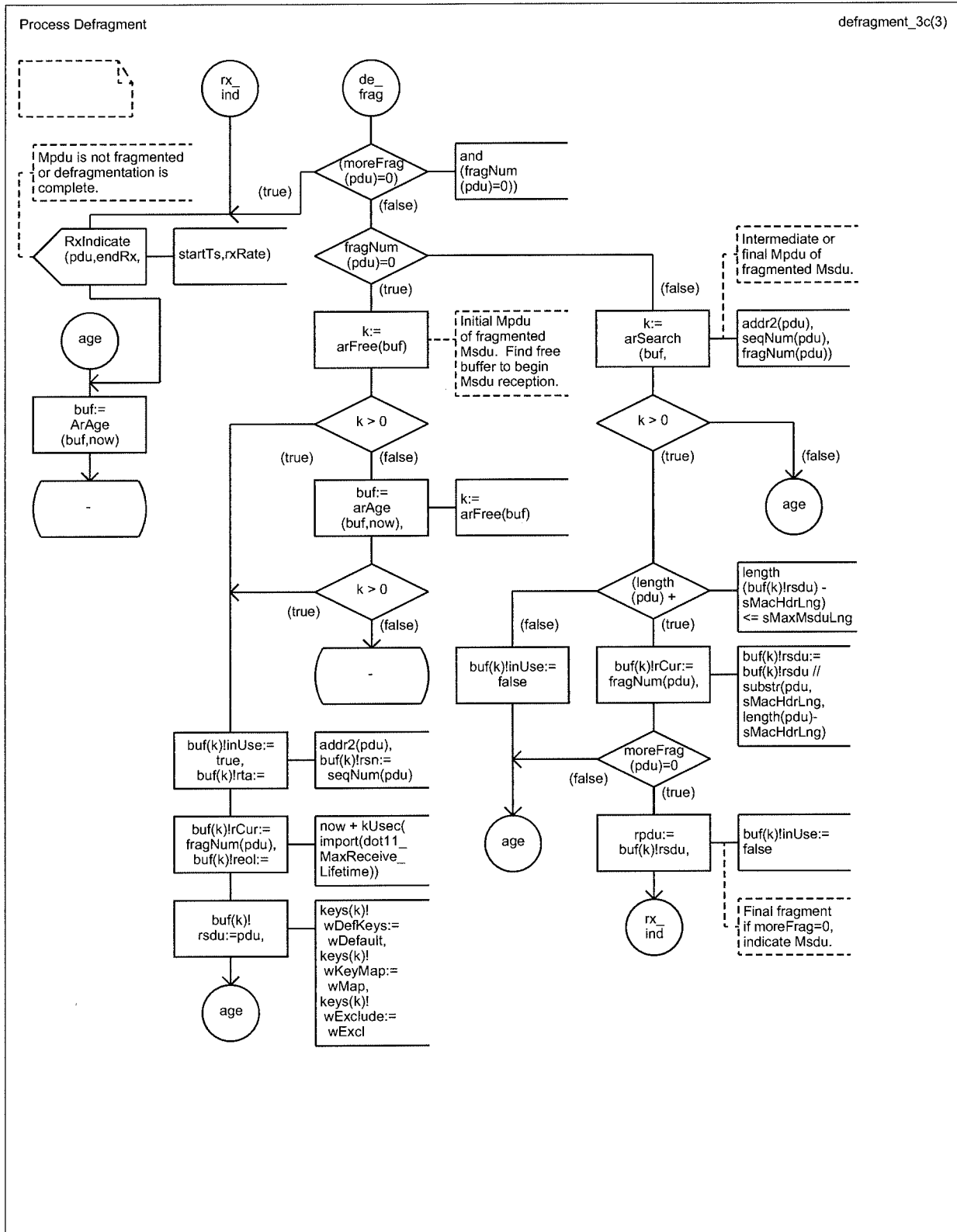


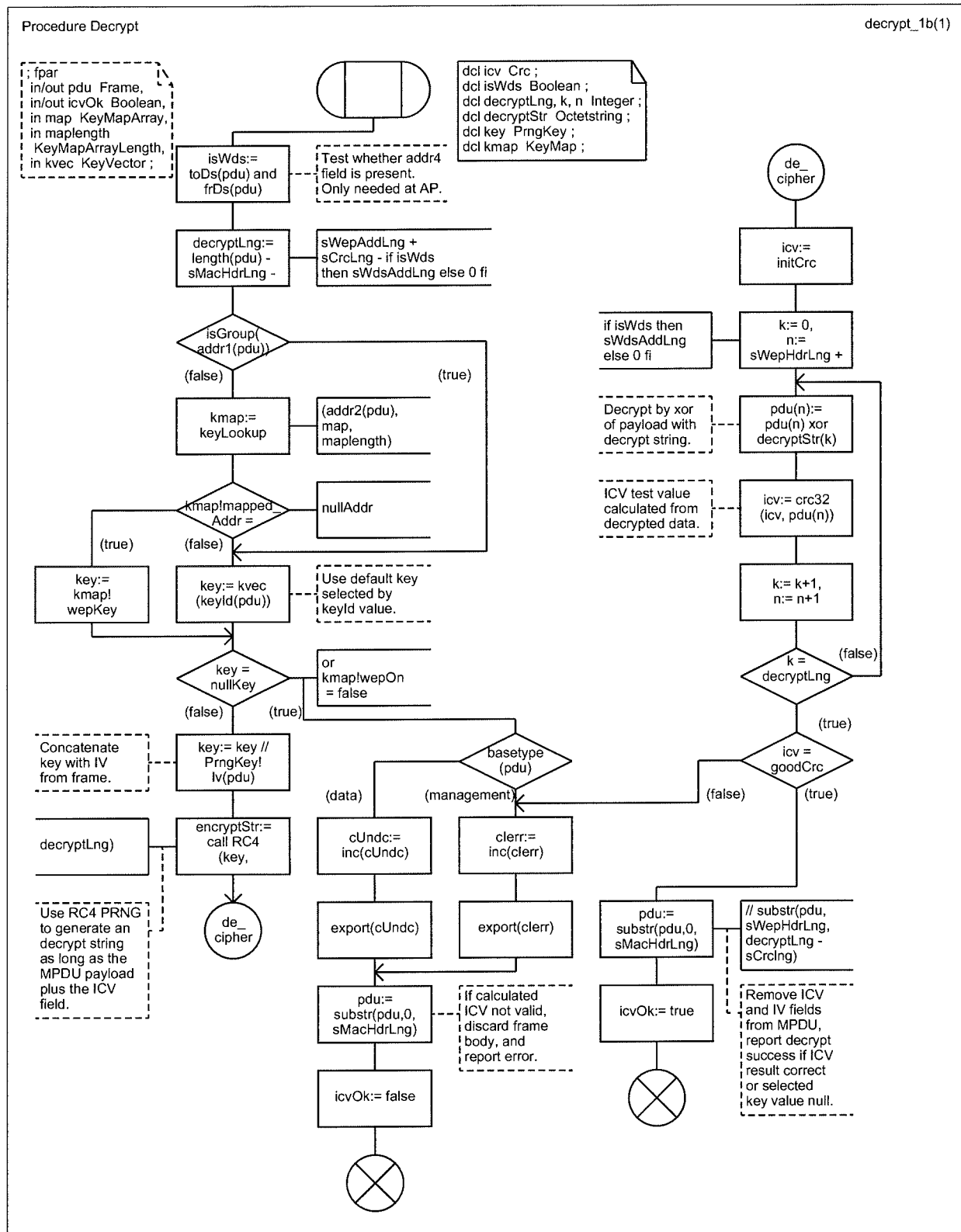


Process Defragment

wep_decrypt_2b(3)







Annex D

(normative)

ASN.1 encoding of the MAC and PHY MIB

```

-- *****
-- * IEEE 802.11 Management Information Base
-- *****
IEEE802dot11-MIB DEFINITIONS ::= BEGIN
    IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE,
        NOTIFICATION-TYPE, Integer32, Counter32 FROM SNMPv2-SMI
        DisplayString, MacAddress, RowStatus,
        TruthValue FROM SNMPv2-TC
        MODULE-COMPLIANCE, OBJECT-GROUP,
        NOTIFICATION-GROUP FROM SNMPv2-CONF
        ifIndex FROM RFC1213-MIB;

-- *****
-- * MODULE IDENTITY
-- *****
ieee802dot11 MODULE-IDENTITY
    LAST-UPDATED "9807080000Z"
    ORGANIZATION "IEEE 802.11"
    CONTACT-INFO
        "WG E-mail: stds-802-11@ieee.org

        Chair: Vic Hayes
        Postal: Lucent Technologies, Inc.
                Zadelstede 1-10
                Nieuwegein, Netherlands
                3431 JZ
        Tel: +31 30 609 7528
        Fax: +31 30 231 6233
        E-mail: vichayes@lucent.com

        Editor: Bob O'Hara
        Postal: Informed Technology, Inc.
                1750 Nantucket Circle, Suite 138
                Santa Clara, CA 95054 USA
        Tel: +1 408 986 9596
        Fax: +1 408 727 2654
        E-mail: bob@informed-technology.com"

    DESCRIPTION
        "The MIB module for IEEE 802.11 entities.
        iso(1).member-body(2).us(840).ieee802dot11(10036) "
        ::= { 1 2 840 10036 }

-- *****
-- * Major sections
-- *****
-- Station Management (SMT) Attributes
--   DEFINED AS "The SMT object class provides the necessary support at the
--   station to manage the processes in the station such that the
--   station may work cooperatively as a part of an IEEE 802.11 network.";

```

```

dot11smt OBJECT IDENTIFIER ::= {ieee802dot11 1}

-- dot11smt GROUPS
-- dot11StationConfigTable ::= {dot11smt 1}
-- dot11AuthenticationAlgorithmsTable ::= {dot11smt 2}
-- dot11WEPDefaultKeysTable ::= {dot11smt 3}
-- dot11WEPKeyMappingsTable ::= {dot11smt 4}
-- dot11PrivacyTable ::= {dot11smt 5}
-- dot11SMTnotification ::= {dot11smt 6}

-- MAC Attributes
-- DEFINED AS "The MAC object class provides the necessary support
-- for the access control, generation, and verification of frame check
-- sequences, and proper delivery of valid data to upper layers.";

dot11mac OBJECT IDENTIFIER ::= {ieee802dot11 2}

-- MAC GROUPS
-- reference IEEE Std 802.1F-1993
-- dot11OperationTable ::= {dot11mac 1}
-- dot11CountersTable ::= {dot11mac 2}
-- dot11GroupAddressesTable ::= {dot11mac 3}

-- Resource Type ID
dot11res OBJECT IDENTIFIER ::= {ieee802dot11 3}
dot11resAttribute OBJECT IDENTIFIER ::= {dot11res 1}

-- PHY Attributes
-- DEFINED AS "The PHY object class provides the necessary support
-- for required PHY operational information that may vary from PHY
-- to PHY and from STA to STA to be communicated to upper layers."

dot11phy OBJECT IDENTIFIER ::= {ieee802dot11 4}

-- phy GROUPS
-- dot11PhyOperationTable ::= {dot11phy 1}
-- dot11PhyAntennaTable ::= {dot11phy 2}
-- dot11PhyTxPowerTable ::= {dot11phy 3}
-- dot11PhyFHSSTable ::= {dot11phy 4}
-- dot11PhyDSSSTable ::= {dot11phy 5}
-- dot11PhyIRTable ::= {dot11phy 6}
-- dot11RegDomainsSupportedTable ::= {dot11phy 7}
-- dot11AntennasListTable ::= {dot11phy 8}
-- dot11SupportedDataRatesTxTable ::= {dot11phy 9}
-- dot11SupportedDataRatesRxTable ::= {dot11phy 10}

-- *****
-- * Textual conventions from 802 definitions
-- *****
WEPKeytype ::= OCTET STRING (SIZE (5))

-- *****
-- * MIB attribute OBJECT-TYPE definitions follow
-- *****

-- *****
-- * SMI Station Config Table
-- *****
dot11StationConfigTable OBJECT-TYPE

```

```

SYNTAX SEQUENCE OF Dot11StationConfigEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Station Configuration attributes. In tabular form to
    allow for multiple instances on an agent."
 ::= { dot11smt 1 }

dot11StationConfigEntry OBJECT-TYPE
SYNTAX Dot11StationConfigEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "An entry in the dot11StationConfigTable. It is
    possible for there to be multiple IEEE 802.11 interfaces
    on one agent, each with its unique MAC address. The
    relationship between an IEEE 802.11 interface and an
    interface in the context of the Internet-standard MIB is
    one-to-one. As such, the value of an ifIndex object
    instance can be directly used to identify corresponding
    instances of the objects defined herein.

    ifIndex - Each 802.11 interface is represented by an
    ifEntry. Interface tables in this MIB module are indexed
    by ifIndex."

INDEX {ifIndex}
 ::= { dot11StationConfigTable 1 }

Dot11StationConfigEntry ::=
SEQUENCE {
    dot11StationID                               MacAddress,
    dot11MediumOccupancyLimit                   INTEGER,
    dot11CFPollable                              TruthValue,
    dot11CFPPeriod                              INTEGER,
    dot11CFPMaxDuration                         INTEGER,
    dot11AuthenticationResponseTimeOut         INTEGER,
    dot11PrivacyOptionImplemented              TruthValue,
    dot11PowerManagementMode                    INTEGER,
    dot11DesiredSSID                            OCTET STRING,
    dot11DesiredBSSType                         INTEGER,
    dot11OperationalRateSet                     OCTET STRING,
    dot11BeaconPeriod                           INTEGER,
    dot11DTIMPeriod                             INTEGER,
    dot11AssociationResponseTimeOut             INTEGER,
    dot11DisassociateReason                     INTEGER,
    dot11DisassociateStation                    MacAddress,
    dot11DeauthenticateReason                   INTEGER,
    dot11DeauthenticateStation                  MacAddress,
    dot11AuthenticateFailStatus                INTEGER,
    dot11AuthenticateFailStation                MacAddress }

dot11StationID OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS read-write
STATUS deprecated
DESCRIPTION
    "The purpose of dot11StationID is to allow a manager to identify

```


a station for its own purposes. This attribute provides for that eventuality while keeping the true MAC address independent. Its syntax is MacAddress. The default value is the station's assigned, unique MAC address."

```
::= { dot11StationConfigEntry 1 }
```

```
dot11MediumOccupancyLimit OBJECT-TYPE
```

```
SYNTAX INTEGER (0..1000)
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

"This attribute shall indicate the maximum amount of time, in TU, that a point coordinator may control the usage of the wireless medium without relinquishing control for long enough to allow at least one instance of DCF access to the medium. The default value of this attribute shall be 100, and the maximum value shall be 1000."

```
::= { dot11StationConfigEntry 2 }
```

```
dot11CFPollable OBJECT-TYPE
```

```
SYNTAX TruthValue
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

"When this attribute is true, it shall indicate that the STA is able to respond to a CF-Poll with a data frame within a SIFS time. This attribute shall be false if the STA is not able to respond to a CF-Poll with a data frame within a SIFS time."

```
::= { dot11StationConfigEntry 3 }
```

```
dot11CFPPeriod OBJECT-TYPE
```

```
SYNTAX INTEGER (0..255)
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

"The attribute shall describe the number of DTIM intervals between the start of CFPs. It is modified by MLME-START.request primitive."

```
::= { dot11StationConfigEntry 4 }
```

```
dot11CFPMaxDuration OBJECT-TYPE
```

```
SYNTAX INTEGER (0..65535)
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

"The attribute shall describe the maximum duration of the CFP in TU that may be generated by the PCF. It is modified by MLME-START.request primitive."

```
::= { dot11StationConfigEntry 5 }
```

```
dot11AuthenticationResponseTimeout OBJECT-TYPE
```

```
SYNTAX INTEGER (1..4294967295)
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

DESCRIPTION

"This attribute shall specify the number of TUs that a responding STA should wait for the next frame in the authentication sequence."

```
::= { dot11StationConfigEntry 6 }
```

dot11PrivacyOptionImplemented OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This attribute, when true, shall indicate that the IEEE 802.11 WEP option is implemented. The default value of this attribute shall be false."

```
::= { dot11StationConfigEntry 7 }
```

dot11PowerManagementMode OBJECT-TYPE

SYNTAX INTEGER { active(1), powersave(2) }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This attribute shall specify the power management mode of the STA. When set to active, it shall indicate that the station is not in power-save mode. When set to powersave, it shall indicate that the station is in power-save mode. The power management mode is transmitted in all frames according to the rules in 7.1.3.1.7."

```
::= { dot11StationConfigEntry 8 }
```

dot11DesiredSSID OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..32))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This attribute reflects the Service Set ID used in the DesiredSSID parameter of the most recent MLME_Scan.request. This value may be modified by an external management entity and used by the local SME to make decisions about the Scanning process."

```
::= { dot11StationConfigEntry 9 }
```

dot11DesiredBSSType OBJECT-TYPE

SYNTAX INTEGER { infrastructure(1), independent(2), any(3) }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This attribute shall specify the type of BSS the station shall use when scanning for a BSS with which to synchronize. This value is used to filter Probe Response frames and Beacons. When set to infrastructure, the station shall only synchronize with a BSS whose Capability Information field has the ESS subfield set to 1. When set to independent, the station shall only synchronize with a BSS whose Capability Information field has the IBSS subfield set to 1. When set to any, the station may synchronize to either type of

```

        BSS."
 ::= { dot11StationConfigEntry 10 }

dot11OperationalRateSet OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1..126))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This attribute shall specify the set of data rates
        at which the station may transmit data. Each octet
        contains a value representing a rate. Each rate
        shall be within the range from 2 to 127,
        corresponding to data rates in increments of
        500 kb/s from 1 Mbit/s to 63.5 Mbit/s, and shall be
        supported (as indicated in the supported rates
        table) for receiving data. This value is reported in
        transmitted Beacon, Probe Request, Probe Response,
        Association Request, Association Response,
        Reassociation Request, and Reassociation Response
        frames, and is used to determine whether a BSS
        with which the station desires to synchronize is
        suitable. It is also used when starting a BSS,
        as specified in 10.3."
 ::= { dot11StationConfigEntry 11 }

dot11BeaconPeriod OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This attribute shall specify the number of TUs that a
        station shall use for scheduling Beacon transmissions.
        This value is transmitted in Beacon and Probe Response
        frames."
 ::= { dot11StationConfigEntry 12 }

dot11DTIMPeriod OBJECT-TYPE
    SYNTAX INTEGER(1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This attribute shall specify the number of beacon
        intervals that shall elapse between transmission of
        Beacons frames containing a TIM element whose DTIM
        Count field is 0. This value is transmitted in
        the DTIM Period field of Beacon frames."
 ::= { dot11StationConfigEntry 13 }

dot11AssociationResponseTimeOut OBJECT-TYPE
    SYNTAX INTEGER(1..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This attribute shall specify the number of TUs that a
        requesting STA should wait for a response to a
        transmitted association-request MMPDU."
 ::= { dot11StationConfigEntry 14 }

dot11DisassociateReason OBJECT-TYPE

```

```

SYNTAX INTEGER(0..65535)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "This attribute holds the most recently transmitted
    Reason Code in a Disassociation frame.  If no
    Disassociation frame has been transmitted,
    the value of this attribute shall be 0."

```

```

REFERENCE "ISO/IEC 8802-11:1999, 7.3.1.7"
 ::= { dot11StationConfigEntry 15 }

```

```

dot11DisassociateStation OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute holds the MAC address from the
        Address 1 field of the most recently transmitted
        Disassociation frame.  If no Disassociation frame has
        been transmitted, the value of this attribute
        shall be 0."
 ::= { dot11StationConfigEntry 16 }

```

```

dot11DeauthenticateReason OBJECT-TYPE
    SYNTAX INTEGER(0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute holds the most recently
        transmitted Reason Code in a Deauthentication frame.
        If no Deauthentication frame has been transmitted, the
        value of this attribute shall be 0."

```

```

REFERENCE "ISO/IEC 8802-11:1999, 7.3.1.7"
 ::= { dot11StationConfigEntry 17 }

```

```

dot11DeauthenticateStation OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute holds the MAC address from the Address 1
        field of the most recently transmitted Deauthentication frame.
        If no Deauthentication frame has been transmitted,
        the value of this attribute shall be 0."
 ::= { dot11StationConfigEntry 18 }

```

```

dot11AuthenticateFailStatus OBJECT-TYPE
    SYNTAX INTEGER(0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute holds the most recently
        transmitted Status Code in a failed Authentication frame.
        If no failed Authentication frame has been transmitted, the
        value of this attribute shall be 0."

```

```

REFERENCE "ISO/IEC 8802-11:1999, 7.3.1.9"

```

```

 ::= { dot11StationConfigEntry 19 }

dot11AuthenticateFailStation OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute holds the MAC address from the
        Address 1 field of the most recently transmitted
        failed Authentication frame.  If no failed Authentication
        frame has been transmitted, the value of this attribute
        shall be 0."
 ::= { dot11StationConfigEntry 20 }

-- *****
-- *      End of dot11StationConfig TABLE
-- *****

-- *****
-- *      AuthenticationAlgorithms TABLE
-- *****
dot11AuthenticationAlgorithmsTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11AuthenticationAlgorithmsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This (conceptual) table of attributes shall be a set of
        all the authentication algorithms supported by the
        stations.  The following are the default values and the
        associated algorithm:
            Value = 1: Open System
            Value = 2: Shared Key"
    REFERENCE "ISO/IEC 8802-11:1999, 7.3.1.1"
 ::= { dot11smt 2 }

dot11AuthenticationAlgorithmsEntry OBJECT-TYPE
    SYNTAX Dot11AuthenticationAlgorithmsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An Entry (conceptual row) in the Authentication
        Algorithms Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry.  Interface tables in this MIB module are indexed
        by ifIndex."

    INDEX { ifIndex,
            dot11AuthenticationAlgorithmsIndex }
 ::= { dot11AuthenticationAlgorithmsTable 1 }

Dot11AuthenticationAlgorithmsEntry ::= SEQUENCE {
    dot11AuthenticationAlgorithmsIndex          Integer32,
    dot11AuthenticationAlgorithm              INTEGER,
dot11AuthenticationAlgorithmsEnable          TruthValue }

dot11AuthenticationAlgorithmsIndex OBJECT-TYPE
    SYNTAX Integer32

```

```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "The auxiliary variable used to identify instances
    of the columnar objects in the Authentication Algorithms Table."
 ::= { dot11AuthenticationAlgorithmsEntry 1 }

dot11AuthenticationAlgorithm OBJECT-TYPE
    SYNTAX INTEGER { openSystem (1), sharedKey (2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute shall be a set of all the authentication
        algorithms supported by the STAs. The following are the
        default values and the associated algorithm.
        Value = 1: Open System
        Value = 2: Shared Key"

 ::= { dot11AuthenticationAlgorithmsEntry 2 }

dot11AuthenticationAlgorithmsEnable OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This attribute, when true at a station,
        shall enable the acceptance of the authentication
        algorithm described in the corresponding table
        entry in authentication frames received by the
        station that have odd authentication sequence numbers.
        The default value of this attribute shall be 1 for
        the Open System table entry and 2 for all other table entries."

 ::= { dot11AuthenticationAlgorithmsEntry 3 }

-- *****
-- *      End of AuthenticationAlgorithms TABLE
-- *****

-- *****
-- *      WEPDefaultKeys TABLE
-- *****

dot11WEPDefaultKeysTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11WEPDefaultKeysEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Conceptual table for WEP default keys. This table shall
        contain the four WEP default secret key values
        corresponding to the four possible KeyID values. The WEP
        default secret keys are logically WRITE-ONLY. Attempts to
        read the entries in this table shall return unsuccessful
        status and values of null or zero. The default value of
        each WEP default key shall be null."

    REFERENCE "ISO/IEC 8802-11:1999, 8.3.2"
 ::= { dot11smt 3 }

dot11WEPDefaultKeysEntry OBJECT-TYPE

```

```

SYNTAX Dot11WEPDefaultKeysEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "An Entry (conceptual row) in the WEP Default Keys Table.

    ifIndex - Each 802.11 interface is represented by an
    ifEntry. Interface tables in this MIB module are indexed
    by ifIndex."

INDEX {ifIndex, dot11WEPDefaultKeyIndex}
 ::= { dot11WEPDefaultKeysTable 1 }

Dot11WEPDefaultKeysEntry ::= SEQUENCE {
    dot11WEPDefaultKeyIndex    INTEGER,
    dot11WEPDefaultKeyValue    WEPKeytype}

dot11WEPDefaultKeyIndex OBJECT-TYPE
    SYNTAX INTEGER (1..4)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances
        of the columnar objects in the WEP Default Keys Table.
        The value of this variable is equal to the WEPDefaultKeyID + 1"
    ::= { dot11WEPDefaultKeysEntry 1 }

dot11WEPDefaultKeyValue OBJECT-TYPE
    SYNTAX WEPKeytype
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "A WEP default secret key value."
    ::= { dot11WEPDefaultKeysEntry 2 }

-- *****
-- *      End of WEPDefaultKeys TABLE
-- *****

-- *****
-- *      WEPKeyMappings TABLE
-- *****

dot11WEPKeyMappingsTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11WEPKeyMappingsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Conceptual table for WEP Key Mappings. The MIB supports
        the ability to share a separate WEP key for each RA/TA
        pair. The Key Mappings Table contains zero or one entry
        for each MAC address and contains two fields for each
        entry: WEPOn and the corresponding WEP key. The WEP key
        mappings are logically WRITE-ONLY. Attempts to read the
        entries in this table shall return unsuccessful status and
        values of null or zero. The default value for all WEPOn
        fields is false."
    REFERENCE "ISO/IEC 8802-11:1999, 8.3.2"
    ::= { dot11smt 4 }

```

```

dot11WEPKeyMappingsEntry OBJECT-TYPE
    SYNTAX Dot11WEPKeyMappingsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An Entry (conceptual row) in the WEP Key Mappings Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."

        INDEX {ifIndex, dot11WEPKeyMappingIndex}
    ::= { dot11WEPKeyMappingsTable 1 }

Dot11WEPKeyMappingsEntry ::= SEQUENCE {
    dot11WEPKeyMappingIndex          Integer32,
    dot11WEPKeyMappingAddress       MacAddress,
    dot11WEPKeyMappingWEPOn        TruthValue,
    dot11WEPKeyMappingValue        WEPKeytype,
    dot11WEPKeyMappingStatus       RowStatus }

dot11WEPKeyMappingIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances
        of the columnar objects in the WEP Key Mappings Table."
    ::= { dot11WEPKeyMappingsEntry 1 }

dot11WEPKeyMappingAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The MAC address of the STA for which the values from this
        key mapping entry are to be used."
    ::= { dot11WEPKeyMappingsEntry 2 }

dot11WEPKeyMappingWEPOn OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Boolean as to whether WEP is to be used when communicating
        with the dot11WEPKeyMappingAddress STA."
    ::= { dot11WEPKeyMappingsEntry 3 }

dot11WEPKeyMappingValue OBJECT-TYPE
    SYNTAX WEPKeytype
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "A WEP secret key value."
    ::= { dot11WEPKeyMappingsEntry 4 }

dot11WEPKeyMappingStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create

```



```

STATUS current
DESCRIPTION
    "The status column used for creating, modifying, and
    deleting instances of the columnar objects in the WEP key
    mapping Table."

    DEFVAL {active}
 ::= { dot11WEPKeyMappingsEntry 5 }

-- *****
-- *      End of WEPKeyMappings TABLE
-- *****

-- *****
-- *      dot11PrivacyTable TABLE
-- *****
dot11PrivacyTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PrivacyEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group containing attributes concerned with IEEE 802.11
        Privacy.  Created as a table to allow multiple
        instantiations on an agent."

 ::= { dot11smt 5 }

dot11PrivacyEntry OBJECT-TYPE
    SYNTAX Dot11PrivacyEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PrivacyTable Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry.  Interface tables in this MIB module are indexed
        by ifIndex."
    INDEX {ifIndex}
 ::= { dot11PrivacyTable 1 }

Dot11PrivacyEntry ::= SEQUENCE {
    dot11PrivacyInvoked                TruthValue,
    dot11WEPDefaultKeyID                INTEGER,
    dot11WEPKeyMappingLength            INTEGER,
    dot11ExcludeUnencrypted             TruthValue,
    dot11WEPICVErrorCount               Counter32,
    dot11WEPExcludedCount                Counter32}

dot11PrivacyInvoked OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "When this attribute is true, it shall indicate that the IEEE
        802.11 WEP mechanism is used for transmitting frames of type
        Data. The default value of this attribute shall be false."
 ::= { dot11PrivacyEntry 1 }

dot11WEPDefaultKeyID OBJECT-TYPE

```

```

SYNTAX INTEGER (0..3)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "This attribute shall indicate the use of the first,
    second, third, or fourth element of the WEPDefaultKeys
    array when set to values of zero, one, two, or three. The
    default value of this attribute shall be 0."
REFERENCE "ISO/IEC 8802-11:1999, 8.3.2"
 ::= { dot11PrivacyEntry 2 }

dot11WEPKeyMappingLength OBJECT-TYPE
SYNTAX INTEGER (10..4294967295)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "The maximum number of tuples that dot11WEPKeyMappings can hold."
REFERENCE "ISO/IEC 8802-11:1999, 8.3.2"
 ::= { dot11PrivacyEntry 3 }

dot11ExcludeUnencrypted OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION

    "When this attribute is true, the STA shall not indicate at
    the MAC service interface received MSDUs that have the WEP
    subfield of the Frame Control field equal to zero. When this
    attribute is false, the STA may accept MSDUs that have the WEP
    subfield of the Frame Control field equal to zero. The default
    value of this attribute shall be false."

 ::= { dot11PrivacyEntry 4 }

dot11WEPICVErrorCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "This counter shall increment when a frame is received with the
    WEP subfield of the Frame Control field set to one and the value
    of the ICV as received in the frame does not match the ICV value
    that is calculated for the contents of the received frame."
 ::= { dot11PrivacyEntry 5 }

dot11WEPExcludedCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "This counter shall increment when a frame is received with the
    WEP subfield of the Frame Control field set to zero and the value
    of dot11ExcludeUnencrypted causes that frame to be discarded."
 ::= { dot11PrivacyEntry 6 }

-- *****
-- *      End of dot11Privacy TABLE
-- *****

```

```

-- *****
-- *      SMT notification Objects
-- *****
dot11SMTnotification OBJECT IDENTIFIER ::= { dot11smt 6 }

dot11Disassociate NOTIFICATION-TYPE
    OBJECTS { ifIndex, dot11DisassociateReason,
dot11DisassociateStation }
    STATUS current
    DESCRIPTION
        "The disassociate notification shall be sent when the STA
        sends a Disassociation frame. The value of the notification
        shall include the MAC address of the MAC to which the
        Disassociation frame was sent and the reason for
        the disassociation.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."

    ::= { dot11SMTnotification 0 1 }

dot11Deauthenticate NOTIFICATION-TYPE
    OBJECTS { ifIndex, dot11DeauthenticateReason,
dot11DeauthenticateStation }
    STATUS current
    DESCRIPTION
        "The deauthenticate notification shall be sent when the STA
        sends a Deauthentication frame. The value of the notification
        shall include the MAC address of the MAC to which the
        Deauthentication frame was sent and the reason for the
        deauthentication.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."

    ::= { dot11SMTnotification 0 2 }

dot11AuthenticateFail NOTIFICATION-TYPE
    OBJECTS { ifIndex, dot11AuthenticateFailStatus,
dot11AuthenticateFailStation }
    STATUS current
    DESCRIPTION
        "The authenticate failure notification shall be sent
        when the STA sends an Authentication frame with a
        status code other than 'successful.' The value of
        the notification shall include the MAC address of the
        MAC to which the Authentication frame was sent and the
        reason for the authentication failure.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."

    ::= { dot11SMTnotification 0 3 }

```

```

-- *****
-- *   End of SMT notification Objects
-- *****

-- *****
-- *   MAC Attribute Templates
-- *****

-- *****
-- *   dot11OperationTable TABLE
-- *****
dot11OperationTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11OperationEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group contains MAC attributes pertaining to the operation
        of the MAC. This has been implemented as a table in order
        to allow for multiple instantiations on an agent."

    ::= { dot11mac 1 }

dot11OperationEntry OBJECT-TYPE
    SYNTAX Dot11OperationEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11OperationEntry Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."
    INDEX {ifIndex}
    ::= { dot11OperationTable 1 }

Dot11OperationEntry ::= SEQUENCE {
    dot11MACAddress          MacAddress,
    dot11RTSThreshold       INTEGER,
    dot11ShortRetryLimit    INTEGER,
    dot11LongRetryLimit     INTEGER,
    dot11FragmentationThreshold  INTEGER,
    dot11MaxTransmitMSDULifetime  INTEGER,
    dot11MaxReceiveLifetime     INTEGER,
    dot11ManufacturerID      DisplayString,
    dot11ProductID           DisplayString}

dot11MACAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Unique MAC Address assigned to the STA."
    ::= { dot11OperationEntry 1 }

dot11RTSThreshold OBJECT-TYPE
    SYNTAX INTEGER (0..2347)
    MAX-ACCESS read-write
    STATUS current

```

DESCRIPTION

"This attribute shall indicate the number of octets in an MPDU, below which an RTS/CTS handshake shall not be performed. An RTS/CTS handshake shall be performed at the beginning of any frame exchange sequence where the MPDU is of type Data or Management, the MPDU has an individual address in the Address1 field, and the length of the MPDU is greater than this threshold. (For additional details, refer to Table 21 in 9.7.) Setting this attribute to be larger than the maximum MSDU size shall have the effect of turning off the RTS/CTS handshake for frames of Data or Management type transmitted by this STA. Setting this attribute to zero shall have the effect of turning on the RTS/CTS handshake for all frames of Data or Management type transmitted by this STA. The default value of this attribute shall be 2347."

```
::= { dot11OperationEntry 2 }
```

dot11ShortRetryLimit OBJECT-TYPE

```
SYNTAX INTEGER (1..255)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
```

"This attribute shall indicate the maximum number of transmission attempts of a frame, the length of which is less than or equal to dot11RTSThreshold, that shall be made before a failure condition is indicated. The default value of this attribute shall be 7."

```
::= { dot11OperationEntry 3 }
```

dot11LongRetryLimit OBJECT-TYPE

```
SYNTAX INTEGER (1..255)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
```

"This attribute shall indicate the maximum number of transmission attempts of a frame, the length of which is greater than dot11RTSThreshold, that shall be made before a failure condition is indicated. The default value of this attribute shall be 4."

```
::= { dot11OperationEntry 4 }
```

dot11FragmentationThreshold OBJECT-TYPE

```
SYNTAX INTEGER (256..2346)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
```

"This attribute shall specify the current maximum size, in octets, of the MPDU that may be delivered to the PHY. An MSDU shall be broken into fragments if its size exceeds the value of this attribute after adding MAC headers and trailers. An MSDU or MMPDU shall be fragmented when the resulting frame has an individual address in the Address1 field, and the length of the frame is larger than this threshold. The default value

for this attribute shall be the lesser of 2346 or the
 aMPDUMaxLength of the attached PHY and shall never exceed the
 lesser of 2346 or the aMPDUMaxLength of the attached PHY. The
 value of this attribute shall never be less than 256."

::= { dot11OperationEntry 5 }

dot11MaxTransmitMSDULifetime OBJECT-TYPE
 SYNTAX INTEGER (1..4294967295)
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION

"The MaxTransmitMSDULifetime shall be the elapsed time in TU,
 after the initial transmission of an MSDU, after which further
 attempts to transmit the MSDU shall be terminated. The default
 value of this attribute shall be 512."

::= { dot11OperationEntry 6 }

dot11MaxReceiveLifetime OBJECT-TYPE
 SYNTAX INTEGER (1..4294967295)
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION

"The MaxReceiveLifetime shall be the elapsed time in TU,
 after the initial reception of a fragmented MMPDU or MSDU,
 after which further attempts to reassemble the MMPDU or
 MSDU shall be terminated. The default value shall be 512."

::= { dot11OperationEntry 7 }

dot11ManufacturerID OBJECT-TYPE
 SYNTAX DisplayString (SIZE(0..128))
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The ManufacturerID shall include, at a minimum, the name
 of the manufacturer. It may include additional
 information at the manufacturer's discretion. The default
 value of this attribute shall be null."

::= { dot11OperationEntry 8 }

dot11ProductID OBJECT-TYPE
 SYNTAX DisplayString (SIZE(0..128))
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The ProductID shall include, at a minimum, an identifier
 that is unique to the manufacturer. It may include
 additional information at the manufacturer's discretion.
 The default value of this attribute shall be null."

::= { dot11OperationEntry 9 }

-- *****
 -- * End of dot11OperationEntry TABLE

```

-- *****
-- *****
-- *      dot11Counters TABLE
-- *****
dot11CountersTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11CountersEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "Group containing attributes that are MAC counters.
        Implemented as a table to allow for multiple
        instantiations on an agent."

    ::= { dot11mac 2 }

dot11CountersEntry OBJECT-TYPE
    SYNTAX Dot11CountersEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "An entry in the dot11CountersEntry Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."
    INDEX {ifIndex}
    ::= { dot11CountersTable 1 }

Dot11CountersEntry ::= SEQUENCE {
    dot11TransmittedFragmentCount Counter32,
    dot11MulticastTransmittedFrameCount Counter32,
    dot11FailedCount Counter32,
    dot11RetryCount Counter32,
    dot11MultipleRetryCount Counter32,
    dot11FrameDuplicateCount Counter32,
    dot11RTSSuccessCount Counter32,
    dot11RTSFailureCount Counter32,
    dot11ACKFailureCount Counter32,
    dot11ReceivedFragmentCount Counter32,
    dot11MulticastReceivedFrameCount Counter32,
    dot11FCSErrorCount Counter32,
    dot11TransmittedFrameCount Counter32,
    dot11WEPUndecryptableCount Counter32 }

dot11TransmittedFragmentCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This counter shall be incremented for an acknowledged
        MPDU with an individual address in the address 1 field
        or an MPDU with a multicast address in the address 1 field
        of type Data or Management."

    ::= { dot11CountersEntry 1 }

```

```

dot11MulticastTransmittedFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This counter shall increment only when the multicast
        bit is set in the destination MAC address of a successfully
        transmitted MSDU.  When operating as a STA in an ESS, where
        these frames are directed to the AP, this implies having
        received an acknowledgment to all associated MPDUs. "

 ::= { dot11CountersEntry 2 }

dot11FailedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This counter shall increment when an MSDU is not
        transmitted successfully due to the number of
        transmit attempts exceeding either the
        dot11ShortRetryLimit or dot11LongRetryLimit. "

 ::= { dot11CountersEntry 3 }

dot11RetryCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This counter shall increment when an MSDU is successfully
        transmitted after one or more retransmissions."

 ::= { dot11CountersEntry 4 }

dot11MultipleRetryCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This counter shall increment when an MSDU is successfully
        transmitted after more than one retransmission."

 ::= { dot11CountersEntry 5 }

dot11FrameDuplicateCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This counter shall increment when a frame is received
        that the Sequence Control field indicates is a
        duplicate."

 ::= { dot11CountersEntry 6 }

```



```

dot11RTSSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

    "This counter shall increment when a CTS is received in
    response to an RTS."

    ::= { dot11CountersEntry 7 }

dot11RTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

    "This counter shall increment when a CTS is not received in
    response to an RTS."

    ::= { dot11CountersEntry 8 }

dot11ACKFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

    "This counter shall increment when an ACK is not received
    when expected."

    ::= { dot11CountersEntry 9 }

dot11ReceivedFragmentCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

    "This counter shall be incremented for each successfully
    received MPDU of type Data or Management."

    ::= { dot11CountersEntry 10 }

dot11MulticastReceivedFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

    "This counter shall increment when a MSDU is received
    with the multicast bit set in the destination
    MAC address."

    ::= { dot11CountersEntry 11 }

dot11FCSErrorCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only

```

```

STATUS current
DESCRIPTION

"This counter shall increment when an FCS error is
detected in a received MPDU."

 ::= { dot11CountersEntry 12 }

dot11TransmittedFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This counter shall increment for each successfully transmitted MSDU."

 ::= { dot11CountersEntry 13 }

dot11WEPUndecryptableCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This counter shall increment when a frame is received with
        the WEP subfield of the Frame Control field set to one and the
        WEPon value for the key mapped to the TA's MAC address
        indicates that the frame should not have been encrypted or
        that frame is discarded due to the receiving STA not
        implementing the privacy option."

 ::= { dot11CountersEntry 14 }

-- *****
-- *      End of dot11CountersEntry TABLE
-- *****

-- *****
-- *      GroupAddresses TABLE
-- *****

dot11GroupAddressesTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11GroupAddressesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "A conceptual table containing a set of MAC addresses
        identifying the multicast addresses for which this STA
        will receive frames. The default value of this attribute
        shall be null."

 ::= { dot11mac 3 }

dot11GroupAddressesEntry OBJECT-TYPE
    SYNTAX Dot11GroupAddressesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "An Entry (conceptual row) in the Group Addresses Table.

```

```

    ifIndex - Each 802.11 interface is represented by an
    ifEntry. Interface tables in this MIB module are indexed
    by ifIndex."
INDEX {ifIndex, dot11GroupAddressesIndex}

 ::= { dot11GroupAddressesTable 1 }

Dot11GroupAddressesEntry ::= SEQUENCE {
    dot11GroupAddressesIndex Integer32,
    dot11Address              MacAddress,
    dot11GroupAddressesStatus RowStatus}

dot11GroupAddressesIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

    "The auxiliary variable used to identify instances
    of the columnar objects in the Group Addresses Table."

 ::= { dot11GroupAddressesEntry 1 }

dot11Address OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION

    "MAC address identifying a multicast addresses
    from which this STA will receive frames."

 ::= { dot11GroupAddressesEntry 2 }

dot11GroupAddressesStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION

    "The status column used for creating, modifying, and
    deleting instances of the columnar objects in the Group
    Addresses Table."

    DEFVAL {active}
 ::= { dot11GroupAddressesEntry 3 }
-- *****
-- *      End of GroupAddress TABLE
-- *****

-- *****
-- *      Resource Type Attribute Templates
-- *****

dot11ResourceTypeIDName OBJECT-TYPE
    SYNTAX DisplayString (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

```

"Contains the name of the Resource Type ID managed object. The attribute is read-only and always contains the value RTID. This attribute value shall not be used as a naming attribute for any other managed object class."

REFERENCE "IEEE Std 802.1F-1993, A.7"

DEFVAL {"RTID"}

::= { dot11resAttribute 1 }

-- *****

-- * dot11ResourceInfo TABLE

-- *****

dot11ResourceInfoTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot11ResourceInfoEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Provides a means of indicating, in data readable from a managed object, information that identifies the source of the implementation."

REFERENCE "IEEE Std 802.1F-1993, A.7"

::= { dot11resAttribute 2 }

dot11ResourceInfoEntry OBJECT-TYPE

SYNTAX Dot11ResourceInfoEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the dot11ResourceInfo Table.

ifIndex - Each 802.11 interface is represented by an ifEntry. Interface tables in this MIB module are indexed by ifIndex."

INDEX {ifIndex}

::= { dot11ResourceInfoTable 1 }

Dot11ResourceInfoEntry ::= SEQUENCE {

dot11manufacturerOUI OCTET STRING,

dot11manufacturerName DisplayString,

dot11manufacturerProductName DisplayString,

dot11manufacturerProductVersion DisplayString}

dot11manufacturerOUI OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(3))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Takes the value of an organizationally unique identifier."

::= { dot11ResourceInfoEntry 1 }

dot11manufacturerName OBJECT-TYPE

SYNTAX DisplayString (SIZE(0..128))

MAX-ACCESS read-only

```

STATUS current
DESCRIPTION

"A printable string used to identify the manufacturer of the
resource. Maximum string length is 128 octets."
 ::= { dot11ResourceInfoEntry 2 }

dot11manufacturerProductName OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..128))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

    "A printable string used to identify the manufacturer's product
    name of the resource. Maximum string length is 128 octets."
 ::= { dot11ResourceInfoEntry 3 }

dot11manufacturerProductVersion OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..128))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

    "Printable string used to identify the manufacturer's product
    version of the resource. Maximum string length is 128 octets."
 ::= { dot11ResourceInfoEntry 4 }

-- *****
-- *      End of dot11ResourceInfo TABLE
-- *****

-- *****
-- *      PHY Attribute Templates
-- *****

-- *****
-- *      dot11PhyOperation TABLE
-- *****
dot11PhyOperationTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyOperationEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

    "PHY level attributes concerned with
    operation. Implemented as a table indexed on
    ifIndex to allow for multiple instantiations on an
    Agent."

 ::= { dot11phy 1 }

dot11PhyOperationEntry OBJECT-TYPE
    SYNTAX Dot11PhyOperationEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

    "An entry in the dot11PhyOperation Table.

```

```

    ifIndex - Each 802.11 interface is represented by an
    ifEntry. Interface tables in this MIB module are indexed
    by ifIndex."
    INDEX {ifIndex}
 ::= { dot11PhyOperationTable 1 }

Dot11PhyOperationEntry ::= SEQUENCE {
    dot11PHYType          INTEGER,
    dot11CurrentRegDomain Integer32,
    dot11TempType         INTEGER
}

dot11PHYType OBJECT-TYPE
    SYNTAX INTEGER {fhss(1), dsss(2), irbaseband(3)}
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This is an 8-bit integer value that identifies the PHY type
        supported by the attached PLCP and PMD. Currently defined
        values and their corresponding PHY types are:

            FHSS 2.4 GHz = 01 , DSSS 2.4 GHz = 02, IR Baseband = 03"

 ::= { dot11PhyOperationEntry 1 }

dot11CurrentRegDomain OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION

        "The current regulatory domain this instance of the PMD is
        supporting. This object corresponds to one of the
        RegDomains listed in dot11RegDomainsSupported."

 ::= { dot11PhyOperationEntry 2 }

dot11TempType OBJECT-TYPE
    SYNTAX INTEGER {tempType1(1), tempType2(2)}
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "There are different operating temperature requirements
        dependent on the anticipated environmental conditions. This
        attribute describes the current PHY's operating temperature
        range capability. Currently defined values and their
        corresponding temperature ranges are:

            Type 1 = X'01'-Commercial range of 0 to 40 degrees C,
            Type 2 = X'02'-Industrial range of -30 to 70 degrees C."

 ::= { dot11PhyOperationEntry 3 }

-- *****
-- *      End of dot11PhyOperation TABLE
-- *****

```

```

-- *****
-- *   dot11PhyAntenna  TABLE
-- *****
dot11PhyAntennaTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyAntennaEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "Group of attributes for PhyAntenna.  Implemented as a
        table indexed on ifIndex to allow for multiple instances on
        an agent."

    ::= { dot11phy 2 }

dot11PhyAntennaEntry OBJECT-TYPE
    SYNTAX Dot11PhyAntennaEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "An entry in the dot11PhyAntenna Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry.  Interface tables in this MIB module are indexed
        by ifIndex."
    INDEX {ifIndex}
    ::= { dot11PhyAntennaTable 1 }

Dot11PhyAntennaEntry ::= SEQUENCE {
    dot11CurrentTxAntenna      Integer32,
    dot11DiversitySupport     INTEGER,
    dot11CurrentRxAntenna     Integer32 }

dot11CurrentTxAntenna OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION

        "The current antenna being used to transmit.  This value
        is one of the values appearing in dot11SupportedTxAntenna.
        This may be used by a management agent to control which
        antenna is used for transmission."

    ::= { dot11PhyAntennaEntry 1 }

dot11DiversitySupport OBJECT-TYPE
    SYNTAX INTEGER {fixedlist(1), notsupported(2), dynamic(3)}
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

        "This implementation's support for diversity, encoded as:

        X'01'-diversity is available and is performed over the fixed
        list of antennas defined in dot11DiversitySelectionRx.

        X'02'-diversity is not supported."

```

```

X'03'-diversity is supported and control of diversity is also
available, in which case the attribute
dot11DiversitySelectionRx can be dynamically modified by the
LME."
 ::= { dot11PhyAntennaEntry 2 }

dot11CurrentRxAntenna OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION

        "The current antenna being used to receive, if the
        dot11 DiversitySupport indicates that diversity is not
        supported. The selected antenna shall be one of
        the antennae marked for receive in the dot11AntennasListTable. "

    ::= { dot11PhyAntennaEntry 3 }

-- *****
-- *      End of dot11PhyAntenna  TABLE
-- *****

-- *****
-- *      dot11PhyTxPower  TABLE
-- *****

dot11PhyTxPowerTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyTxPowerEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "Group of attributes for dot11PhyTxPowerTable. Implemented
        as a table indexed on STA ID to allow for multiple
        instances on an Agent."

    ::= { dot11phy 3}

dot11PhyTxPowerEntry OBJECT-TYPE
    SYNTAX Dot11PhyTxPowerEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "An entry in the dot11PhyTxPower Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."
    INDEX {ifIndex}
    ::= { dot11PhyTxPowerTable 1 }

Dot11PhyTxPowerEntry ::= SEQUENCE {
    dot11NumberSupportedPowerLevels  INTEGER,
    dot11TxPowerLevel1                INTEGER,
    dot11TxPowerLevel2                INTEGER,
    dot11TxPowerLevel3                INTEGER,
    dot11TxPowerLevel4                INTEGER,
    dot11TxPowerLevel5                INTEGER,

```



```

dot11TxPowerLevel6          INTEGER,
dot11TxPowerLevel7          INTEGER,
dot11TxPowerLevel8          INTEGER,
dot11CurrentTxPowerLevel    INTEGER}

```

```

dot11NumberSupportedPowerLevels OBJECT-TYPE
    SYNTAX INTEGER (1..8)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of power levels supported by the PMD.
        This attribute can have a value of 1 to 8."
    ::= { dot11PhyTxPowerEntry 1 }

```

```

dot11TxPowerLevel1 OBJECT-TYPE
    SYNTAX INTEGER (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The transmit output power for LEVEL1 in mW.
        This is also the default power level."
    ::= { dot11PhyTxPowerEntry 2 }

```

```

dot11TxPowerLevel2 OBJECT-TYPE
    SYNTAX INTEGER (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The transmit output power for LEVEL2 in mW."
    ::= { dot11PhyTxPowerEntry 3 }

```

```

dot11TxPowerLevel3 OBJECT-TYPE
    SYNTAX INTEGER (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The transmit output power for LEVEL3 in mW."
    ::= { dot11PhyTxPowerEntry 4 }

```

```

dot11TxPowerLevel4 OBJECT-TYPE
    SYNTAX INTEGER (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The transmit output power for LEVEL4 in mW."
    ::= { dot11PhyTxPowerEntry 5 }

```

```

dot11TxPowerLevel5 OBJECT-TYPE
    SYNTAX INTEGER (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The transmit output power for LEVEL5 in mW."
    ::= { dot11PhyTxPowerEntry 6 }

```

```

dot11TxPowerLevel6 OBJECT-TYPE
    SYNTAX INTEGER (0..10000)
    MAX-ACCESS read-only
    STATUS current

```

```

DESCRIPTION
    "The transmit output power for LEVEL6 in mW."
 ::= { dot11PhyTxPowerEntry 7 }

dot11TxPowerLevel7 OBJECT-TYPE
    SYNTAX INTEGER (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The transmit output power for LEVEL7 in mW."
 ::= { dot11PhyTxPowerEntry 8 }

dot11TxPowerLevel8 OBJECT-TYPE
    SYNTAX INTEGER (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The transmit output power for LEVEL8 in mW."
 ::= { dot11PhyTxPowerEntry 9 }

dot11CurrentTxPowerLevel OBJECT-TYPE
    SYNTAX INTEGER (1..8)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The TxPowerLevel N currently being used to transmit data.
        Some PHYs also use this value to determine the receiver
        sensitivity requirements for CCA."
 ::= { dot11PhyTxPowerEntry 10 }

-- *****
-- *      End of dot11PhyTxPower TABLE
-- *****

-- *****
-- *      dot11PhyFHSS TABLE
-- *****
dot11PhyFHSSTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyFHSSEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group of attributes for dot11PhyFHSSTable.  Implemented as a
        table indexed on STA ID to allow for multiple instances on
        an Agent."
 ::= { dot11phy 4 }

dot11PhyFHSSEntry OBJECT-TYPE
    SYNTAX Dot11PhyFHSSEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyFHSS Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry.  Interface tables in this MIB module are indexed
        by ifIndex."

```

```

    INDEX {ifIndex}
 ::= { dot11PhyFHSSTable 1 }

Dot11PhyFHSSEntry ::= SEQUENCE {
    dot11HopTime                INTEGER,
    dot11CurrentChannelNumber   INTEGER,
    dot11MaxDwellTime           INTEGER,
    dot11CurrentDwellTime       INTEGER,
    dot11CurrentSet             INTEGER,
    dot11CurrentPattern         INTEGER,
    dot11CurrentIndex           INTEGER}

dot11HopTime OBJECT-TYPE
    SYNTAX INTEGER (224)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The time in microseconds for the PMD to change from
        channel 2 to channel 80"
 ::= { dot11PhyFHSSEntry 1 }

dot11CurrentChannelNumber OBJECT-TYPE
    SYNTAX INTEGER (0..99)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The current channel number of the frequency output by the RF
        synthesizer"
 ::= { dot11PhyFHSSEntry 2 }

dot11MaxDwellTime OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The maximum time in TU that the transmitter
        is permitted to operate on a single channel."
 ::= { dot11PhyFHSSEntry 3 }

dot11CurrentDwellTime OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The current time in TU that the transmitter shall operate
        on a single channel, as set by the MAC. Default is 19 TU."
 ::= { dot11PhyFHSSEntry 4 }

dot11CurrentSet OBJECT-TYPE
    SYNTAX INTEGER (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The current set of patterns the PHY
        LME is using to determine the hopping sequence. "
 ::= { dot11PhyFHSSEntry 5 }

dot11CurrentPattern OBJECT-TYPE
    SYNTAX INTEGER (0..255)

```

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "The current pattern the PHY LME is
    using to determine the hop sequence."
 ::= { dot11PhyFHSSEntry 6 }

dot11CurrentIndex OBJECT-TYPE
    SYNTAX INTEGER (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The current index value the PHY LME is using to determine
        the CurrentChannelNumber."

 ::= { dot11PhyFHSSEntry 7 }

-- *****
-- *      End of dot11PhyFHSS  TABLE
-- *****

-- *****
-- *      dot11PhyDSSSEntry  TABLE
-- *****
dot11PhyDSSSTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyDSSSEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry of attributes for dot11PhyDSSSEntry.  Implemented as a
        table indexed on ifIndex allow for multiple instances on
        an Agent."

 ::= { dot11phy 5 }

dot11PhyDSSSEntry OBJECT-TYPE
    SYNTAX Dot11PhyDSSSEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyDSSSEntry Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry.  Interface tables in this MIB module are indexed
        by ifIndex."
    INDEX {ifIndex}
 ::= { dot11PhyDSSSTable 1 }

Dot11PhyDSSSEntry ::= SEQUENCE {
    dot11CurrentChannel    INTEGER,
    dot11CCAModeSupported  INTEGER,
    dot11CurrentCCAMode    INTEGER,
    dot11EDThreshold       Integer32}

dot11CurrentChannel OBJECT-TYPE
    SYNTAX INTEGER (1..14)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION

```

```

        "The current operating frequency channel of the DSSS
        PHY. Valid channel numbers are as defined in 15.4.6.2"
 ::= { dot11PhyDSSSEntry 1 }

dot11CCAModeSupported OBJECT-TYPE
    SYNTAX INTEGER (1..7)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "dot11CCAModeSupported is a bit-significant value, representing all of the
        CCA modes supported by the PHY. Valid values are:

            energy detect only (ED_ONLY) = 01,
            carrier sense only (CS_ONLY) = 02,
            carrier sense and energy detect (ED_and_CS)= 04

or the logical sum of any of these values."
 ::= { dot11PhyDSSSEntry 2 }

dot11CurrentCCAMode OBJECT-TYPE
    SYNTAX INTEGER {edonly(1), csonly(2), edandcs(4)}
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The current CCA method in operation. Valid values are:
            energy detect only (edonly) = 01,
            carrier sense only (csonly) = 02,
            carrier sense and energy detect (edandcs)= 04."
 ::= { dot11PhyDSSSEntry 3 }

dot11EDThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The current Energy Detect Threshold being used by the DSSS PHY."
 ::= { dot11PhyDSSSEntry 4 }

-- *****
-- * End of dot11PhyDSSSEntry TABLE
-- *****

-- *****
-- * dot11PhyIR TABLE
-- *****
dot11PhyIRTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyIREntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group of attributes for dot11PhyIRTable. Implemented as a
        table indexed on ifIndex to allow for multiple instances on
        an Agent."

 ::= { dot11phy 6 }

dot11PhyIREntry OBJECT-TYPE
    SYNTAX Dot11PhyIREntry
    MAX-ACCESS not-accessible

```

```

STATUS current
DESCRIPTION
    "An entry in the dot11PhyIR Table.

    ifIndex - Each 802.11 interface is represented by an
    ifEntry. Interface tables in this MIB module are indexed
    by ifIndex."
INDEX {ifIndex}
 ::= { dot11PhyIRTable 1 }

Dot11PhyIREntry ::= SEQUENCE {
    dot11CCAWatchdogTimerMax      Integer32,
    dot11CCAWatchdogCountMax     Integer32,
    dot11CCAWatchdogTimerMin     Integer32,
    dot11CCAWatchdogCountMin     Integer32}

dot11CCAWatchdogTimerMax OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This parameter, together with CCAWatchdogCountMax,
        determines when energy detected in the channel can be
        ignored."

 ::= { dot11PhyIREntry 1 }

dot11CCAWatchdogCountMax OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This parameter, together with CCAWatchdogTimerMax,
        determines when energy detected in the channel can be
        ignored."

 ::= { dot11PhyIREntry 2 }

dot11CCAWatchdogTimerMin OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The minimum value to which CCAWatchdogTimerMax can be
        set."

 ::= { dot11PhyIREntry 3 }

dot11CCAWatchdogCountMin OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The minimum value to which CCAWatchdogCount can be set."

 ::= { dot11PhyIREntry 4 }

-- *****
-- *      End of dot11PhyIR TABLE
-- *****

```

```

-- *****
-- *      dot11RegDomainsSupported TABLE
-- *****
dot11RegDomainsSupportedTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11RegDomainsSupportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "There are different operational requirements dependent on
        the regulatory domain. This attribute list describes the
        regulatory domains the PLCP and PMD support in this
        implementation. Currently defined values and their
        corresponding Regulatory Domains are:

        FCC (USA) = X'10', DOC (Canada) = X'20', ETSI (most of
        Europe) = X'30', Spain = X'31', France = X'32', MKK
        (Japan) = X'40' "

    ::= { dot11phy 7 }

dot11RegDomainsSupportEntry OBJECT-TYPE
    SYNTAX Dot11RegDomainsSupportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11RegDomainsSupport Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."
    INDEX { ifIndex, dot11RegDomainsSupportIndex }
    ::= { dot11RegDomainsSupportedTable 1 }

Dot11RegDomainsSupportEntry ::= SEQUENCE {
    dot11RegDomainsSupportIndex Integer32,
    dot11RegDomainsSupportValue INTEGER }

dot11RegDomainsSupportIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances
        of the columnar objects in the RegDomainsSupport Table."
    ::= { dot11RegDomainsSupportEntry 1 }

dot11RegDomainsSupportValue OBJECT-TYPE

    SYNTAX INTEGER { fcc(16), doc(32), etsi(48), spain (49), france
    (50), mkk (64) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "There are different operational requirements dependent on
        the regulatory domain. This attribute list describes the
        regulatory domains the PLCP and PMD support in this
        implementation. Currently defined values and their
        corresponding Regulatory Domains are:

```

```

        FCC (USA) = X'10', DOC (Canada) = X'20', ETSI (most of
        Europe) = X'30', Spain = X'31', France = X'32', MKK
        (Japan) = X'40' "
 ::= { dot11RegDomainsSupportEntry 2 }

-- *****
-- *      End of dot11RegDomainsSupported TABLE
-- *****

-- *****
-- *      dot11AntennasList TABLE
-- *****
dot11AntennasListTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11AntennasListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table represents the list of antennae. An
        antenna can be marked to be capable of transmitting,
        receiving, and/or for participation in receive diversity. Each
        entry in this table represents a single antenna with
        its properties. The maximum number of antennae that can
        be contained in this table is 255."
 ::= { dot11phy 8 }

dot11AntennasListEntry OBJECT-TYPE
    SYNTAX Dot11AntennasListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11AntennasListTable,
        representing the properties of a single antenna.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."
    INDEX {ifIndex, dot11AntennaListIndex}
 ::= { dot11AntennasListTable 1 }

Dot11AntennasListEntry ::= SEQUENCE {
    dot11AntennaListIndex      Integer32,
    dot11SupportedTxAntenna    TruthValue,
    dot11SupportedRxAntenna    TruthValue,
    dot11DiversitySelectionRx  TruthValue }

dot11AntennaListIndex OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The unique index of an antenna which is
        used to identify the columnar objects in
        the dot11AntennasList Table."
 ::= { dot11AntennasListEntry 1 }

dot11SupportedTxAntenna OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current

```



```

DESCRIPTION
    "When true, this object indicates that the
    antenna represented by dot11AntennaIndex
    can be used as a transmit antenna."

 ::= { dot11AntennasListEntry 2 }

dot11SupportedRxAntenna OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "When true, this object indicates that the
        antenna represented by the dot11AntennaIndex
        can be used as a receive antenna."

 ::= { dot11AntennasListEntry 3 }

dot11DiversitySelectionRx OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "When true, this object indicates that the
        antenna represented by dot11AntennaIndex can
        be used for receive diversity. This object
        may only be true if the antenna can be used
        as a receive antenna, as indicated by
        dot11SupportedRxAntenna."

 ::= { dot11AntennasListEntry 4 }

-- *****
-- *      End of dot11AntennasList TABLE
-- *****

-- *****
-- *      SupportedDataRatesTx TABLE
-- *****

dot11SupportedDataRatesTxTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11SupportedDataRatesTxEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "The Transmit bit rates supported by the PLCP and PMD,
        represented by a count from X'02-X'7f, corresponding to data
        rates in increments of 500Kb/s from 1 Mbit/s to 63.5 Mbit/s subject
        to limitations of each individual PHY."

 ::= { dot11phy 9 }

dot11SupportedDataRatesTxEntry OBJECT-TYPE
    SYNTAX Dot11SupportedDataRatesTxEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

        "An Entry (conceptual row) in the dot11SupportedDataRatesTx

```

Table.

ifIndex - Each 802.11 interface is represented by an ifEntry. Interface tables in this MIB module are indexed by ifIndex."

INDEX {ifIndex, dot11SupportedDataRatesTxIndex}

::= { dot11SupportedDataRatesTxTable 1 }

Dot11SupportedDataRatesTxEntry ::= SEQUENCE {
 dot11SupportedDataRatesTxIndex Integer32,
 dot11SupportedDataRatesTxValue Integer32}

dot11SupportedDataRatesTxIndex OBJECT-TYPE

SYNTAX Integer32 (1..8)
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"Index object which identifies which data rate to access.
 Range is 1..8."

::= { dot11SupportedDataRatesTxEntry 1 }

dot11SupportedDataRatesTxValue OBJECT-TYPE

SYNTAX Integer32 (2..127)
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The Transmit bit rates supported by the PLCP and PMD,
 represented by a count from X'02-X'7f, corresponding to data
 rates in increments of 500Kb/s from 1 Mbit/s to 63.5 Mbit/s subject
 to limitations of each individual PHY."

::= { dot11SupportedDataRatesTxEntry 2 }

-- *****
 -- * End of dot11SupportedDataRatesTx TABLE
 -- *****

-- *****
 -- * SupportedDataRatesRx TABLE
 -- *****

dot11SupportedDataRatesRxTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot11SupportedDataRatesRxEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"The receive bit rates supported by the PLCP and PMD,
 represented by a count from X'002-X'7f, corresponding to data
 rates in increments of 500Kb/s from 1 Mbit/s to 63.5 Mbit/s."

::= { dot11phy 10 }

dot11SupportedDataRatesRxEntry OBJECT-TYPE

SYNTAX Dot11SupportedDataRatesRxEntry
 MAX-ACCESS not-accessible

```

STATUS current
DESCRIPTION

    "An Entry (conceptual row) in the
    dot11SupportedDataRatesRx Table.

        ifIndex - Each 802.11 interface is represented by an
        ifEntry. Interface tables in this MIB module are indexed
        by ifIndex."

INDEX {ifIndex, dot11SupportedDataRatesRxIndex}

 ::= { dot11SupportedDataRatesRxTable 1 }

Dot11SupportedDataRatesRxEntry ::= SEQUENCE {
    dot11SupportedDataRatesRxIndex Integer32,
    dot11SupportedDataRatesRxValue Integer32}

dot11SupportedDataRatesRxIndex OBJECT-TYPE
    SYNTAX Integer32 (1..8)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index object which identifies which data rate to access.
        Range is 1..8."
 ::= { dot11SupportedDataRatesRxEntry 1 }

dot11SupportedDataRatesRxValue OBJECT-TYPE
    SYNTAX Integer32 (2..127)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The receive bit rates supported by the PLCP and PMD,
        represented by a count from X'02-X'7f, corresponding to data
        rates in increments of 500 Kb/s from 1 Mbit/s to 63.5 Mbit/s."
 ::= { dot11SupportedDataRatesRxEntry 2 }

-- *****
-- *      End of dot11SupportedDataRatesRx TABLE
-- *****

-- *****
-- *      conformance information
-- *****

dot11Conformance OBJECT IDENTIFIER ::= { ieee802dot11 5 }
dot11Groups OBJECT IDENTIFIER ::= { dot11Conformance 1 }
dot11Compliances OBJECT IDENTIFIER ::= { dot11Conformance 2 }

-- *****
-- *      compliance statements
-- *****

dot11Compliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for SNMPv2 entities
        that implement the IEEE 802.11 MIB."

MODULE -- this module
    MANDATORY-GROUPS {

```

```

dot11SMTbase2,
dot11MACbase, dot11CountersGroup,
dot11SmtAuthenticationAlgorithms,
dot11ResourceTypeID, dot11PhyOperationComplianceGroup }

GROUP dot11PhyDSSSComplianceGroup
DESCRIPTION
    "Implementation of this group is required when object
dot11PHYType has the value of dsss. This group is
mutually exclusive with the groups dot11PhyIRComplianceGroup and
dot11PhyFHSSComplianceGroup."

GROUP dot11PhyIRComplianceGroup
DESCRIPTION
    "Implementation of this group is required when object
dot11PHYType has the value of irbaseband. This group is
mutually exclusive with the groups dot11PhyDSSSComplianceGroup and
dot11PhyFHSSComplianceGroup."

GROUP dot11PhyFHSSComplianceGroup
DESCRIPTION
    "Implementation of this group is required when object
dot11PHYType has the value of fhss. This group is
mutually exclusive with the groups dot11PhyDSSSComplianceGroup and
dot11PhyIRComplianceGroup."

-- OPTIONAL-GROUPS { dot11SMTprivacy, dot11MACStatistics,
-- dot11PhyAntennaComplianceGroup, dot11PhyTxPowerComplianceGroup,
-- dot11PhyRegDomainsSupportGroup,
-- dot11PhyAntennasListGroup, dot11PhyRateGroup }
--

::= { dot11Compliances 1 }

-- *****
-- * Groups - units of conformance
-- *****
dot11SMTbase OBJECT-GROUP
OBJECTS { dot11StationID, dot11MediumOccupancyLimit,
dot11CFPollable,
dot11CFPPeriod,
dot11CFPMaxDuration,
dot11AuthenticationResponseTimeOut,
dot11PrivacyOptionImplemented,
dot11PowerManagementMode,
dot11DesiredSSID, dot11DesiredBSSType,
dot11OperationalRateSet,
dot11BeaconPeriod, dot11DTIMPeriod,
dot11AssociationResponseTimeOut
}
STATUS deprecated
DESCRIPTION
    "The SMT object class provides the necessary support at the
STA to manage the processes in the STA such that the STA may
work cooperatively as a part of an IEEE 802.11 network."
::= {dot11Groups 1 }

```

```

dot11SMTprivacy OBJECT-GROUP
  OBJECTS { dot11PrivacyInvoked,
            dot11WEPKeyMappingLength, dot11ExcludeUnencrypted,
            dot11WEPICVErrorCount , dot11WEPExcludedCount ,
            dot11WEPDefaultKeyID,
            dot11WEPDefaultKeyValue,
            dot11WEPKeyMappingWEPOn,
            dot11WEPKeyMappingValue , dot11WEPKeyMappingAddress,
            dot11WEPKeyMappingStatus }

  STATUS current
  DESCRIPTION
    "The SMTPrivacy package is a set of attributes that shall be
    present if WEP is implemented in the STA."
  ::= {dot11Groups 2 }

dot11MACbase OBJECT-GROUP

  OBJECTS { dot11MACAddress, dot11Address,
            dot11GroupAddressesStatus,
            dot11RTSThreshold, dot11ShortRetryLimit,
            dot11LongRetryLimit, dot11FragmentationThreshold,
            dot11MaxTransmitMSDULifetime,
            dot11MaxReceiveLifetime, dot11ManufacturerID,
            dot11ProductID

            }

  STATUS current
  DESCRIPTION
    "The MAC object class provides the necessary support for the
    access control, generation, and verification of frame check
    sequences, and proper delivery of valid data to upper
    layers."
  ::= {dot11Groups 3 }

dot11MACStatistics OBJECT-GROUP

  OBJECTS { dot11RetryCount, dot11MultipleRetryCount,
            dot11RTSSuccessCount, dot11RTSFailureCount,
            dot11ACKFailureCount, dot11FrameDuplicateCount }

  STATUS current
  DESCRIPTION
    "The MACStatistics package provides extended statistical
    information on the operation of the MAC. This
    package is completely optional."
  ::= {dot11Groups 4 }

dot11ResourceTypeID OBJECT-GROUP
  OBJECTS { dot11ResourceTypeIDName, dot11manufacturerOUI,
            dot11manufacturerName, dot11manufacturerProductName,
            dot11manufacturerProductVersion }
  STATUS current
  DESCRIPTION
    "Attributes used to identify a STA, its manufacturer,
    and various product names and versions."
  ::= {dot11Groups 5 }

```

```

dot11SmtAuthenticationAlgorithms OBJECT-GROUP
    OBJECTS { dot11AuthenticationAlgorithm,
              dot11AuthenticationAlgorithmsEnable }
    STATUS current
    DESCRIPTION
        "Authentication Algorithm Table."
    ::= { dot11Groups 6 }

dot11PhyOperationComplianceGroup OBJECT-GROUP
    OBJECTS { dot11PHYType, dot11CurrentRegDomain, dot11TempType }
    STATUS current
    DESCRIPTION
        "PHY layer operations attributes."
    ::= { dot11Groups 7 }

dot11PhyAntennaComplianceGroup OBJECT-GROUP
    OBJECTS {dot11CurrentTxAntenna, dot11DiversitySupport,
            dot11CurrentRxAntenna }
    STATUS current
    DESCRIPTION
        "Attributes for Data Rates for IEEE 802.11."
    ::= { dot11Groups 8 }

dot11PhyTxPowerComplianceGroup OBJECT-GROUP
    OBJECTS {dot11NumberSupportedPowerLevels, dot11TxPowerLevel1,
            dot11TxPowerLevel2, dot11TxPowerLevel3, dot11TxPowerLevel4,
            dot11TxPowerLevel5, dot11TxPowerLevel6, dot11TxPowerLevel7,
            dot11TxPowerLevel8, dot11CurrentTxPowerLevel }
    STATUS current
    DESCRIPTION
        "Attributes for Control and Management of transmit power."
    ::= { dot11Groups 9 }

dot11PhyFHSSComplianceGroup OBJECT-GROUP

    OBJECTS {dot11HopTime, dot11CurrentChannelNumber, dot11MaxDwellTime,
            dot11CurrentDwellTime, dot11CurrentSet, dot11CurrentPattern,
            dot11CurrentIndex}
    STATUS current
    DESCRIPTION
        "Attributes that configure the Frequency Hopping for IEEE
        802.11."
    ::= { dot11Groups 10 }

dot11PhyDSSSComplianceGroup OBJECT-GROUP

    OBJECTS {dot11CurrentChannel, dot11CCAModeSupported,
            dot11CurrentCCAMode, dot11EDThreshold}
    STATUS current
    DESCRIPTION
        "Attributes that configure the DSSS for IEEE 802.11."
    ::= { dot11Groups 11 }

dot11PhyIRComplianceGroup OBJECT-GROUP
    OBJECTS {dot11CCAWatchdogTimerMax, dot11CCAWatchdogCountMax,
            dot11CCAWatchdogTimerMin, dot11CCAWatchdogCountMin}

```

```

        STATUS current
        DESCRIPTION
            "Attributes that configure the baseband IR for IEEE 802.11."
        ::= { dot11Groups 12 }

dot11PhyRegDomainsSupportGroup OBJECT-GROUP
    OBJECTS { dot11RegDomainsSupportValue}
    STATUS current
    DESCRIPTION
        "Attributes that specify the supported Regulation Domains."
    ::= { dot11Groups 13}

dot11PhyAntennasListGroup OBJECT-GROUP
    OBJECTS { dot11SupportedTxAntenna,
              dot11SupportedRxAntenna, dot11DiversitySelectionRx }
    STATUS current
    DESCRIPTION
        "Attributes that specify the supported Regulation Domains."
    ::= { dot11Groups 14 }

dot11PhyRateGroup OBJECT-GROUP
    OBJECTS {dot11SupportedDataRatesTxValue,
            dot11SupportedDataRatesRxValue
    }
    STATUS current
    DESCRIPTION
        "Attributes for Data Rates for IEEE 802.11."
    ::= { dot11Groups 15 }

dot11CountersGroup OBJECT-GROUP
    OBJECTS {
        dot11TransmittedFragmentCount ,
        dot11MulticastTransmittedFrameCount ,
        dot11FailedCount, dot11ReceivedFragmentCount,
        dot11MulticastReceivedFrameCount ,
        dot11FCSErrorCount,
        dot11WEPUndecryptableCount,
        dot11TransmittedFrameCount }

    STATUS current
    DESCRIPTION
        "Attributes from the dot11CountersGroup that are not described
        in the dot11MACStatistics group. These objects are
        mandatory."
    ::= {dot11Groups 16 }

dot11NotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS { dot11Disassociate,
                   dot11Deauthenticate,
                   dot11AuthenticateFail }
    STATUS current
    DESCRIPTION
        "IEEE 802.11 notifications"
    ::= { dot11Groups 17 }

dot11SMTbase2 OBJECT-GROUP

```

```

OBJECTS { dot11MediumOccupancyLimit,
          dot11CFPollable,
          dot11CFPPeriod,
          dot11CFPMaxDuration,
          dot11AuthenticationResponseTimeOut,
          dot11PrivacyOptionImplemented,
          dot11PowerManagementMode,
          dot11DesiredSSID, dot11DesiredBSSType,
          dot11OperationalRateSet,
          dot11BeaconPeriod, dot11DTIMPeriod,
          dot11AssociationResponseTimeOut,
          dot11DisassociateReason,
          dot11DisassociateStation,
          dot11DeauthenticateReason,
          dot11DeauthenticateStation,
          dot11AuthenticateFailStatus,
          dot11AuthenticateFailStation
        }
STATUS current
DESCRIPTION
    "The SMTbase2 object class provides the necessary support at the
    STA to manage the processes in the STA such that the STA may
    work cooperatively as a part of an IEEE 802.11 network."
 ::= {dot11Groups 18 }

-- *****
-- *   End of 80211 MIB
-- *****
END

```


Annex E

(informative)

Bibliography

E.1 General

[B1] ANSI Z136.1-1993, American National Standard for the Safe Use of Lasers.

[B2] IEC 60825-1 (1993), Safety of laser products—Part 1: Equipment classification, requirements and user's guide.

[B3] IEEE Std 802.10-1998, IEEE Standards for Local and Metropolitan Area Networks: Interoperable LAN/MAN Security (SILS).

[B4] Schneier, Bruce, *Applied Cryptography, Protocols, Algorithms and Source Code in C*. New York: Wiley, 1994.

E.2 Specification and description language (SDL) documentation

[B5] Belina, Ferenc, Dieter Hogrefe, and Amardeo Sarma, *SDL with Applications from Protocol Specification*. UK: Prentice Hall Europe, Hertfordshire, 1991.⁸

[B6] Ellsberger, Jan, Dieter Hogrefe, and Amardeo Sarma, *SDL, Formal Object-Oriented Language for Communicating Systems*. Hertfordshire, UK: Prentice Hall Europe, 1997.⁹

[B7] Faergemand, Ove and Anders Olsen, "New Features in SDL-92," *SDL Newsletter* (ISSN 1023-7151), no. 16 (May 1993), pp. 10–29. Also available online at <http://www.tdr.dk/public/SDL/SDL.html>.¹⁰

[B8] Olsen, Anders, Ove Faergemand, Birger Moller-Pedersen, Rick Reed, and T. R. W. Smith, *Systems Engineering Using SDL-92*. Amsterdam, the Netherlands: Elsevier Science B.V., 1994.¹¹

⁸An introductory text on SDL, also useful as a language reference (for SDL-88).

⁹A recently published book, which appears to be the most comprehensive single-volume introduction and reference for SDL-92, including its object-oriented extensions.

¹⁰This provides a summary of the changes from SDL-88 to SDL-92.

¹¹A detailed guide to using SDL-92, including a thorough explanation of abstract data type mechanism and SDL combined with ASN.1 (ITU-T Recommendation Z.105).