802.11r™

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

Amendment 2: Fast Basic Service Set (BSS) Transition

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

15 July 2008

IEEE Std 802.11r™-2008
(Amendment to
IEEE Std 802.11™-2007
as amended by
IEEE Std 802.11k™-2008)

IEEE Std 802.11r™-2008
(Amendment to
IEEE Std 802.11™-2007
as amended by
IEEE Std 802.11k™-2008)

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

Amendment 2: Fast Basic Service Set (BSS) Transition

Sponsored by

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 9 May 2008

IEEE-SA Standards Board

Abstract: This amendment specifies the extensions to IEEE Std 802.11-2007 for wireless local area networks (WLANs) providing mechanisms for fast basic service set (BSS) transition.
Keywords: LAN, local area network, wireless LAN, WLAN

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

PDF: ISBN 973-07381-5442-0 STD95794
Print: ISBN 973-07381-5423-7 STDPD95794

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

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

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

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

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

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

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

Introduction

This introduction is not part of IEEE Std 802.11-2007, IEEE Standard for information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 2: Fast Basic Service Set (BSS) Transition.

This amendment describes mechanisms that minimize the amount of time data connectivity is lost between the station (STA) and the distribution system (DS) during a basic service set (BSS) transition. The STA determines when to transition and to which access point (AP) to transition based on a number of factors, some of which may be out of the scope of this standard.

The following summarizes the typical behavior of the non-AP STA and AP when a transition occurs without fast BSS transition (FT) services:

— The STA uses scanning or neighbor reports to discover APs available for transition.
— The STA chooses a target AP and performs an IEEE 802.11 authentication exchange with that AP; typically this exchange will be an "open auth" exchange. During this time, the STA may still exchange data with the DS through its current AP.
— The STA sends a (Re)Association frame to establish a connection at the target AP.
— In a robust security network (RSN), the STA and the AP then generate and confirm matching temporal keys based on a preshared key (PSK) or an IEEE 802.1X authentication (which could be through an earlier preauthentication or key caching).
— In an RSN, the STA and AP install the keys and start to exchange data with the DS.
— For a quality of service (QoS) STA connected to a QoS AP, the STA may then request QoS resources by issuing one or more ADDTS (add traffic stream) requests.

The FT mechanism allows a STA to establish security and/or QoS state at the target AP prior to or during reassociation, avoiding delays in connecting to the DS after transition. The overall changes to the protocol do not introduce any new security vulnerabilities beyond the current IEEE 802.11 standard and its amendments. The FT mechanism preserves the behavior of legacy STAs and APs.

The FT time is the total transition time that starts after the receipt of the last acknowledged data frame sent within an originating BSS and ends after the receipt of the first acknowledged data frame sent within the destination BSS, while the non-AP STA transitions from one BSS to another using the FT mechanisms.

This amendment addresses solutions to two classes of network infrastructures from a QoS perspective: one where the transition-enabled AP is willing to provision QoS resources at reassociation time; and another where the AP needs to reserve the network infrastructure resources before transitioning.

This amendment does not specifically address the solution of when or where a STA will roam. Other tools give the STA information that could be used in making this decision.

IEEE 802.11 enables the AP to convey a BSS Load information element in Probe Response and Beacon frames. The BSS Load information element has three fields that indicate the number of associated STAs, the channel utilization for the BSS, and the available admission capacity. These QoS BSS metrics give information on the AP's ability to accept new QoS streams.

IEEE 802.11 defines the neighbor reports, which can assist in optimizing scanning.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association website at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at http://standards.ieee.org.

Errata

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

Interpretations

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

Patents

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

applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates, terms, and conditions of the license agreements offered by patent holders or patent applicants. Further information may be obtained from the IEEE Standards Department.

Participants

At the time this amendment was sent to sponsor ballot, the IEEE 802.11 Working Group had the following officers:

Stuart J. Kerry, *Chair*
Al Petrick, *Vice Chair, Treasurer and Chair, Task Group mb*
Harry R. Worstell, *Vice Chair*
Stephen McCann, *Secretary and Chair, Publicity Standing Committee*
Teik-Kheong Tan, *Chair, Wireless Next Generation Standing Committee*
Terry L. Cole, *Technical Editor and Assigned Number Authority*

Richard H. Paine, *Chair, Task Group k*
Bruce P. Kraemer, *Chair, Task Group n and Co-Chair IMT-Advanced Ad hoc Committee*
Sheung Li, *Vice Chair, Task Group n*
Lee Armstrong, *Chair, Task Group p*
Donald E. Eastlake III, *Chair, Task Group s*
Neeraj Sherma, *Chair, Task Group T*
Stephen McCann, *Chair, Task Group u*
Dorothy V. Stanley, *Chair, Task Group v & IETF Ad hoc Committee*
Jesse Walker, *Chair, Task Group w & JCT1 Ad hoc Committee*
Peter Ecclesine, *Chair, Task Group y*
Menzo Wentink, *Direct Link Setup Study Group*
Bob O'Hara, *QoS Extensions Study Group*
Ganesh Venkatesan, *Video Transport Stream Study Group*
Eldad Perahia, *Very High Throughput Study Group*
Darwin Engwer, Co-*Chair, IMT-Advanced Ad hoc Committee*

At the time this amendment was submitted to sponsor ballot, the Task Group r had the following officers:

Clint Chaplin, *Chair*
Michael Montemurro, *Secretary*
Bill Marshall, *Technical Editor*

When the IEEE 802.11 Working Group approved this amendment, the Working Group had the following membership:

Osama Aboul-Magd
Tomoko Adachi
Carlos Aldana
Thomas Alexander
Keith Amann
David Andrus
Takashi Aramaki
Sirikiat Lek Ariyavisitakul
Larry Arnett
Alex Ashley
Arthur Astrin
Malik Audeh
Geert Awater
Floyd Backes
David Bagby
Michael Bahr
Dennis Baker
Amit Bansal
John Barr
Gal Basson
Anuj Batra
Moussa Bavafa
Mathilde Benveniste
Bjorn Bjerke
Tony Braskich
George Bumiller
Alistair Buttar
Pat Calhoun
Nancy Cam-Winget
James Carlo
Pat Carson
Broady Cash
Dave Cavalcanti
Douglas Chan
Yi-Ming Chen
Hong Cheng
Paul Cheng
Aik Chindapol
Abhijit Choudhury
Liwen Chu
Frank Ciotti
W. Steven Conner
Charles Cook
Steven Crowley
David Cypher
Marc de Courville

Sabine Demel
Dee Denteneer
Susan Dickey
Yoshiharu Doi
John Dorsey
Roger Durand
Donald E. Eastlake, III
Hesham Elbakoury
Michael Ellis
Stephen Emeott
Marc Emmelmann
Darwin Engwer
Joseph Epstein
Leonid Epstein
Vinko Erceg
Lars Falk
Stefan Fechtel
Paul Feinberg
Matthew Fischer
Wayne Fisher
Michael Foegelle
Edoardo Gallizio
Matthew Gast
Sudhanshu Gaur
James Gilb
Tim Godfrey
Michelle Gong
Hrishikesh Gossain
Jeremy Gosteau
Sudheer Grandhi
Gordon Gray
Larry Green
Pratibha Gupta
Robert J. Hall
Mark Hamilton
Christopher Hansen
Daniel Harkins
Brian Hart
Amer Hassan
Myron Hattig
James Hauser
Shigenori Hayase
Kevin Hayes
Robert Heile
Eleanor Hepworth
Karl Heubaum
Guido Hiertz
Garth Hillman
Christopher Hinsz
Robert Hsieh
Wendong Hu
Jiyoung Huh
David Hunter
Yasuhiko Inoue
Marc Jalfon
Jorjeta Jetcheva
Lusheng Ji
Daniel Jiang
Jari Jokela
V. K. Jones
Padam Kafle
Carl Kain
Naveen Kakani
Srinivas Kandala
Assaf Kasher
Masato Kato
Douglas Kavner
Richard Kennedy
Stuart J. Kerry
John Ketchum
JinKyeong (Joseph) Kim
Joonsuk Kim
Kyeongsoo Kim
Tae-eun Kim
Guenter Kleindl
Jarkko Kneckt
Mark Kobayashi
Benjamin Koh
Thomas Kolze
Gopal Krishnan
Jan Kruys
Thomas Kuehnel
Christian Kuhtz
Rajendra Kumar
Rajneesh Kumar
Thomas Kurihara
Joe Kwak
Jeremy Landt
Joseph Lauer
Jehun Lee
Jin Lee
Martin Lefkowitz
Uriel Lemberger
Joseph Levy
Azman-Osman Lim
Huashih Lin
Hang Liu
Michael Livshitz
Peter Loc
Peter Lojko
Dan Lubar
Krishna Sankar Madhavan Pillai
Alastair Malarky
Majid Malek
Jouni Malinen
Mahalingam Mani
William Marshall
Sudheer Matta
Matthieu Maupetit
William McFarland
Darren McNamara
Justin McNew
Irina Medvedev
Pratik Mehta
Sven Mesecke
Klaus Meyer
Robert Miller
Hidekazu Miyoshi
Fanny Mlinarsky
Patrick Mo
Andreas Molisch
Michael Montemurro
Gabriel Montenegro
Rajendra Moorti
Hitoshi Morioka
Yuichi Morioka
James Murphy
Peter Murray
Andrew Myles
Rohit Nabar
Yukimasa Nagai
Tetsuya Nakamura
Seigo Nakao
Ravi Nalamati
Sanjiv Nanda
Partha Narasimhan
Chiu Ngo
Eero Nikula
Gunnar Nitsche
Erwin Noble
Richard Noens
Hideaki Odagiri
Bob O'Hara
Eric Ojard
Chandra Olson
Satoshi Oyama
Richard Paine
Subra Parameswaran
Xavier Perez Costa
James Petranovich
Fahd Pirzada
Masood Pirzada
Victoria Poncini
Subbu Ponnuswamy
James Portaro
Henry Ptasinski
Emily Qi
Luke Qian
Jim Raab
Vinuth Rai
Ali Raissinia
Stephen Rayment
Ivan Reede
Joe Repice
Edward Reuss
Carlos Rios
Jon Rosdahl
Kazuyuki Sakoda
Atul Salhotra
Anil Sanwalka
Nicholas J Sargologos
Ambatipudi Sastry
Vincenzo Scarpa
Donald Schultz
Erik Schylander
Huai-Rong Shao
Neeraj Sharma
Suman Sharma
Stephen Shellhammer
Ian Sherlock
Kai Shi
Donghee Shim
D. J. Shyy
Massimiliano Siti
Matt Smith
Kapil Sood
Amjad Soomro
Srinivas Sreemanthula
Robert Stafford
Dorothy Stanley
Adrian Stephens
David Stephenson

Fabrice Stevens
Carl Stevenson
Guenael Strutt
Winston Sun
Shravan Surineni
Hideyuki Suzuki
Eiji Takagi
Mineo Takai
Daisuke Takeda
Tsuyoshi Tamaki
Allan Thomson
Jerry Thrasher
Alexander Tolpin
Jason Trachewsky
Solomon Trainin
Richard Van Nee
Allert van Zelst
Prabodh Varshney
Ganesh Venkatesan
Dalton Victor
George Vlantis
Tim Wakeley
Brad Wallace
Huihui Wang
Qi Wang
Xudong Wang
Dennis Ward
Deric Waters
Filip Weytjens
Stephen Whitesell
James Worsham
Charles Wright
Akiyoshi Yagi
Katsuhiko Yamada
Masaya Yamada
Takeshi Yamamoto
Tomoya Yamaura
Yuli Yang
Chi-Hsiang Yeh
Kanji Yokohira
Seiji Yoshida
Chris Young
Artur Zaks
Jinyun Zhang
Junping Zhang
Juan-Carlos Zuniga
Johnny Zweig

Major contributions were received from the following individuals:

Bernard Aboba
Peyush Agarwal
Areg Alimian
Keith Amann
Bob Beach
Stefan Berg
Florent Bersani
Tony Braskich
Bill Burr
Pat Calhoun
Alan Carlton
Nancy Cam-Winget
Clint Chaplin
James Chen
Lily Chen
Randy Chou
Frank Ciotti
Charles Clancy
Steve Connor
Anupam Datta
Chris Durand
Jon Edney
Steve Emeott
Darwin Engwer
Stefano Faccin
Paul Funk
Wolfgang Groting
Du Hanmei
Dan Harkins
Kevin Hayes
Haixiang He
Xiaoning He
Eleanor Hepworth
Katrin Hoeper
Russ Housley
Srinivas Inguva
Marc Jalfon
Moo Ryong Jeong
Theodore Karoubalis
Toshiro Kawahara
Scott Kelly
Joe Kubler
Dirk Kuijsten
Rajneesh Kumar
Martin Lefkowitz
Jie Liang
Zhibin Lin
Peter Loc
Robert Love
Mani Mahalingam
Jouni Malinen
Bill Marshall
John C. Mitchell
Michael Montemurro
Tim Moore
Mike Moreton
Patrick Mourot
Paul Newton
Bob O'Hara
Soohong Daniel Park
Chris Polanec
Henry Ptasinski
Emily Qi
Arnab Roy
Marian Rudolf
Suresh Satapati
Ioanna Samprakou
Dan Simon
Floyd Simpson
Vishal Sinha
Matt Smith
Kapil Sood
Jeremy Spilman
Dorothy Stanley
Hui Tang
Chris Trecker
Sandy Turner
Jesse Walker
Stephen Wang
Fujio Watanabe
Jim Wendt
Michael Williams
Charles Wright
Gang Wu
Artur Zaks
Meiyuan Zhao
Juan-Carlos Zuniga
Johnny Zweig

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

Thomas Alexander
Keith Amann
Danilo Antonelli
Roger Berg
Gennaro Boggia
William Byrd
Peter J. Calderon
James Carlo
Juan Carreon
Jay Catelli
Clint Chaplin
Lidong Chen
Yi-Ming Chen
Hong Cheng
Aik Chindapol
Keith Chow
Charles Cook
Russell Dietz
Petar Djukic
Vern A. Dubendorf
Guy R. Duryee
Sourav Dutta
Donald E. Eastlake, III
Paul Eastman

Richard Eckard
Jonathan Edney
Joseph Epstein
Matthew Fischer
Wayne K. Fisher
C. Fitzgerald
Andre Fournier
Prince Francis
Avraham Freedman
Devon Gayle
Randall Groves
C. Guy
C. Hansen
Karl Heubaum
Russell Housley
David Hunter
Yasuhiko Inoue
Atsushi Ito
Raj Jain
Jari Jokela
Bobby Jose
Junghong Kao
Stuart J. Kerry
Brian Kiernan
Eunkyung Kim
Yongbum Kim
Yongho Kim
Joseph Kubler
Thomas Kurihara
Jeremy Landt
Daniel Levesque
Joseph Levy
Jan-Ray Liao
Jouni Malinen
Sudheer Matta
W. Kyle Maus
Stephen McCann
Gary Michel
R. Miller
Apurva Mody
Michael Montemurro
Jose Morales
Joseph Moran
Andrew Myles
Michael S. Newman
Richard Noens
Satoshi Obara
Bob O'Hara
Satoshi Oyama
Stephen Palm
Michael Probasco
Henry S. Ptasinski
Maximilian Riegel
Robert Robinson
Randall Safier
Osman Sakr
John Sargent
Peter Saunderson
Suman Sharma
Kapil Sood
Amjad Soomro
Srinvias Sreemanthula
Dorothy Stanley
Thomas Starai
Adrian P. Stephens
Rene Struik
Walter Struppler
Alourdes Sully
Masahiro Takagi
Solomon Trainin
Ganesh Venkatesan
John Vergis
Stanley Wang
Stephen Webb
Stephen Whitesell
Harry Worstell
Oren Yuen
Paolo Zangheri
Surong Zeng

When the IEEE-SA Standards Board approved this amendment on 9 May 2008, it had the following membership:

Robert M. Grow, *Chair*
Thomas Prevost, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Victor Berman
Richard DeBlasio
Andy Drozd
Mark Epstein
Alexander Gelman
William R. Goldbach
Arnold M. Greenspan
Kenneth S. Hanus
Jim Hughes
Richard H. Hulett
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Glenn Parsons
Ronald C. Petersen
Chuck Powers
Narayanan Ramachandran
Jon Walter Rosdahl
Robby Robson
Anne-Marie Sahazizia
Malcolm V. Thaden
Howard L. Wolfman
Don Wright

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*
Michael H. Kelly, *NIST Representative*

Michelle Turner
IEEE Standards Program Manager, Document Development

Michael K. Kipness
IEEE Standards Program Manager, Technical Program Development

CONTENTS

List of figures

List of tables

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

Amendment 2: Fast Basic Service Set (BSS) Transition

(This amendment is based on IEEE Std 802.11™-2007, as amended by IEEE Std 802.11k™-2008.)

NOTE—The editing instructions contained in this amendment define how to merge the material contained herein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in ***bold italic***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instructions. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editorial notes will not be carried over into future editions because the changes will be incorporated into the base standard.

2. Normative references

Insert the following references in alpha numeric order into Clause 2:

FIPS PUB 180-2-2002, Secure Hash Standard.

FIPS SP800-38B, Dworkin, M., "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication."

3. Definitions

Change the following definitions in Clause 3 as indicated:

3.97 pairwise master key (PMK): The ~~highest order~~ key ~~used within this standard. The PMK may be~~ derived from a key generated by an Extensible Authentication Protocol (EAP) method or ~~may be~~ obtained directly from a preshared key (PSK).

3.99 pairwise transient key (PTK): ~~A value that is derived form the pairwise master key (PMK), Authenticator address (AA), Supplicant address (SPA), Authenticator nonce (ANonce), and Supplicant nonce (SNonce) using the pseudo-random function (PRF) and that is split up into as many as five keys, i.e. temporal encryption key, two temporal message integrity code (MIC) keys, EAPOL-Key encryption key (KEK), EAPOL-Key confirmation key (KCK).~~ A concatenation of session keys derived from the pairwise master key (PMK) or from the PMK-R1. Its components include a key confirmation key (KCK), a key encryption key (KEK), and one or more temporal keys that are used to protect information exchanged over the link.

3.130 robust security network association (RSNA) key management: Key management that includes the 4-Way Handshake, the Group Key Handshake, and the PeerKey Handshake. If fast basic service set (BSS) transition (FT) is enabled, the FT 4-Way Handshake and FT authentication sequence are also included.

Insert the following new definitions in alphabetical order into Clause 3, renumbering as necessary:

3.193 fast basic service set (BSS) transition: A station (STA) movement that is from one BSS in one extended service set (ESS) to another BSS within the same ESS and that minimizes the amount of time that data connectivity is lost between the STA and the distribution system (DS).

3.194 fast basic service set (BSS) transition (FT) 4-Way Handshake: A pairwise key management protocol used during FT initial mobility domain association. This handshake confirms mutual possession of a pairwise master key, the PMK-R1, by two parties and distributes a group temporal key (GTK).

3.195 fast basic service set (BSS) transition (FT) initial mobility domain association: The first association or first reassociation procedure within a mobility domain, during which a station (STA) indicates its intention to use the FT procedures.

3.196 mobility domain: A set of basic service sets (BSSs), within the same extended service set (ESS), that support fast BSS transitions between themselves and that are identified by the set's mobility domain identifier (MDID).

3.197 mobility domain identifier (MDID): An identifier that names a mobility domain.

3.198 network access server (NAS) client: The client component of a NAS that communicates with the Authentication Server (AS).

3.199 over-the-air fast basic service set (BSS) transition (FT): An FT method in which the station (STA) communicates over a direct IEEE 802.11 link to the target access point (AP).

3.200 over-the-DS (distribution system) fast basic service set (BSS) transition (FT): An FT method in which the station (STA) communicates with the target access point (AP) via the current AP.

3.201 pairwise master key R0 (PMK-R0): The key at the first level of the fast basic service set (BSS) transition (FT) key hierarchy.

3.202 pairwise master key (PMK) R0 key holder (R0KH): The component of robust security network association (RSNA) key management of the Authenticator that is authorized to derive and hold the PMK-R0, derive the PMK-R1s, and distribute the PMK-R1s to the R1KHs.

3.203 pairwise master key (PMK) R0 key holder identifier (R0KH-ID): An identifier that names the holder of the PMK-R0 in the Authenticator.

3.204 pairwise master key R1 (PMK-R1): A key at the second level of the fast basic service set (BSS) transition (FT) key hierarchy.

3.205 pairwise master key (PMK) R1 key holder (R1KH): The component of robust security network association (RSNA) key management of the Authenticator that receives a PMK-R1 from the R0KH, holds the PMK-R1, and derives the PTKs.

3.206 pairwise master key (PMK) R1 key holder identifier (R1KH-ID): An identifier that names the holder of a PMK-R1 in the Authenticator.

3.207 pairwise master key (PMK) S0 key holder (S0KH): The component of robust security network association (RSNA) key management of the Supplicant that derives and holds the PMK-R0, derives the PMK-R1s, and provides the PMK-R1s to the S1KH.

3.208 pairwise master key (PMK) S0 key holder identifier (S0KH-ID): An identifier that names the holder of the PMK-R0 in the Supplicant.

3.209 pairwise master key (PMK) S1 key holder (S1KH): The component of robust security network association (RSNA) key management in the Supplicant that receives a PMK-R1 from the S0KH, holds the PMK-R1, and derives the PTKs.

3.210 pairwise master key (PMK) S1 key holder identifier (S1KH-ID): An identifier that names the holder of the PMK-R1 in the Supplicant.

3.211 pairwise transient key (PTK) name (PTKName): An identifier that names the PTK.

3.212 pairwise master key (PMK) R0 name (PMKR0Name): An identifier that names the PMK-R0.

3.213 pairwise master key (PMK) R1 name (PMKR1Name): An identifier that names a PMK-R1.

3.214 remote request broker (RRB): The component of the station management entity (SME) of an access point (AP) that supports fast basic service set (BSS) transitions over the distribution system (DS).

3.215 resource information container (RIC): A sequence of information elements that include resource request and response parameters.

4. Abbreviations and acronyms

Insert the following abbreviations in alphabetical order into Clause 4:

AES-128-CMAC	advanced encryption standard (with 128-bit key) cipher-based message authentication code
FT	fast BSS transition
FTAA	fast BSS transition authentication algorithm

FTIE	fast BSS transition information element
KDF	key derivation function
MDID	mobility domain identifier
MDIE	Mobility Domain information element
NAS	network access server
PMK-R0	pairwise master key, first level
PMK-R1	pairwise master key, second level
RDIE	RIC Data information element
RIC	resource information container
RRB	remote request broker
RSNIE	Robust Security Network information element
R0KH	PMK-R0 key holder in the Authenticator
R0KH-ID	PMK-R0 key holder identifier in the Authenticator
R1KH	PMK-R1 key holder in the Authenticator
R1KH-ID	PMK-R1 key holder identifier in the Authenticator
S0KH	PMK-R0 key holder in the Supplicant
S0KH-ID	PMK-R0 key holder identifier in the Supplicant
S1KH	PMK-R1 key holder in the Supplicant
S1KH-ID	PMK-R1 key holder identifier in the Supplicant
TIE	Timeout Interval information element

5. General description

5.2 Components of the IEEE 802.11 architecture

5.2.3 Distribution system (DS) concepts

5.2.3.2 Robust security network association (RSNA)

Change the first paragraph of 5.2.3.2 as follows:

An RSNA defines a number of security features in addition to wired equivalent privacy (WEP) and IEEE 802.11 authentication. These features include the following:

- Enhanced authentication mechanisms for STAs
- Key management algorithms
- Cryptographic key establishment
- An enhanced data cryptographic encapsulation mechanism, called Counter mode with Cipher-block chaining Message authentication code Protocol (CCMP), and, optionally, Temporal Key Integrity Protocol (TKIP)
- Fast basic service set (BSS) transition (FT) mechanism

Insert the following paragraph at the end of 5.2.3.2:

An RSNA using fast BSS transition relies on an external protocol to distribute keys between the pairwise master key (PMK) R0 key holder (R0KH) and PMK-R1 key holder (R1KH) Authenticator components. The requirements for this protocol are described in 11A.2.2.

5.4 Overview of the services

5.4.2 Services that support the distribution services

5.4.2.1 Mobility types

Change list item (b) in 5.4.2.1 as follows:

b) *BSS-transition:* This type is defined as a station (STA) movement from one BSS in one extended service set (ESS) to another BSS within the same ESS. A fast BSS transition is a BSS transition that establishes the state necessary for data connectivity before the reassociation rather than after the reassociation.

Insert the following paragraph after the lettered list in 5.4.2.1:

The FT Protocol provides a mechanism for a non-AP STA to perform a BSS transition between access points (APs) in a robust security network (RSN) or when quality of service (QoS) admission control is enabled in the ESS.

5.4.3 Access control and data confidentiality services

5.4.3.1 Authentication

Change the third paragraph of 5.4.3.1 as follows:

IEEE Std 802.11 defines ~~two~~three authentication methods: Open System authentication, ~~and~~ Shared Key authentication, and FT authentication. Open System authentication admits any STA to the DS. Shared Key authentication relies on WEP to demonstrate knowledge of a WEP encryption key. FT authentication relies on keys derived during the initial mobility domain association to authenticate the non-AP stations as defined in Clause 11A. The IEEE 802.11 authentication mechanism also allows definition of new authentication methods.

5.4.3.4 Key management

Change 5.4.3.4 as follows:

The enhanced data confidentiality, data authentication, and replay protection mechanisms require fresh cryptographic keys. The procedures defined in this standard provide fresh keys by means of protocols called the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol, and Group Key Handshake.

Insert the following new subclause (5.4.3.7) after 5.4.3.6:

5.4.3.7 Fast BSS transition

The FT mechanism defines a means for a non-AP STA to set up security and QoS parameters prior to reassociation to a new AP. This mechanism allows time-consuming operations to be removed from the time-critical reassociation process.

5.8 IEEE Std 802.11 and IEEE Std 802.1X-2004

5.8.1 IEEE 802.11 usage of IEEE Std 802.1X-2004

Change the second paragraph of 5.8.1 as follows:

IEEE Std 802.11 depends upon IEEE Std 802.1X-2004 and the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol, and Group Key Handshake, described in Clause 8 and Clause 11A, to establish and change cryptographic keys. Keys are established after authentication has completed. Keys may change for a variety of reasons, including expiration of an IEEE 802.1X authentication timer, key compromise, danger of compromise, or policy.

5.8.2 Infrastructure functional model overview

5.8.2.1 Authentication and key management (AKM) operations with Authentication Server (AS)

Change the second paragraph of 5.8.2.1 as follows:

A 4-Way Handshake or FT 4-Way Handshake utilizing IEEE 802.1X EAPOL-Key frames is initiated by the Authenticator to do the following:

- Confirm that a live peer holds the PMK.
- Confirm that the PMK is current.
- In the case of fast BSS transition, derive PMK-R0s and PMK-R1s.
- Derive a fresh pairwise transient key (PTK) from the PMK or, in the case of fast BSS transition, from the PMK-R1.
- Install the pairwise encryption and integrity keys into IEEE Std 802.11.
- Transport the group temporal key (GTK) and GTK sequence number from Authenticator to Supplicant and install the GTK and GTK sequence number in the STA and, if not already installed, in the AP.
- Confirm the cipher suite selection.

6. Medium access control (MAC) service definition

6.1 Overview of MAC services

6.1.2 Security services

Change the third paragraph of 6.1.2 as follows:

During the authentication exchange, both parties exchange authentication information as described in Clause 8 and Clause 11A.

7. Frame formats

7.2 Format of individual frame types

7.2.3 Management frames

7.2.3.1 Beacon frame format

Insert order 33 information field into Table 7-8:

Table 7-8—Beacon frame body

Order	Information	Notes
33	Mobility domain	The Mobility Domain information element (MDIE) is present when dot11FastBSSTransitionEnabled is set to TRUE.

7.2.3.4 Association Request frame format

Insert order 11 information field into Table 7-10:

Table 7-10—Association Request frame body

Order	Information	Notes
11	Mobility domain	The MDIE is present in an Association Request frame if dot11FastBSS-TransitionEnabled is set to TRUE and if the frame is being sent to an AP that advertised its FT capability in the MDIE in its Beacon or Probe Response frame (i.e., AP also has dot11FastBSSTransitionEnabled set to TRUE).

7.2.3.5 Association Response frame format

Insert order 10 and 11 information fields into Table 7-11:

Table 7-11—Association Response frame body

Order	Information	Notes
10	Mobility domain	An MDIE is present in an Association Response frame when dot11FastBSSTransitionEnabled is set to TRUE and this frame is a response to an Association Request frame that contained an MDIE (i.e., an FT initial mobility domain association exchange).
11	Fast BSS transition	A Fast BSS Transition information element (FTIE) is present in an Association Response frame when dot11FastBSSTransitionEnabled is set to True, dot11RSNAEnabled is set to TRUE and this frame is a response to an Association Request frame that contained an MDIE (i.e., an FT initial mobility domain association exchange in an RSN).

7.2.3.6 Reassociation Request frame format

Insert order 12, 13, and 14 information fields into Table 7-12:

Table 7-12—Reassociation Request frame body

Order	Information	Notes
12	Mobility domain	The MDIE is present in a Reassociation Request frame when dot11FastBSSTransitionEnabled is set to True and the frame is being sent to an AP that advertised its FT Capability in the MDIE in its Beacon or Probe Response frame (i.e., AP also has dot11FastBSSTransitionEnabled is set to TRUE).
13	Fast BSS transition	An FTIE is present in a Reassociation Request frame when dot11FastBSSTransitionEnabled is set to TRUE and dot11RSNAAuthenticationSuiteSelected is 00-0F-AC:3 or 00-0F-AC:4 (i.e., part of a fast BSS transition in an RSN).
14	Resource information container (RIC)	The set of information elements that formulate a RIC-Request may be present in a Reassociation Request frame when — dot11FastBSSTransitionEnabled is set to TRUE, — The FT Resource Request Protocol is not used, — The frame is being sent to an AP that advertised its FT capability in the MDIE in its Beacon or Probe Response frame (i.e., AP also has dot11FastBSSTransitionEnabled is set to TRUE), and — Either dot11RSNAAuthenticationSuiteSelected is 00-0F-AC:3 or 00-0F-AC:4 (i.e., part of a fast BSS transition in an RSN) or dot11RSNAEnabled is set to FALSE (i.e., not in an RSN).

7.2.3.7 Reassociation Response frame format

Insert order 10 through 13 information fields into Table 7-13:

Table 7-13—Reassociation Response frame body

Order	Information	Notes
10	RSN	An RSN information element (RSNIE) is present in a Reassociation Response frame when dot11FastBSSTransitionEnabled is set to True, dot11RSNAEnabled is set to TRUE and this frame is a response to a Reassociation Request frame that contained an FTIE (i.e., part of a fast BSS transition in an RSN).
11	Mobility domain	An MDIE is present in a Reassociation Response frame when dot11FastBSSTransitionEnabled is set to TRUE and this frame is a response to a Reassociation Request frame that contained an MDIE (i.e., either an FT initial mobility domain association exchange or part of a fast BSS transition).
12	Fast BSS transition	An FTIE is present in a Reassociation Response frame when dot11FastBSSTransitionEnabled is set to TRUE, dot11RSNAEnabled is set to TRUE and this frame is a response to a Reassociation Request frame that contained an MDIE (i.e., either an FT initial mobility domain association exchange or part of a fast BSS transition in an RSN).

Table 7-13—Reassociation Response frame body *(continued)*

Order	Information	Notes
13	RIC	The set of information elements that formulate a RIC-Response is present in a Reassociation Response frame when dot11FastBSSTransitionEnabled is set to TRUE and this frame is a response to a Reassociation Request frame that contained a RIC-Request.

7.2.3.9 Probe Response frame format

Insert order 31 information field into Table 7-15:

Table 7-15—Probe Response frame body

Order	Information	Notes
31	Mobility domain	The MDIE is present when dot11FastBSSTransitionEnabled is set to TRUE.

7.2.3.10 Authentication frame format

Change the first paragraph of 7.2.3.10 as shown:

The frame body of a management frame of subtype Authentication contains the information shown in Table 7-16. Only Authentication frames with the authentication algorithm set to Open System authentication or FT authentication may be used within an RSNA. RSNA STAs shall not associate if shared authentication was invoked prior to RSN association. FT authentication is used when FT support is advertised by the AP and dot11FastBSSTransitionEnabled is set to TRUE in the non-AP STA.

Insert order 5 through 9 information fields into Table 7-16:

Table 7-16—Authentication frame body

Order	Information	Notes
5	RSN	The RSNIE is present in the FT Authentication frames as defined in Table 7-17.
6	Mobility domain	The MDIE is present in the FT Authentication frames as defined in Table 7-17.
7	Fast BSS transition	An FTIE is present in the FT Authentication frames as defined in Table 7-17.
8	Timeout interval (reassociation deadline)	A Timeout Interval information element (TIE) containing the reassociation deadline interval is present in the FT Authentication frames as defined in Table 7-17.
9	RIC	A Resource Information Container, containing a variable number of information elements, is present in the FT Authentication frames as defined in Table 7-17.

Change Table 7-17 as follows:

Table 7-17—Presence of ~~challenge text~~ information elements in Authentication frames

Authentication algorithm	Authentication transaction sequence no.	Status code	~~Challenge Text~~ Presence of fields 4–9
Open System	1	Reserved	Not present
Open System	2	Status	Not present
Shared Key	1	Reserved	Not present
Shared Key	2	Status	Challenge text p~~P~~resent
Shared Key	3	Reserved	Challenge text p~~P~~resent
Shared Key	4	Status	Not present
FT	1	Reserved	Mobility domain is present. If dot11RSNAEnabled is set to TRUE fast BSS transition and RSN are present.
FT	2	Status	If Status is zero, Mobility domain is present. If Status is zero and dot11RSNAEnabled is set to TRUE fast BSS transition and RSN are present.
FT	3	Reserved	Mobility domain is present. If dot11RSNAEnabled is set to TRUE fast BSS transition and RSN are present. RIC may be present.
FT	4	Status	If Status is zero, mobility domain is present. If Status is zero and dot11RSNAEnabled is set to TRUE fast BSS transition and RSN are present. If Status is zero, RIC is optionally present. If a RIC is present in a non-RSN, timeout interval (reassociation deadline) is present.

7.3 Management frame body components

7.3.1 Fields that are not information elements

7.3.1.1 Authentication Algorithm Number field

Insert the following line after "Authentication algorithm number = 1":

Authentication algorithm number = 2: Fast BSS Transition

7.3.1.9 Status Code field

Insert status codes 28 and 52 through 55 and change the Reserved status code rows in Table 7-23 as follows (note that the entire table is not shown here):

Table 7-23—Status codes

Status Code	Meaning
27~~-31~~	Reserved
28	R0KH unreachable
29-31	Reserved
52	Invalid FT Action frame count
53	Invalid pairwise master key identifier (PMKID)
54	Invalid MDIE
55	Invalid FTIE
~~52~~56-65535	Reserved

7.3.1.11 Action field

Insert category code 6 and change the Reserved category code row in Table 7-24 as follows (note that the entire table is not shown here):

Table 7-24—Category values

Code	Meaning	See subclause
6	Fast BSS Transition	7.4.8
~~6~~7-126	Reserved	—

7.3.2 Information elements

Insert element identifiers (IDs) 54 through 57 and element ID 75 and change the Reserved element ID rows in Table 7-26 as follows (note that the entire table is not shown here):

Table 7-26—Element IDs

Information Element	Element ID	Length (in octets)	Extensible
Mobility Domain (MDIE) (see 7.3.2.47)	54	5	
Fast BSS Transition (FTIE) (see 7.3.2.48)	55	84 to 257	
Timeout Interval (see 7.3.2.49)	56	7	

Table 7-26—Element IDs *(continued)*

Information Element	Element ID	Length (in octets)	Extensible
RIC Data (RDIE) (see 7.3.2.50)	57	6	
Reserved	~~54~~58–62		
Reserved	72–74~~126~~		
RIC Descriptor (see 7.3.2.51)	75	3–257	
Reserved	76–126		

7.3.2.25 RSN information element (RSNIE)

7.3.2.25.2 AKM suites

Insert suite types 3 and 4 and change the Reserved suite type row in Table 7-34 as shown (note that the entire table is not shown here):

Table 7-34—AKM suite selectors

OUI	Suite type	Meaning	
		Authentication type	Key management type
00-0F-AC	3	FT authentication negotiated over IEEE 802.1X	FT key management as defined in 8.5.1.5
00-0F-AC	4	FT authentication using PSK	FT key management as defined in 8.5.1.5
00-0F-AC	~~3~~5–255	Reserved	Reserved

7.3.2.25.4 PMKID

Change first two paragraphs of 7.3.2.25.4 as follows:

The PMKID Count and List fields ~~shall be~~are used only in the RSNIE in the (Re)Association Request frame to an AP and in FT authentication sequence frames. The PMKID Count specifies the number of PMKIDs in the PMKID List field. The PMKID list contains 0 or more PMKIDs that the STA believes to be valid for the destination AP. The PMKID can refer to

a) A cached pairwise master key security association (PMKSA) that has been obtained through preauthentication with the target AP

b) A cached PMKSA from an EAP authentication

c) A PMKSA derived from a PSK for the target AP

d) A PMK-R0 security association derived as part of an FT initial mobility domain association

e) A PMK-R1 security association derived as part of an FT initial mobility domain association or as part of a fast BSS transition.

See 8.5.1.2 for the construction of the PMKID, 11A.8 for the population of PMKID for fast BSS transitions, and 8.5.1.5 for the construction of PMKR0Name and PMKR1Name.

7.3.2.37 Neighbor Report element

Replace Figure 7-95c with the following:

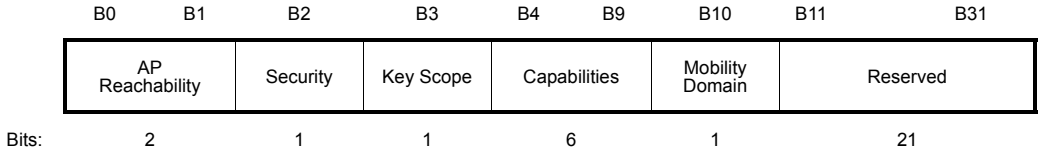

Figure 7-95c—BSSID Information field

Insert the following paragraph after the ninth paragraph in 7.3.2.37 as follows:

The Mobility Domain bit, when set to 1, indicates that the AP represented by this basic service set identification (BSSID) is including an MDIE in its Beacon frames and that the contents of that MDIE are identical to the MDIE advertised by the AP sending the report.

Change the following paragraph in 7.3.2.37 as follows:

Bits ~~10~~11–31 are reserved.

Insert the following subclauses (7.3.2.47 through 7.3.2.51) after 7.3.2.46:

7.3.2.47 Mobility domain information element (MDIE)

The MDIE contains the Mobility Domain Identifier (MDID) and the FT Capability and Policy Field. The AP uses the MDIE to advertise that it is included in the group of APs that constitute a mobility domain, to advertise its support for FT capability, and to advertise its FT policy information. The format for this information element is given in Figure 7-95o1.

Figure 7-95o1—MDIE format

The Length field is set to 3.

The MDID field is a 2-octet value that follows the ordering conventions defined in 7.1.1.

The FT Capability and Policy field is one octet. The FT Capability and Policy field is defined in Figure 7-95o2.

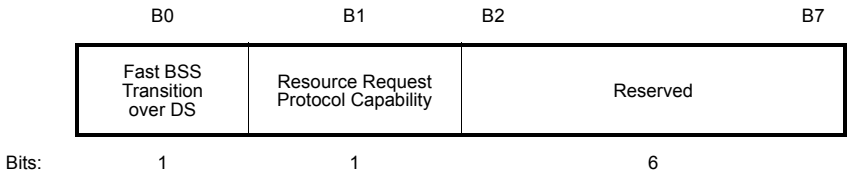

Figure 7-95o2—FT Capability and Policy field

Bits 0–1 of the FT Capability and Policy field control the behavior of STAs performing fast BSS transitions (see 11A.3). The STA can use information from the MDIE to determine the transition methods recommended by the AP and protocols supported by the AP. The choice of executing any specific transition method is outside the scope of this standard.

If Resource Request Protocol Capability subfield is set to 1, then the STA may perform the FT Resource Request Protocol of 11A.6.

When sent by a non-AP STA to a target AP, the FT Capability and Policy field matches the value advertised by that target AP. See 11A.8.

7.3.2.48 Fast BSS transition information element (FTIE)

The FTIE includes information needed to perform the FT authentication sequence during a fast BSS transition in an RSN. This information element is shown in Figure 7-95o3.

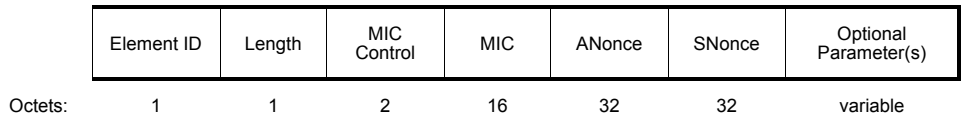

Figure 7-95o3—FTIE format

The Length field for this element indicates the length of the information field, as defined below.

The MIC Control field is two octets and is defined in Figure 7-95o4.

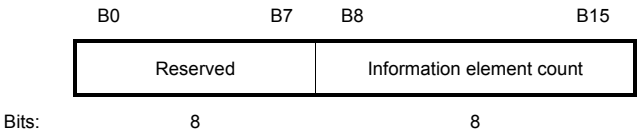

Figure 7-95o4—MIC Control field

The Information Element Count subfield of the MIC Control field contains the number of information elements that are included in the message integrity code (MIC) calculation. A value of zero indicates no MIC is present.

The MIC field contains a MIC that is calculated using the algorithm specified in 11A.8.4 and 11A.8.5.

The ANonce field contains a value chosen by the R1KH. It is encoded following the conventions in 7.1.1.

The SNonce field contains a value chosen by the S1KH. It is encoded following the conventions in 7.1.1.

The format of the Optional Parameter(s) field is shown in Figure 7-95o5.

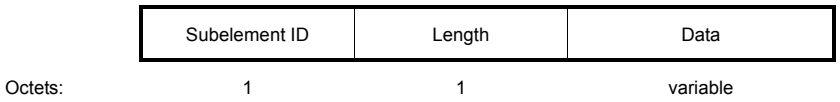

Figure 7-95o5—Optional Parameter(s) field

The Subelement ID is one of the values from Table 7-43g:

Table 7-43g—Subelement IDs

Value	Contents of Data field	Length (in octets)
0	Reserved	
1	PMK-R1 key holder identifier (R1KH-ID)	6
2	GTK	15–42
3	PMK-R0 key holder identifier (R0KH-ID)	1–48
4–255	Reserved	

R1KH-ID indicates the identity of the R1KH, which is used by the S0KH and the R0KH for deriving the PMK-R1s. It is encoded following the conventions in 7.1.1.

The GTK subelement contains the group temporal key, which is encrypted (see procedures in 11A.8.5) and is defined in Figure 7-95o6.

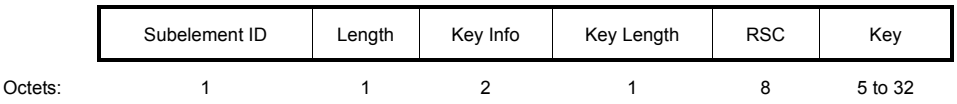

Figure 7-95o6—GTK subelement format

The GTK subelement Key Info subfield is defined in Figure 7-95o7.

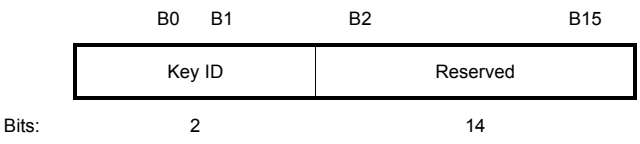

Figure 7-95o7—GTK subelement's Key Info subfield

Key Length field is the length of the Key field in octets. This length value does not include the possible padding (see 11A.8.5).

RSC field contains the receive sequence counter (RSC) for the GTK being installed. The RSC field gives the current message number for the GTK to allow a STA to identify replayed MPDUs. If the RSC field value is less than 8 octets in length, the remaining octets are set to 0. The least significant octet of the transmit sequence counter (TSC) or packet number (PN) is in the first octet of the RSC field.

NOTE—The RSC field value for TKIP is the TSC and is stored in the first 6 octets; for CCMP, it is the PN and is stored in the first 6 octets. See Table 61.[1]

For WEP, the RSC value is set to 0 on transmit and is not used at the receiver.

When sent by a non-AP STA, the R0KH-ID indicates the R0KH with which the S0KH negotiated the PMK-R0 it is using for this transition. When sent by an AP, the R0KH-ID indicates the R0KH that the S0KH will be using to generate a PMK-R0 security association. It is encoded following the conventions from 7.1.1.

7.3.2.49 Timeout Interval information element (TIE)

The TIE specifies time intervals and timeouts. Figure 7-95o8 shows this information element.

	Element ID	Length	Timeout Interval Type	Timeout Interval Value
Octets:	1	1	1	4

Figure 7-95o8—TIE format

The Length field is set to 5.

The Timeout Interval Type field contains one of the values from Table 7-43h.

Table 7-43h—Timeout Interval Type field value

Timeout interval type	Meaning	Units
0	Reserved	
1	Reassociation deadline interval	Time units (TUs)
2	Key lifetime interval	Seconds
3–255	Reserved	

The Timeout Interval Value field contains an unsigned 32-bit integer. It is encoded according to the conventions in 7.1.1.

A reassociation deadline interval value of zero indicates no deadline exists. A key lifetime interval value of zero is reserved.

[1]Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement the standard.

7.3.2.50 RIC Data information element (RDIE)

The RIC refers to a collection of information elements that are used to express a resource request and to convey responses to the corresponding requests.

A RIC is a sequence of one or more Resource Requests, or a sequence of one or more Resource Responses. Each Resource Request or Response consists of a RDIE, followed by one or more information elements that describe that resource. See 11A.11 for examples and procedures.

The RDIE format is shown in Figure 7-95o9.

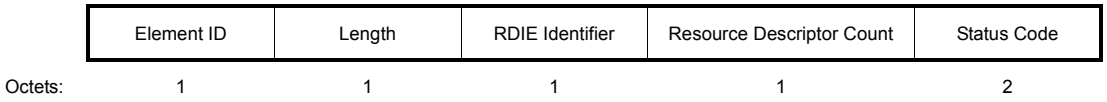

Figure 7-95o9—RDIE format

The Length field is set to 4.

The RDIE Identifier field has an arbitrary 8-bit value, chosen by the resource requestor to uniquely identify the RDIE within the RIC.

The Resource Descriptor Count field indicates the number of alternative Resource Descriptors that follow this RDIE.

The Status Code field is used in Resource Responses to indicate the result of the request. Valid values for the Status Code field are given in 7.3.1.9. When an RDIE is included in a Resource Request, the Status Code field is set to 0 and ignored upon receipt.

7.3.2.51 RIC Descriptor information element

The RIC Descriptor information element is used with an RDIE during a fast BSS transition to negotiate resources that are not otherwise described by information elements. See 11A.11 for procedures for including this information element in a RIC.

Figure 7-95o10 shows the format of this information element.

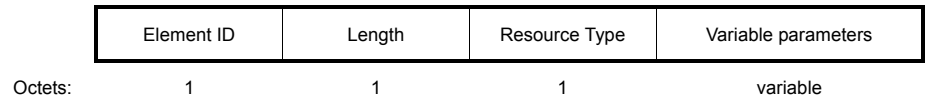

Figure 7-95o10—RIC Descriptor information element format

The Length field is set to the number of octets in this information element (variable).

The Resource Type field contains one of the values given in Table 7-43i.

Variable parameters contain any additional data based on the resource type.

Table 7-43i—Resource type code in RIC Descriptor information element

Resource type value	Meaning	Variable parameters
1	Block Ack	Block Ack parameter set as defined in 7.3.1.14, Block Ack timeout value as defined in 7.3.1.15, and Block Ack starting sequence control as defined in 7.2.1.7.
0, 2–255	Reserved	

7.4 Action frame format details

Insert the following subclauses (7.4.8 through 7.4.8.4) after 7.4.7.2.

7.4.8 FT Action frame details

Four Action frame formats are defined to support fast BSS transitions over the DS, which are initiated through the currently associated AP. The FT Action frames are sent over the air between the STA and the current AP. The Action frame is used as a transport mechanism for data that are destined for the target AP. The Action field values associated with each FT Action frame format are defined in Table 7-57g.

Table 7-57g—Action field values in FT Action frames

Action field value	Description
0	Reserved
1	FT Request frames
2	FT Response frames
3	FT Confirm frames
4	FT Ack frames
5–255	Reserved

7.4.8.1 FT Request frame

The FT Request frame is sent by the STA to its associated AP to initiate an over-the-DS fast BSS transition.

Figure 7-101i shows the format of the FT Request frame.

	Category	Action	STA Address	Target AP Address	FT Request frame body
Octets:	1	1	6	6	variable

Figure 7-101i—FT Request frame format

The Category field is set to the value given in 7.3.1.11 for FT Action frames.

The Action field is set to the value given in Table 7-57g for FT Request frames.

The STA Address field is set to the STA's MAC address.

The Target AP Address field is set to the BSSID value of the target AP.

The FT Request frame body contains the information shown in Table 7-57h.

Table 7-57h—FT Request frame body

Order	Information	Notes
1	RSN	A RSNIE is present if dot11RSNAEnabled is set to TRUE.
2	Mobility domain	The MDIE is present.
3	Fast BSS transition	An FTIE is present if dot11RSNAEnabled is set to TRUE.

The usage of these information elements is defined in 11A.8.2.

7.4.8.2 FT Response frame

The FT Response frame is transmitted by the currently associated AP as a response to the STA's FT Request frame. Figure 7-101j shows the format of the FT Response frame.

	Category	Action	STA Address	Target AP Address	Status Code	FT Response frame body
Octets:	1	1	6	6	2	variable

Figure 7-101j—FT Response frame format

The Category field is set to the value given in 7.3.1.11 for FT Action frames.

The Action field is set to the value given in Table 7-57g for FT Response frames.

The STA Address field is set to the STA's MAC address.

The Target AP Address field is set to the BSSID value of the target AP.

The Status Code field is a value from the options listed in 7.3.1.9.

If the Status Code field is zero, then the FT Response frame body contains the information shown in Table 7-57i.

The usage of these information elements is defined in 11A.8.3.

Table 7-57i—FT Response frame body

Order	Information	Notes
1	RSN	The RSNIE is present if dot11RSNAEnabled is set to TRUE.
2	Mobility domain	The MDIE is present.
3	Fast BSS transition	An FTIE is present if dot11RSNAEnabled is set to TRUE.

7.4.8.3 FT Confirm frame

The FT Confirm frame is an RSN confirms to the target AP receipt of the ANonce and indicate the liveness of the PTKSA. The FT Confirm frame is optionally used by the STA to request resources. Figure 7-101k shows the FT Confirm frame.

	Category	Action	STA Address	Target AP Address	FT Confirm frame body
Octets:	1	1	6	6	variable

Figure 7-101k—FT Confirm frame format

The Category field is set to the value given in 7.3.1.11 for FT Action frames.

The Action field is set to the value given in Table 7-57g for FT Confirm frames.

The STA Address field is set to the STA's MAC address.

The Target AP Address field is set to the BSSID value of the target AP.

The FT Confirm frame body contains the information shown in Table 7-57j.

Table 7-57j—FT Confirm frame body

Order	Information	Notes
1	RSN	The RSNIE is present if dot11RSNAEnabled is set to TRUE.
2	Mobility domain	The MDIE is present.
3	Fast BSS transition	An FTIE is present if dot11RSNAEnabled is set to TRUE.
4	RIC	The RIC Request field is present if resources are being requested.

The usage of these information elements is defined in 11A.8.4.

7.4.8.4 FT Ack frame

The FT Ack frame is transmitted by the currently associated AP as a response to the STA's FT Confirm frame. Figure 7-101l shows the FT Ack frame.

	Category	Action	STA Address	Target AP Address	Status Code	FT Ack frame body
Octets	1	1	6	6	2	variable

Figure 7-101l—FT Ack frame format

The Category field is set to the value given in 7.3.1.11 for FT Action frames.

The Action field is set to the value given in Table 7-57g for FT Ack frames.

The STA Address field is set to the STA's MAC address.

The Target AP Address field is set to the BSSID value of the target AP.

The Status Code field is a value from the options listed in 7.3.1.9.

If the Status Code field is zero, then the FT Ack frame body contains the information shown in Table 7-57k.

Table 7-57k—FT Ack frame body

Order	Information	Notes
1	RSN	The RSNIE is present if dot11RSNAEnabled is set to TRUE.
2	Mobility domain	The MDIE is present.
3	Fast BSS transition	An FTIE is present if dot11RSNAEnabled is set to TRUE.
4	Timeout interval (reassociation deadline)	A TIE containing the reassociation deadline interval is present if resources were requested in the FT Confirm frame and dot11RSNAEnabled is set to FALSE.
5	RIC	The RIC Response field is present if resources were requested in the FT Confirm frame.

The usage of these information elements is defined in 11A.8.5.

8. Security

8.4 RSNA Security association management

8.4.1 Security associations

8.4.1.1 Security association definitions

Change the dashed list at the end of 8.4.1.1 as follows:

— PMKSA: A result of a successful IEEE 802.1X exchange, preshared PMK information, or PMK cached via some other mechanism.

— PMK-R0 security association: A result of a successful FT initial mobility domain association.

- PMK-R1 security association: A result of a successful FT initial mobility domain association or FT authentication sequence.
- PTKSA: A result of a successful 4-Way Handshake, FT 4-Way Handshake, or FT authentication sequence.
- GTKSA: A result of a successful Group Key Handshake, ~~or successful~~ 4-Way Handshake, FT 4-Way Handshake, or FT authentication sequence.
- SMKSA: A result of a successful initial SMK Handshake.
- STKSA: A result of a successful 4-way STK Handshake following the initial SMK Handshake or subsequent rekeying.

Insert the following subclauses (8.4.1.1.1a and 8.4.1.1.1b) after 8.4.1.1.1:

8.4.1.1.1a PMK-R0 security association

The PMK-R0 security association is the result of a successful completion of the IEEE 802.1X authentication or use of PSK during the FT initial mobility domain association. This security association is bidirectional. It consists of the following elements:

- SSID
- MDID
- PMK-R0
- R0KH-ID
- PMKR0Name
- S0KH-ID
- PMK-R0 lifetime
- Pairwise cipher suite selector
- All authorization parameters specified by the AS or local configuration

8.4.1.1.1b PMK-R1 security association

The PMK-R1 security association is the result of

- A successful completion of the IEEE 802.1X authentication or use of PSK during the FT initial mobility domain association or
- A successful completion of the authentication phase in the fast BSS transition to the target AP

This security association is bidirectional. It consists of the following elements:

- SSID
- MDID
- PMK-R1
- PMK-R1 lifetime
- PMKR1Name
- R1KH-ID
- R0KH-ID
- PMKR0Name
- S0KH-ID
- S1KH-ID
- Pairwise cipher suite selector
- All authorization parameters specified by the AS or local configuration

8.4.1.1.2 PTKSA

Change 8.4.1.1.2 as follows:

The PTKSA is a result of the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, or FT Resource Request Protocol. This security association is also bidirectional. ~~The PTKSA is used to create the key hierarchy.~~ PTKSAs are cached for the life of the PMKSA or PMK-R1 security association. Because the PTKSA is tied to the PMKSA or to a PMK-R1 security association, it only has the additional information from the 4-Way Handshake. For the PTKSA derived as a result of the 4-Way Handshake, ~~T~~there shall be only one PTKSA with the same Supplicant and Authenticator MAC addresses. For the PTKSA derived as a result of an initial mobility domain association or fast BSS transition, there shall be only one PTKSA with the same non-AP STA MAC address and BSSID.

During the 4-Way Handshake defined in 8.5.3.4 and the FT 4-Way Handshake defined in 11A.4.2, t~~T~~here is state created between Message 1 and Message 3 of a 4-Way Handshake. This does not create a PTKSA until Message 3 is validated ~~on~~by the Supplicant and Message 4 is validated by the Authenticator.

During the FT authentication sequence defined in 11A.8, the PTKSA is validated when Message 3 is validated by the R1KH and Message 4 is validated by the S1KH.

The PTKSA consists of the following elements:

- PTK
- Pairwise cipher suite selector
- Supplicant MAC address or non-AP STA's MAC address
- Authenticator MAC address or BSSID
- If FT key hierarchy is used,
  - R1KH-ID
  - S1KH-ID
  - PTKName

8.4.6 RSNA authentication in an ESS

8.4.6.1 Preauthentication and RSNA key management

Change first paragraph of 8.4.6.1 as follows:

A STA shall not use preauthentication except when pairwise keys are employed. A STA shall not use preauthentication within the same mobility domain if AKM suite type 00-0F-AC:3 or 00-0F-AC:4 is used in the current association. Preauthentication shall not be used unless the new AP advertises the preauthentication capability in the RSNIE.

8.4.10 RSNA security association termination

Change the first two sentences of the first paragraph of 8.4.10 as follows:

When a non-AP STA station management entity (SME) receives a successful MAC sublayer management entity (MLME) Association or Reassociation confirm primitive that is not part of a fast BSS transition or receives or invokes an MLME Disassociation or Deauthentication primitive, it will delete some security associations. Similarly, when an AP SME receives an MLME Association or Reassociation indication primitive that is not part of a fast BSS transition or receives or invokes an MLME Disassociation or Deauthentication primitive, it will delete some security associations.

8.5 Keys and key distribution

8.5.1 Key hierarchy

Insert the following subclauses (8.5.1.5 through 8.5.1.5.5) after 8.5.1.4:

8.5.1.5 FT key hierarchy

8.5.1.5.1 Overview

This subclause describes the FT key hierarchy and its supporting architecture. The FT key hierarchy is designed to allow a STA to make fast BSS transitions between APs without the need to perform an IEEE 802.1X authentication at every AP within the mobility domain.

The FT key hierarchy can be used with either IEEE 802.1X authentication or PSK authentication.

A three-level key hierarchy provides key separation between the key holders. The FT key hierarchy for the Authenticator is shown in Figure 8-22a. An identical key hierarchy exists for the Supplicant, and identical functions are performed by the corresponding S0KH and S1KH.

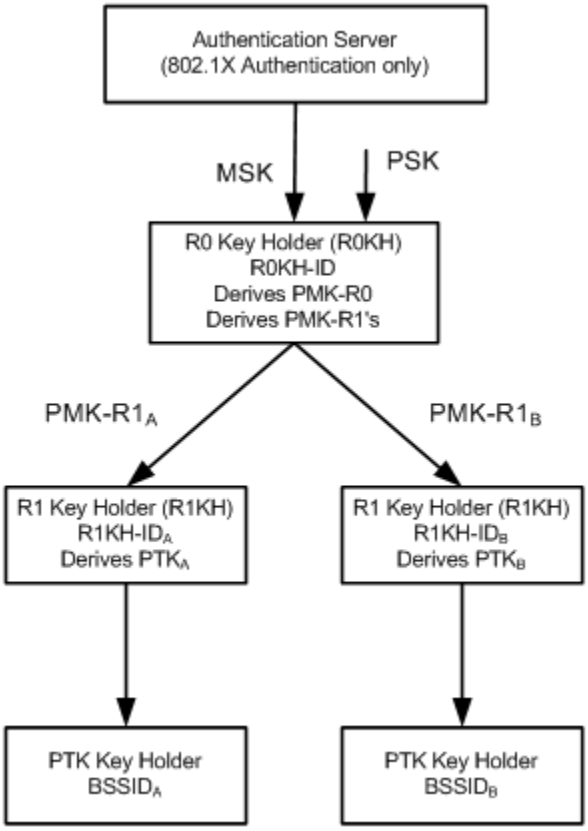

Figure 8-22a—FT key hierarchy at an Authenticator

The FT key hierarchy shown in Figure 8-22a consists of three levels whose keys are derived using the key derivation function (KDF) described in 8.5.1.5.2.

a) PMK-R0 – the first-level key of the FT key hierarchy. This key is derived as a function of the master session key (MSK) or PSK. It is stored by the PMK-R0 key holders, R0KH and S0KH.

b) PMK-R1 – the second-level key of the FT key hierarchy, This key is mutually derived by the S0KH and R0KH.

c) PTK – the third-level key of the FT key hierarchy that defines the IEEE 802.11 and IEEE 802.1X protection keys. The PTK is mutually derived by the PMK-R1 key holders, R1KH and S1KH.

As shown in Figure 8-22a, the R0KH computes the PMK-R0 either from the PSK or from the MSK resulting (per IETF RFC 3748-2004 [B26][2]) from a successful IEEE 802.1X authentication between the AS and the Supplicant. Upon a successful authentication, the R0KH shall delete any prior PMK-R0 security association for this mobility domain pertaining to this S0KH. The R0KH shall also delete all PMK-R1 security associations derived from that prior PMK-R0 security association. The PMK-R1s are generated by the R0KH and are assumed to be delivered from the R0KH to the R1KHs within the same mobility domain. The PMK-R1s are used for PTK generation. Upon receiving a new PMK-R1 for an S0KH, an R1KH deletes the prior PMK-R1 security association and PTKSAs derived from the prior PMK-R1.

It is assumed by this standard that the PSK is specific to a single S0KH and a single R0KH.

The lifetime of the PMK-R0, PMK-R1, and PTK are bound to the lifetime of the PSK or MSK. For example, the AS may communicate the MSK lifetime with the MSK. If such an attribute is provided, the lifetime of the PMK-R0 shall be not more than the lifetime of the MSK. The lifetime of the PTK and PMK-R1 is the same as that of the PMK-R0. When the key lifetime expires, each key holder shall delete its respective PMK-R0, PMK-R1 or PTK security association.

The FT key hierarchy derives its keys using the KDF defined in 8.5.1.5.2 with separate labels to further distinguish derivations.

During a fast BSS transition, a non-AP STA shall negotiate the same pairwise cipher suite with target APs as was negotiated in the FT initial mobility domain association. Using the pairwise cipher suite selector value in the PMK-R1 security association received from the R0KH, the target AP shall verify that the same pairwise cipher suite selector is being used.

The distribution of keys from the R0KH to the R1KHs is outside the scope of this standard. It is assumed that the PMK-R1s are distributed from the R0KH to the R1KHs following the requirements specified in 11A.2.2.

The PMK-R0 may be deleted by the R0KH after PMK-R1s have been derived. When the PMK-R0 is deleted, the R0KH needs only to maintain the PMK-R1 security associations.

8.5.1.5.2 Key derivation function (KDF)

The KDF for the FT key hierarchy is a variant of the pseudo-random function (PRF) defined in 8.5.1.1 and is defined as follows:

Output ← KDF-Length (K, label, Context) where
Input: *K*, a 256-bit key derivation key
label, a string identifying the purpose of the keys derived using this KDF
Context, a bit string that provides context to identify the derived key
Length, the length of the derived key in bits
Output: a *Length*-bit derived key

[2]The numbers in brackets correspond to those of the bibliography in Annex P.

result ← ""
iterations ← (*Length*+255)/256
do *i* = 1 **to** *iterations*
    result ← *result* || HMAC-SHA256(*K*, *i* || *label* || *Context* || *Length*)
od
return first *Length* bits of *result,* and securely delete all unused bits

In this algorithm, *i* and *Length* are encoded as 16-bit unsigned integers, represented using the bit ordering conventions of 7.1.1. *K*, *label*, and *Context* are bit strings and are represented using the ordering conventions of 7.1.1.

8.5.1.5.3 PMK-R0

The first-level FT key hierarchy key, PMK-R0, is derived using the KDF defined in 8.5.1.5.2. The PMK-R0 is the first level 256-bit keying material used to derive the next level keys (PMK-R1s):

R0-Key-Data = KDF-384(XXKey, "FT-R0", SSIDlength || SSID || MDID || R0KHlength || R0KH-ID || S0KH-ID)

PMK-R0 = L(R0-Key-Data, 0, 256)

PMK-R0Name-Salt = L(R0-Key-Data, 256, 128)

where

— KDF-384 is the KDF as defined in 8.5.1.5.2 used to generate a key of length 384 bits.
— L(-) is defined in 8.5.1.
— If the AKM negotiated is 00-0F-AC:3, then XXKey shall be the second 256 bits of the MSK (which is derived from the IEEE 802.1X authentication), i.e., XXKey = L(MSK, 256, 256). If the AKM negotiated is 00-0F-AC:4, then XXKey shall be the PSK.
— "FT-R0" is 0x46 0x54 0x2D 0x52 0x30.
— SSIDlength is a single octet whose value is the number of octets in the SSID.
— SSID is the service set identifier, a variable length sequence of octets, as it appears in the Beacon and Probe Response frames.
— MDID is the Mobility Domain Identifier field from the MDIE that was used during FT initial mobility domain association.
— R0KHlength is a single octet whose value is the number of octets in the R0KH-ID.
— R0KH-ID is the identifier of the holder of PMK-R0 in the Authenticator.
— S0KH-ID is the Supplicant's MAC address (SPA).

PMK-R0 shall be computed as the first 256 bits (bits 0-255) of the R0-Key-Data. The latter 128 bits of R0-Key-Data shall be used as the PMK-R0Name-Salt to generate the PMKR0Name.

The PMK-R0 is referenced and named as follows:

PMKR0Name = Truncate-128(SHA-256("FT-R0N" || PMK-R0Name-Salt))

where

— "FT-R0N" is 0x46 0x54 0x2D 0x52 0x30 0x4E.
— Truncate-128(-) returns the first 128 bits of its argument and securely destroys the remainder.

The PMKR0Name is used to identify the PMK-R0.

8.5.1.5.4 PMK-R1

The second-level key in the FT key hierarchy, PMK-R1, is a 256-bit key used to derive the PTK. The PMK-R1 is derived using the KDF defined in 8.5.1.5.2:

PMK-R1 = KDF-256(PMK-R0, "FT-R1", R1KH-ID || S1KH-ID)

where

- KDF-256 is the KDF as defined in 8.5.1.5.2 used to generate a key of length 256 bits.
- PMK-R0 is the first level key in the FT key hierarchy.
- "FT-R1" is 0x46 0x54 0x2D 0x52 0x31.
- R1KH-ID is a MAC address of the holder of the PMK-R1 in the Authenticator of the AP.
- S1KH-ID is the SPA.

The PMK-R1 is referenced and named as follows:

PMKR1Name = Truncate-128(SHA-256(“FT-R1N” || PMKR0Name || R1KH-ID || S1KH-ID))

where

- "FT-R1N" is 0x46 0x54 0x2D 0x52 0x31 0x4E.

PMKR1Name is used to identify the PMK-R1.

8.5.1.5.5 PTK

The third level of the key hierarchy is the PTK. This key is mutually derived by the S1KH and the R1KH used by the target AP, with the key length being a function of the negotiated cipher suite as defined by Table 8-2 in 8.5.2.

Using the KDF defined in 8.5.1.5.2, the PTK derivation is as follows:

PTK = KDF-PTKLen(PMK-R1, "FT-PTK", SNonce || ANonce || BSSID || STA-ADDR)

where

- KDF-PTKLen is the KDF as defined in 8.5.1.5.2 used to generate a PTK of length PTKLen.
- PMK-R1 is the key that is shared between the S1KH and the R1KH.
- "FT-PTK" is 0x46 0x54 0x2D 0x50 0x54 0x4B.
- SNonce is a 256-bit random bit string contributed by the S1KH.
- ANonce is a 256-bit random bit string contributed by the R1KH.
- STA-ADDR is the non-AP STA’s MAC address.
- BSSID is the BSSID of the target AP.
- PTKlen is the total number of bits to derive, i.e., number of bits of the PTK. The length is dependent on the negotiated cipher suites as defined by Table 8-2 in 8.5.2.

Each PTK has three component keys, KCK, KEK, and a temporal key, derived as follows:

The KCK shall be computed as the first 128 bits (bits 0–127) of the PTK:

KCK = L(PTK, 0, 128)

where L(-) is defined in 8.5.1.

The KCK is used to provide data origin authenticity in EAPOL-Key messages, as defined in 8.5.2, and in the FT authentication sequence, as defined in 11A.8.

The KEK shall be computed as bits 128–255 of the PTK:

KEK = L(PTK, 128, 128)

The KEK is used to provide data confidentiality for certain fields (KeyData) in EAPOL-Key messages, as defined in 8.5.2, and in the FT authentication sequence, as defined in 11A.8.

The temporal key (TK) shall be computed as bits 256–383 (for CCMP) of the PTK:

TK = L(PTK, 256, 128)

For vendor-specific cipher suites, the length of the temporal key (and the value of PTKLen) depend on the vendor-specific algorithm.

The temporal key is configured into the STA by the SME through the use of the MLME-SETKEYS.request primitive. The STA uses the temporal key with the pairwise cipher suite; interpretation of this value is specific to the cipher suite.

The PTK is referenced and named as follows:

PTKName = Truncate-128(SHA-256(PMKR1Name || “FT-PTKN” || SNonce || ANonce || BSSID || STA-ADDR))

where

— "FT-PTKN" is 0x46 0x54 0x2D 0x50 0x54 0x4B 0x4E.

The PTKName is used to identify the PTK key.

8.5.2 EAPOL-Key frames

Change list item 1) of list item b) Key Information in 8.5.2 as shown:

1) Key Descriptor Version (bits 0–2) specifies the key descriptor version type.
   i) The value 1 shall be used for all EAPOL-Key frames to and from a STA when the negotiated AKM is 00-0F-AC:1 or 00-0F-AC:2 and neither the group nor pairwise ciphers are CCMP for Key Descriptor 1. This value indicates the following:
      — HMAC-MD5 is the EAPOL-Key MIC.
      — ARC4 is the EAPOL-Key encryption algorithm used to protect the Key Data field.
   ii) The value 2 shall be used for all EAPOL-Key frames to and from a STA when the negotiated AKM is 00-0F-AC:1 or 00-0F-AC:2 and either the pairwise or the group cipher is AES-CCMP for Key Descriptor 2. This value indicates the following:

— HMAC-SHA1-128 is the EAPOL-Key MIC. HMAC is defined in IETF RFC 2104; and SHA1, by FIPS PUB 180-1-1995. The output of the HMAC-SHA1 shall be truncated to its 128 MSBs (octets 0-15 of the digest output by HMAC-SHA1), i.e., the last four octets generated shall be discarded.

— The NIST AES key wrap is the EAPOL-Key encryption algorithm used to protect the Key Data field. IETF RFC 3394 defines the NIST AES key wrap algorithm.

iii) The value 3 shall be used for all EAPOL-Key frames to and from a STA when the negotiated AKM is 00-0F-AC:3 or 00-0F-AC:4. This value indicates the following:

— AES-128-CMAC is the EAPOL-Key MIC. AES-128-CMAC is defined by FIPS SP800-38B.[3] The output of the AES-128-CMAC shall be 128 bits.

— The NIST AES key wrap is the EAPOL-Key encryption algorithm used to protect the Key Data field. IETF RFC 3394 defines the NIST AES key wrap algorithm.

Change list item h) Key MIC in 8.5.2 as shown:

h) **Key MIC.** This field is 16 octets in length when the Key Descriptor Version subfield is 1, 2 or ~~3~~3. The EAPOL-Key MIC is a MIC of the EAPOL-Key frames, from and including the EAPOL protocol version field to and including the Key Data field, calculated with the Key MIC field set to 0. If the Encrypted Key Data subfield (of the Key Information field) is set, the Key Data field is encrypted prior to computing the MIC.

1) **Key Descriptor Version 1:** HMAC-MD5; IETF RFC 2104 and IETF RFC 1321 together define this function.

2) **Key Descriptor Version 2:** HMAC-SHA1-128.

3) **Key Descriptor Version 3:** AES-128-CMAC.

Change final paragraph of 8.5.2 as follows:

The key wrap algorithm selected depends on the key descriptor version.

— **Key Descriptor Version 1**: ARC4 is used to encrypt the Key Data field using the KEK field from the derived PTK. No padding shall be used. The encryption key is generated by concatenating the EAPOL-Key IV field and the KEK. The first 256 octets of the ARC4 key stream shall be discarded following ARC4 stream cipher initialization with the KEK, and encryption begins using the 257th key stream octet.

— **Key Descriptor Version 2 or 3**: AES key wrap, defined in IETF RFC 3394, shall be used to encrypt the Key Data field using the KEK field from the derived PTK. The key wrap default initial value shall be used.

NOTE—The cipher text output of the AES key wrap algorithm is 8 octets longer than the plaintext input.

8.5.2.1 EAPOL-Key frame notation

Change 8.5.2.1 as follows:

The following notation is used throughout the remainder of 8.5 and 11A.4 to represent EAPOL-Key frames:

EAPOL-Key(S, M, A, I, K, SM, KeyRSC, ANonce/SNonce, MIC, DataKDs)

(Note that the entire list is not shown here.)

DataKDs is a sequence of zero or more information elements and key data encapsulation (KDEs), contained in the Key Data field, which may contain the following:

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

RSNIE	is the RSN information element, described in 7.3.2.25.
RSNIE[KeyName]	is the RSN information element, with the PMKID field set to KeyName
GTK[N]	is the GTK, with Key ID field set to N. The key ID specifies which index should be used for this GTK. Index 0 shall not be used for GTKs, except in mixed environments, as described in 8.5.1.
FTIE	is the Fast BSS Transition information element, described in 7.3.2.48
MDIE	is the Mobility Domain information element, described in 7.3.2.47
TIE[IntervalType]	is a Timeout Interval information element of type IntervalType, as described in 7.3.2.49, containing e.g., for type KeyLifetime, the lifetime of the FT key hierarchy.

(The rest of the list remains unchanged.)

8.5.3 4-Way Handshake

8.5.3.1 4-Way Handshake Message 1

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.3.1 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

8.5.3.2 4-Way Handshake Message 2

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.3.2 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC) – same as Message 1.

8.5.3.3 4-Way Handshake Message 3

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.3.3 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC) – same as Message 1.

8.5.3.4 4-Way Handshake Message 4

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.3.4 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC) – same as Message 1.

8.5.4 Group Key Handshake

8.5.4.1 Group Key Handshake Message 1

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.4.1 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

8.5.4.2 Group Key Handshake Message 2

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.4.2 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC) – same as Message 1.

8.5.8 PeerKey Handshake

8.5.8.1 SMK Handshake

8.5.8.1.1 SMK Handshake Message 1

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.8.1.1 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

8.5.8.1.2 SMK Handshake Message 2

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.8.1.2 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

8.5.8.1.3 SMK Handshake Message 3

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.8.1.3 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

8.5.8.1.4 SMK Handshake Message 4

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.8.1.4 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

8.5.8.1.5 SMK Handshake Message 5

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.8.1.5 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

8.5.8.3 STKSA rekeying

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.8.3 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

8.5.8.4 Error reporting

Change the Key Descriptor Version subfield description under the Key Information field in 8.5.8.4 as follows:

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC)

10. Layer management

10.3 MLME SAP interface

10.3.4 Authenticate

10.3.4.1 MLME-AUTHENTICATE.request

10.3.4.1.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.4.1.2 as follows:

MLME-AUTHENTICATE.request (
PeerSTAAddress,
AuthenticationType,
AuthenticateFailureTimeout,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Change the AuthenticationType row of the untitled table defining the primitive parameters in 10.3.4.1.2 as shown:

Name	Type	Valid range	Description
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY, FAST_BSS_TRANSITION	Specifies the type of authentication algorithm to use during the authentication process.

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.4.1.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements to be included in the first message of the FT authentication sequence, as described in 11A.8.2. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.4.2 MLME-AUTHENTICATE.confirm

10.3.4.2.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.4.2.2 as follows:

MLME-AUTHENTICATE.confirm (
PeerSTAAddress,
AuthenticationType,
ResultCode,

Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Change the AuthenticationType row of the untitled table defining the primitive parameters in 10.3.4.2.2 as shown:

Name	Type	Valid range	Description
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY, FAST_BSS_TRANSITION	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 primitive

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.4.2.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements included in the second message of the FT authentication sequence, as described in 11A.8.3. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.4.3 MLME-AUTHENTICATE.indication

10.3.4.3.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.4.3.2 as follows:

MLME-AUTHENTICATE.indication (
PeerSTAAddress,
AuthenticationType,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Change the AuthenticationType row of the untitled table defining the primitive parameters in 10.3.4.3.2 as shown:

Name	Type	Valid range	Description
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY, FAST_BSS_TRANSITION	Specifies the type of authentication algorithm that was used during the authentication process.

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.4.3.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements included in the first message of the FT authentication sequence, as described in 11A.8.2. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.4.4 MLME-AUTHENTICATE.response

10.3.4.4.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.4.4.2 as follows:

MLME-AUTHENTICATE.response (
PeerSTAAddress,
ResultCode,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.4.4.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements to be included in the second message of the FT authentication sequence, as described in 11A.8.3. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.6 Associate

10.3.6.1 MLME-ASSOCIATE.request

10.3.6.1.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.6.1.2 as follows:

MLME-ASSOCIATE.request (
PeerSTAAddress,
AssociateFailureTimeout,
CapabilityInformation,
ListenInterval,
SupportedChannels,
RSN,
QoSCapability,

Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.6.1.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.4	The set of information elements to be included in the initial mobility domain association request, as described in 11A.4. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.6.2 MLME-ASSOCIATE.confirm

10.3.6.2.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.6.2.2 as follows:

MLME-ASSOCIATE.confirm (
ResultCode,
CapabilityInformation,
AssociationID,
SupportedRates,
EDCAParameterSet,
RCPI.request,
RSNI.request,
RCPI.response,
RSNI.response,
RRMEnabledCapabilities,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.6.2.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.4	The set of information elements included in the initial mobility domain association response, as described in 11A.4. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.6.3 MLME-ASSOCIATE.indication

10.3.6.3.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.6.3.2 as follows:

MLME-ASSOCIATE.indication (
PeerSTAAddress,
CapabilityInformation,
ListenInterval,
SSID
SupportedRates,
RSN,
QoSCapability,
RCPI,
RSNI,
RRMEnabledCapabilities,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.6.3.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.4	The set of information elements included in the initial mobility domain association, as described in 11A.4. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.6.4 MLME-ASSOCIATE.response

10.3.6.4.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.6.4.2 as follows:

MLME-ASSOCIATE.response (
PeerSTAAddress,
ResultCode,
CapabilityInformation,
AssociationID,
EDCAParameterSet,
RCPI,
RSNI,
RRMEnabledCapabilities,
Content of FT Authentication Information Elements,
VendorSpecificInfo)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.6.4.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.4	The set of information elements to be included in the initial mobility domain association response, as described in 11A.4. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.7 Reassociate

10.3.7.1 MLME-REASSOCIATE.request

10.3.7.1.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.7.1.2 as follows:

MLME-REASSOCIATE.request (
NewAPAddress,
ReassociateFailureTimeout,
CapabilityInformation,
ListenInterval,
SupportedChannels,
RSN,
QoSCapability,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.7.1.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements to be included in the third message of the FT authentication sequence, as described in 11A.8.4. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.7.2 MLME-REASSOCIATE.confirm

10.3.7.2.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.7.2.2 as follows:

MLME-REASSOCIATE.confirm (
ResultCode,
CapabilityInformation,
AssociationID,
SupportedRates,
EDCAParameterSet,

RCPI.request,
RSNI.request,
RCPI.response,
RSNI.response,
RRMEnabledCapabilities,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.4.1.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements included in the fourth message of the FT authentication sequence, as described in 11A.8.5. This includes an optional response to a resource request (RIC). Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.7.3 MLME-REASSOCIATE.indication

10.3.7.3.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.7.3.2 as follows:

MLME-REASSOCIATE.indication (
PeerSTAAddress,
CurrentAPAddress
CapabilityInformation,
ListenInterval,
SSID
SupportedRates,
RSN,
QoSCapability,
RCPI,
RSNI,
RRMEnabledCapabilities,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.7.3.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements included in the third message of the FT authentication sequence, as described in 11A.8.4. Present only when dot11FastBSSTransitionEnabled is set to TRUE.

10.3.7.4 MLME-REASSOCIATE.response

10.3.7.4.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.7.4.2 as follows:

MLME-REASSOCIATE.response (
PeerSTAAddress,
ResultCode,
CapabilityInformation,
AssociationID,
EDCAParameterSet,
RCPI,
RSNI,
RRMEnabledCapabilities,
Content of FT Authentication Information Elements,
VendorSpecificInfo
)

Insert the following row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.7.4.2:

Name	Type	Valid range	Description
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements to be included in the fourth message of the FT authentication sequence, as described in 11A.8.5. This includes an optional response to a resource request (RIC). Present only when dot11FastBSSTransitionEnabled is set to TRUE.

Insert the following subclauses (10.3.33 through 10.3.34.3.4) after 10.3.32.2.4.

10.3.33 MLME SAP interface for resource request

10.3.33.1 MLME-RESOURCE_REQUEST.request

10.3.33.1.1 Function

This primitive is used to perform the over-the-air resource request of an FT Resource Request Protocol. The over-the-air resource request is performed using Authentication frames, with an authentication algorithm of FT authentication and transaction sequence number of 3 or 4.

10.3.33.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-RESOURCE_REQUEST.request (
PeerMACAddress,
Contents of FT Authentication Information Elements
)

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the AP that is the intended immediate recipient of the resource request.
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements to be included in the FT Confirm frame, as described in 11A.8.4.

10.3.33.1.3 When generated

This primitive is generated by the SME at a non-AP STA to send the third frame of the over-the-air FT Resource Request Protocol. The third frame is an Authentication frame, with an authentication algorithm of FT authentication and transaction sequence value of 3.

10.3.33.1.4 Effect of receipt

Upon receipt of this primitive, the MLME constructs the appropriate Authentication frame and causes it to be transmitted to the peer MAC address.

10.3.33.2 MLME-RESOURCE_REQUEST.indication

10.3.33.2.1 Function

This primitive is used to enact the security and QoS resource request with a specified peer MAC entity.

10.3.33.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESOURCE_REQUEST.indication (
                                    PeerMACAddress,
                                    Content of FT Authentication Information Elements
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that was the sender of the resource request.
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements included in the FT Confirm frame, as described in 11A.8.4.

10.3.33.2.3 When generated

This primitive is generated by the MLME at an AP to indicate that the third frame of the over-the-air FT Resource Request Protocol has been received. The third frame is an Authentication frame, with an authentication algorithm of FT authentication and transaction sequence value of 3.

10.3.33.2.4 Effect of receipt

Upon receipt of this primitive, the SME examines the Transition information element and RSNIE contents and responds to the peer MAC address using the MLME-RESOURCE_REQUEST.response primitive.

10.3.33.3 MLME-RESOURCE_REQUEST.response

10.3.33.3.1 Function

This primitive is used to enact the security and QoS resource request protocol with a specified peer MAC entity.

10.3.33.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESOURCE_REQUEST.response (
                                    PeerMACAddress,
                                    Content of FT Authentication Information Elements
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the intended immediate recipient of the resource response.
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements to be included in the FT Ack frame, as described in 11A.8.5. This includes an optional response to a resource request (RIC).

10.3.33.3.3 When generated

This primitive is generated by the SME at an AP to cause the transmission of the fourth frame in the over-the-air FT Resource Request Protocol. The fourth frame is an Authentication frame, with an authentication algorithm of FT authentication and transaction sequence value of 4.

10.3.33.3.4 Effect of receipt

Upon receipt of this primitive, the MLME constructs the appropriate Authentication frame and causes it to be transmitted to the peer MAC address.

10.3.33.4 MLME-RESOURCE_REQUEST.confirm

10.3.33.4.1 Function

This primitive is used to enact the security and QoS resource request protocol with a specified peer MAC entity.

10.3.33.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-RESOURCE_REQUEST.confirm (
PeerMACAddress,
Content of FT Authentication Information Elements
)

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the AP that was the sender of the resource response.
Content of FT Authentication Information Elements	Sequence of Information Elements	As defined in 11A.8	The set of information elements included in the FT Ack frame, as described in 11A.8.5. This includes an optional response to a resource request (RIC).

10.3.33.4.3 When generated

This primitive is generated by the MLME on receipt of the fourth frame in the FT Resource Request Protocol.

10.3.33.4.4 Effect of receipt

Upon receipt of this primitive, the SME examines the content of the message and completes its processing of the resource request.

10.3.33.5 MLME-RESOURCE_REQUEST_LOCAL.request

10.3.33.5.1 Function

This primitive is used to enact the over-the-DS FT Resource Request Protocol for a specified peer MAC entity. The over-the-DS FT Resource Request Protocol is performed by communication between the STA and the SME of the target AP, bypassing the MAC of the target AP. This MLME function is used to allow the MAC of the target AP to process the resource requests.

10.3.33.5.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-RESOURCE_REQUEST_LOCAL.request (
MACAddress,
Content of Resource Descriptor(s)
)

Name	Type	Valid range	Description
MACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is making the resource request.
Content of Resource Descriptor(s)	Sequence of Information Elements	As defined in 11A.11.2	Specifies the resource(s) that are being requested.

10.3.33.5.3 When generated

This primitive is generated by the SME at a target AP upon receiving an over-the-DS resource request to request resources within the local MAC.

10.3.33.5.4 Effect of receipt

Upon receipt of this primitive, the MAC checks for resource availability and allocates resources as requested.

10.3.33.6 MLME-RESOURCE_REQUEST_LOCAL.confirm

10.3.33.6.1 Function

This primitive is used to respond to a local resource request for resources from the SME.

10.3.33.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESOURCE_REQUEST_LOCAL.confirm (
                                    MACAddress,
                                    Content of Resource Descriptor(s),
                                    ResultCode
                                    )
```

Name	Type	Valid range	Description
MACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is making the resource request.
Content of Resource Descriptor(s)	Sequence of Information Elements	As defined in 11A.11.2	Specifies the resource (s) that were allocated or could have been allocated.
ResultCode	Enumeration	SUCCESS, INVALID PARAMETERS, REFUSED, or UNSPECIFIED FAILURE	Indicates the result of the outcome of a resource request.

10.3.33.6.3 When generated

This primitive is generated by the MAC in response to a local resource request for resources via MLME-RESOURCE_REQUEST_LOCAL.request primitive.

10.3.33.6.4 Effect of receipt

Upon receipt of this primitive, the SME prepares a success or failure response to be sent to the STA via the current AP.

10.3.34 MLME SAP interface for remote requests

10.3.34.1 MLME-REMOTE_REQUEST.request

10.3.34.1.1 Function

This primitive is used by the SME of a non-AP STA (to send over-the-DS requests) and the SME of an AP (to send over-the-DS responses) to request the MAC to send an FT Action frame.

10.3.34.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REMOTE_REQUEST.request (
                                    PeerMACAddress,
                                    Content of FT Action Frame
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the destination of the Action frame
Content of FT Action Frame	Sequence of octets	As defined in 7.4.8	The Action frame to send to the STA.

10.3.34.1.3 When generated

This primitive is generated by the SME to send an FT Action frame to a specific peer MAC entity.

10.3.34.1.4 Effect of receipt

Upon receipt of this primitive, the MAC forwards the Action frame to the STA identified in the Action frame.

10.3.34.2 MLME-REMOTE_REQUEST.indication

10.3.34.2.1 Function

This primitive is used by the MAC to indicate to the SME the reception of an FT Action frame.

10.3.34.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REMOTE_REQUEST.indication (
                                    PeerMACAddress,
                                    Contents of FT Action Frame
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that issued the Action frame.
Content of FT Action Frame	Sequence of octets	As defined in 7.4.8	The Action frame received from the STA.

10.3.34.2.3 When generated

This primitive is generated by the MAC as a result of the receipt of an FT Action frame from a specific peer MAC entity.

10.3.34.2.4 Effect of receipt

Upon receipt of this primitive, the remote request broker (RRB) in the SME of the current AP forwards the Action frame to the target AP identified in the Action frame.

10.3.34.3 MLME-REMOTE_REQUEST.confirm

10.3.34.3.1 Function

This primitive is used by the MAC to indicate that it has completed sending an FT Action frame.

10.3.34.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REMOTE_REQUEST.confirm (
                                    PeerMACAddress,
                                    ResultCode
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the destination of the Action frame.
ResultCode	Enumeration	SUCCESS, FAIL	Indicates the status of the Action frame transmission.

10.3.34.3.3 When generated

This primitive is generated by the MAC to indicate that it has completed sending an FT Action frame.

10.3.34.3.4 Effect of receipt

Upon receipt of this primitive, the SME knows whether the Action frame was sent successfully.

11. MLME

11.3 STA authentication and association

11.3.1 Authentication and deauthentication

11.3.1.1 Authentication—originating STA

Change 11.3.1.1 as follows:

Upon receipt of an MLME-AUTHENTICATE.request primitive, the originating STA shall authenticate with the indicated STA using the following procedure:

a) In an ESS, or optionally in an IBSS, the STA shall execute one of the following:~~the authentication mechanism described in 8.2.2.2.~~:

   — For the Open System or Shared Key authentication algorithm, the authentication mechanism described in 8.2.2.2 or 8.2.2.3, respectively.

   — For the FT authentication algorithm in an ESS, the authentication mechanism described in 11A.5.

b) If the authentication was successful, the state variable for the indicated STA shall be set to State 2.

c) The STA shall issue an MLME-AUTHENTICATE.confirm primitive to inform the SME of the result of the authentication.

If the requested authentication mechanism is other than FT authentication, the~~The~~ STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) before invoking MLME-AUTHENTICATE.request primitive.

11.3.1.2 Authentication—destination STA

Change list item a) of the lettered list in the first paragraph in 11.3.1.2 as follows:

a) The STA shall execute one of the following:~~the authentication mechanism described in 8.2.2.2.~~

   — For the Open System or Shared Key authentication algorithm, the authentication mechanism described in 8.2.2.2 or 8.2.2.3, respectively.

   — For the FT authentication algorithm, the authentication mechanism described in 11A.5.

Change the second paragraph of 11.3.1.2 as follows:

If the requested authentication mechanism is other than FT authentication, the~~The~~ STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-AUTHENTICATE.indication primitive.

11.3.2 Association, reassociation, and disassociation

11.3.2.3 STA reassociation procedures

Change the final paragraph of 11.3.2.3 as follows:

Except when the association is part of a fast BSS transition, the~~The~~ STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) before invoking MLME-REASSOCIATE.request primitive.

11.3.2.4 AP reassociation procedures

Change the final paragraph of 11.3.2.4 as follows:

Except when the association is part of a fast BSS transition, the~~The~~ STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-REASSOCIATE.indication primitive.

11.4 Traffic stream (TS) operation

11.4.1 Introduction

Change the third paragraph of 11.4.1 as follows:

Traffic specification (TSPEC) and the optional traffic classification (TCLAS) elements are transported on the air by the ADDTS, in the corresponding QoS Action frame and across the MLME SAP by the MLME-ADDTS primitives. In addition, a TS could be created if a STA sends a resource request to an AP prior to initiating a transition to that AP or in the Reassociation Request frame to that AP.

11.4.3 TS lifecycle

Insert the following paragraph after the second paragraph (which starts with "Initially a TS") of 11.4.3:

A TS may be established by a Resource Request appearing in a message as part of a fast BSS transition from a STA. Such a TS is created in the accepted state. If the STA subsequently reassociates with this AP, then the TS becomes active. If the STA does not reassociate prior to the expiration of the reassociation timeout, then the TS becomes inactive.

Replace Figure 11-7 with the following:

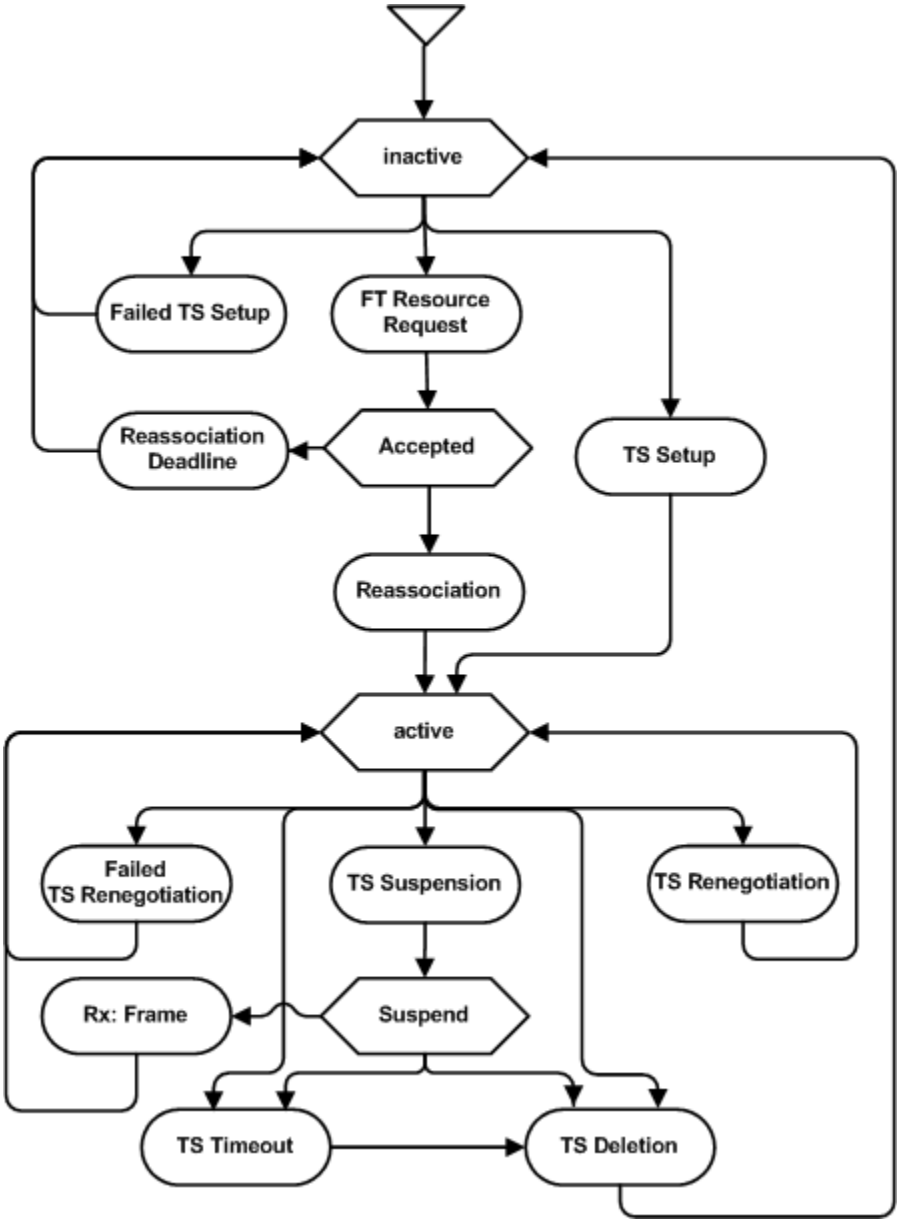

Figure 11-7—TS lifecycle

Insert the following subclause (11.4.4a) after 11.4.4:

11.4.4a TS setup by resource request during a fast BSS transition

A non-AP QoS STA may transmit a TSPEC as part of a RIC-Request in a resource request message. The SME in the hybrid coordinator (HC) decides whether to accept the TSPEC as specified, or refuse the TSPEC, or not accept but suggest an alternative TSPEC. It then generates a RIC-Response, according to the procedures given in 11A.11.

Each TS established by this resource request is placed in the accepted state. This state is an intermediate state between inactive and active. In the accepted state, the inactivity and suspension timers shall not be started for the TS. For a TS based on hybrid coordination function (HCF) controlled channel access (HCCA), the HC shall not generate CF-Poll for the TS.

The SME may take the resource/timing requirements of the TS in the accepted state into consideration before assigning any further resources to any other admitted or accepted TS, and in calculating the available admission capacity for the BSS Load information element.

The TS is moved to the active state once the STA performs a reassociation to the AP (see 11A.11.3). Once the TS becomes active, the inactivity and suspension timers are started.

If the reassociation timer times out and the TS is not yet in the active state, the TS goes back to the inactive state.

Insert the following clause (Clause 11A) after Clause 11:

11A. Fast BSS transition

11A.1 Overview

Fast BSS transition seeks to reduce the length of time that connectivity is lost between the STA and the DS during a BSS transition. The FT protocols are part of the reassociation service and only apply to STA transitions between APs within the same mobility domain within the same ESS.

The FT protocols require information to be exchanged during the initial association (or a later reassociation) between the non-AP STA and AP. The initial exchange is referred to as the *FT initial mobility domain association*. Subsequent reassociations to APs within the same mobility domain may make use of the FT protocols.

Two FT protocols are defined:

— *FT Protocol.* This protocol is executed when a STA makes a transition to a target AP and does not require a resource request prior to its transition.
— *FT Resource Request Protocol.* This protocol is executed when a STA requires a resource request prior to its transition.

For a STA to move from its current AP to a target AP utilizing the FT protocols, the message exchanges are performed using one of two methods:

— *Over-the-Air.* The STA communicates directly with the target AP using IEEE 802.11 authentication with the FT authentication algorithm.
— *Over-the-DS.* The STA communicates with the target AP via the current AP. The communication between the STA and the target AP is carried in FT Action frames between the STA and the current AP. Between the current AP and target AP, communication is via an encapsulation method described in 11A.10.3. The current AP converts between the two encapsulations.

APs advertise both capabilities and policies for supporting the FT protocols and methods.

NOTE—Throughout this clause, the notation *Authentication-Request* refers to an Authentication frame with the Authentication Transaction Sequence Number field set to 1; *Authentication-Response* refers to an Authentication frame with the Authentication Transaction Sequence Number field set to 2; *Authentication-Confirm* refers to an Authentication frame with the Authentication Transaction Sequence Number field set to 3; *Authentication-Ack* refers to an Authentication frame with the Authentication Transaction Sequence Number field set to 4. The first parameter to the above four messages is the authentication algorithm, such as Open System authentication algorithm (i.e., *Open* in figures in this clause) or FT authentication algorithm (i.e., *FTAA* in figures in this clause).

11A.2 Key holders

11A.2.1 Introduction

The FT key holder architecture, shown in Figure 11A-1, describes the FT key management entities and is defined in the context of the IEEE 802.11 basic reference model (see Figure 5-10 in 5.7).

Figure 11A-1—FT key holder architecture

The R0KH and R1KH are part of AP SME RSNA key management. The computation of PMK-R0 and PMK-R1, and all the intermediate results in the computations, shall be restricted to the R0KH. The computation of PTK, and all intermediate results in its computation, shall be restricted to the R1KH.

The S0KH and S1KH are part of the non-AP STA SME RSNA key management. The computation of PMK-R0 and PMK-R1, and all the intermediate results in the computations, shall be restricted to the S0KH. The computation of PTK, and all intermediate results in its computation, shall be restricted to the S1KH.

11A.2.2 Authenticator key holders

The R0KH and R1KH are responsible for the derivation of keys in the FT key hierarchy. For fast BSS transition, the functions of the IEEE 802.1X Authenticator are distributed among the R0KH and R1KHs.

The R0KH interacts with the IEEE 802.1X Authenticator to receive the MSK resulting from an EAP authentication. The R1KH interacts with the IEEE 802.1X Authenticator to open the Controlled Port. Both the R0KH and R1KH interactions with the IEEE 802.1X Authenticator occur within the SME.

The R0KH derives the PMK-R0 for use in the mobility domain utilizing either the MSK (when the AKM negotiated is 00-0F-AC:3) or the PSK (when the AKM negotiated is 00-0F-AC:4). The R0KH shall be responsible for deriving a PMK-R1 for each R1KH within the mobility domain.

The R1KH and S1KH each derive the PTK.

Each R0KH-ID and R1KH-ID is assumed to be expressed as a unique identifier within the mobility domain. This identifier is communicated to the non-AP STA and other key holders. The R0KH-ID is bound into the PMK-R0 derivation and the R1KH-ID is bound into the PMK-R1 derivation.

The R0KH shall meet the following requirements:

— The R0KH shall be co-located with the network access server (NAS) Client functionality of the IEEE 802.1X Authenticator.

— The R0KH-ID shall be set to the identity of the co-resident NAS Client (e.g., NAS-Identifier as defined in RFC 2865 if RADIUS is used as the backend protocol). R0KH-ID shall not be longer than 48 octets to fit in the length limitation of the FTIE.

— When the PMK-R0 lifetime expires, the R0KH shall delete the PMK-R0 security association and shall revoke within the R0KH all PMK-R1s derived from the PMK-R0.

— The R0KH shall not expose the PMK-R0 to other parties.

— The R0KH shall not expose the PMK-R1 to parties other than the authorized R1KH.

The R1KH shall meet the following requirements:

— The R1KH-ID shall be set to a MAC address of the physical entity that stores the PMK-R1 and uses it to generate the PTK. That same MAC address shall be used to advertise the PMK-R1 identity to the STA and the R0KH.

— The R1KH shall derive and distribute the GTK to all connected STAs.

— When the PMK-R1 lifetime expires, the R1KH shall delete the PMK-R1 PMKSA and shall revoke all PTKSAs derived from the PMK-R1 using the MLME-DELETEKEYS primitive.

— The R1KH shall not expose the PMK-R1 to other parties.

The management information base (MIB) variables dot11FTR0KeyHolderID and dot11FTR1KeyHolderID shall contain the values of R0KH-ID and R1KH-ID as defined in this clause, respectively.

The R0KH and the R1KH are assumed to have a secure channel between them that can be used to exchange cryptographic keys without exposure to any intermediate parties. The cryptographic strength of the secure channel between the R0KH and R1KH is assumed to be greater than or equal to the cryptographic strength of the channels for which the keys will be used. This standard assumes that the key transfer includes the PMK-R1, the PMK-R1 PMKSA, the PMK-R1 context, and the associated key authorizations. The protocol for distribution of keying material from the R0KH to the R1KH is outside the scope of this standard.

The PMK-R1 distribution from the R0KH to the R1KHs within the same mobility domain shall satisfy the following assumptions:

— The R0KH authenticates a potential R1KH with the same identity as is included in the PMK-R1 derivation. The cryptographic strength of the authentication is assumed to be greater than or equal to the cryptographic strength of the authentication between the Supplicant and AS.

— The authorization of holding a PMK-R1 is based on the authentication of the R1KH.

— The protected channel provides confidentiality and integrity protection.

11A.2.3 Supplicant key holders

The S0KH and S1KH are responsible for the derivation of keys in the FT key hierarchy. The S0KH and S1KH are entities that are assumed to physically reside in the Supplicant.

The S0KH interacts with the IEEE 802.1X functional block (see Figure 5-10 in 5.7) to receive the MSK resulting from an EAP authentication. The S1KH interacts with 802.1X to open the Controlled Port. Both the S0KH and S1KH interactions with 802.1X occur within the SME of a STA.

The S0KH derives the PMK-R0 for use in the mobility domain utilizing either the MSK (when the AKM negotiated is 00-0F-AC:3) or the PSK (when the AKM negotiated is 00-0F-AC:4).

The S1KH shall derive the PTK mutually with the R1KH.

The S0KH and S1KH shall be identified by the SPA. The S0KH shall not expose the PMK-R0 to other parties and shall not expose the PMK-R1 to parties other than the authorized S1KH. The S1KH shall not expose the PMK-R1 to other parties.

11A.3 Capability and policy advertisement

The FT capability is advertised in the Beacon and Probe Response frames by including the MDIE. The MDIE is advertised in the Beacon and Probe Response frames to indicate the MDID, FT capability, and the FT policy.

The MDID field shall be the value of dot11FTMobilityDomainID. The Fast BSS Transition Policy bits in the MDIE, i.e., Fast BSS Transition over DS subfield and Resource Request Protocol Capability subfield, shall be set according to the values of the MIB variables dot11FTOver-DSEnabled, and dot11FTResourceRequestSupported, respectively.

NOTE—It is assumed by this standard that the Fast BSS Transition Policy bits in the MDIE are administered consistently across the mobility domain.

The capability is advertised in the Neighbor Report information element. See 11.11 and 7.3.2.37.

If an FTIE is included in a Request information element in a Probe Request frame, the FTIE in the Probe Response frame shall contain the R0KH-ID and R1KH-ID (set according to the values of the MIB variables dot11FTR0KeyHolderID and dot11FTR1KeyHolderID), and all other fields shall be set to 0.

11A.4 FT initial mobility domain association

11A.4.1 Overview

The FT initial mobility domain association is the first (re)association in the mobility domain, where the SME of the non-AP STA enables its future use of the FT procedures.

FT initial mobility domain association will typically be the first association within the ESS. In addition to association frames, reassociation frames are supported in the initial mobility domain association to enable both FT and non-FT APs to be present in a single ESS.

11A.4.2 FT initial mobility domain association in an RSN

The STA indicates its support for the FT procedures by including the MDIE in the (Re)Association Request frame and indicates its support of security by including the RSNIE. The AP responds by including the FTIE, MDIE, and RSNIE in the (Re)Association Response frame. After a successful IEEE 802.1X authentication (if needed), the STA and AP perform an FT 4-Way Handshake. At the end of the sequence, the IEEE 802.1X Controlled Port is opened, and the FT key hierarchy has been established. The message flow is shown in Figure 11A-2.

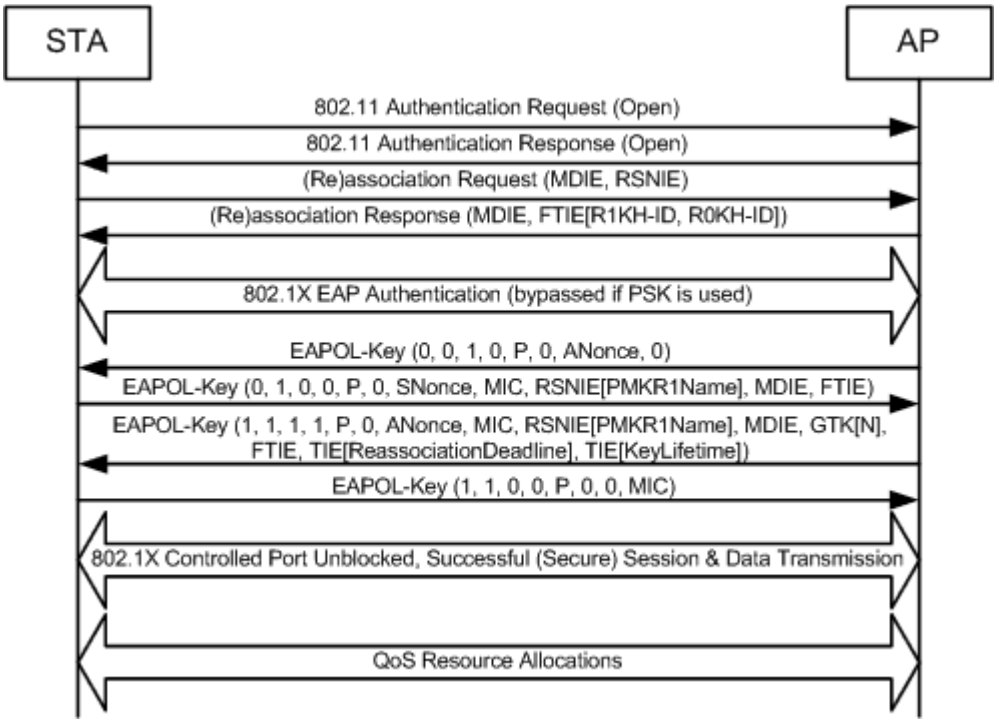

Figure 11A-2—FT initial mobility domain association in an RSN

The STA initiates the FT initial mobility domain association procedures by performing an IEEE 802.11 authentication using the Open System authentication algorithm.

STA→AP: Authentication-Request (Open System authentication algorithm)
AP→STA: Authentication-Response (Open System authentication algorithm, Status)

The SME of the STA initiates the authentication exchange, through the use of the primitive MLME-AUTHENTICATE.request, and the SME of the AP responds with MLME-AUTHENTICATE.response primitive. See 11.3.1.

Upon successful IEEE 802.11 Open System authentication, the STA shall send a (Re)Association Request frame to the AP that includes the MDIE. The contents of the MDIE shall be the values advertised by the AP in its Beacon or Probe Response frames. Additionally, the STA includes its security capabilities in the RSNIE.

STA→AP: (Re)Association Request (MDIE, RSNIE)
AP→STA: (Re)Association Response (MDIE, FTIE[R1KH-ID, R0KH-ID])

The SME of the STA initiates the (re)association through the use of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive. The SME of the AP responds to the indication with MLME-ASSOCIATE.response or MLME-REASSOCIATE.response primitive. See 11.3.2.

If the contents of the MDIE received by the AP do not match the contents advertised in the Beacon and Probe Response frames, the AP shall reject the (Re)Association Request frame with status code 54 (i.e., Invalid MDIE). If an MDIE is present in the (Re)Association Request frame and the contents of the RSNIE do not indicate a negotiated AKM of Fast BSS Transition (suite type 00-0F-AC:3 or 00-0F-AC:4), the AP shall reject the (Re)Association Request frame with status code 43 (i.e., Invalid AKMP).

The (Re)Association Response frame from the AP shall contain an MDIE, with contents as presented in Beacon and Probe Response frames. The FTIE shall include the key holder identities of the AP, the R0KH-ID and R1KH-ID, set to the values of dot11FTR0KeyHolderID and dot11FTR1KeyHolderID, respectively. The FTIE shall have a MIC information element count of zero (i.e., no MIC present) and have ANonce, SNonce, and MIC fields set to 0.

On successful (re)association, the S0KH on the STA and the R0KH on the AP then proceed with an IEEE 802.1X authentication using EAPOL messages carried in IEEE 802.11 data frames. The S0KH shall use the value of R0KH-ID as the endpoint identifier of the NAS Client (NAS-Identifier if RADIUS is used) in the exchange as defined in IETF RFC 3748-2004 [B26].

Upon successful completion of the IEEE 802.1X authentication, the R0KH receives the MSK and authorization attributes. If a key hierarchy already exists for this non-AP STA belonging to the same mobility domain (i.e., having the same MDID), the R0KH shall delete the existing PMK-R0 security association and PMK-R1 security associations. It then calculates the PMK-R0, PMKR0Name, and PMK-R1 and makes the PMK-R1 available to the R1KH of the AP with which the STA is associated.

If the SME of the STA cannot authenticate the AS, then it shall disassociate with an MLME-DISASSOCIATE.request primitive. If the AS signals the Authenticator that the STA cannot be authenticated, then the SME of the AP shall disassociate with an MLME-DISASSOCIATE.request primitive.

If the MSK lifetime attribute is provided by the AS, the lifetime of the PMK-R0 shall not be more than the lifetime of the MSK. If the MSK lifetime attribute is not provided, the PMK-R0 lifetime shall be the value of the MIB variable dot11FTR0KeyLifetime. For PSK, the PMK-R0 lifetime shall be the value of the MIB variable dot11FTR0KeyLifetime. The lifetime of the PMK-R1s and PTK shall be the same as the lifetime of PMK-R0. When the key lifetime expires, each key holder shall delete its respective PMK-R0, PMK-R1, and PTK SAs.

The R1KH and S1KH then perform an FT 4-Way Handshake. The EAPOL-Key frame notation is defined in 8.5.2.1.

R1KH→S1KH: Data(EAPOL-Key(0, 0, 1, 0, P, 0, 0, ANonce, 0))
S1KH→R1KH: Data(EAPOL-Key(0, 1, 0, 0, P, 0, 0, SNonce, MIC,
RSNIE[PMKR1Name], MDIE, FTIE))
R1KH→S1KH: Data(EAPOL-Key(1, 1, 1, 1, P, 0, 0, ANonce, MIC,
RSNIE[PMKR1Name], MDIE, GTK[N], FTIE,
TIE[ReassociationDeadline], TIE[KeyLifetime]))
S1KH→R1KH: Data(EAPOL-Key(1, 1, 0, 0, P, 0, 0, 0, MIC))

The message sequence is similar to that of 8.5.3. The contents of each message shall be as described in 8.5.3 except as follows:

— Message 2: the S1KH shall include the PMKR1Name in the PMKID field of the RSNIE. The PMKR1Name shall be as calculated by the S1KH according to the procedures of 8.5.1.5.4; all other fields of the RSNIE shall be identical to the RSNIE present in the (Re)Association Request frame. The S1KH shall include the FTIE and MDIE; the FTIE and MDIE shall be the same as those provided in the AP's (Re)Association Response frame.

— Message 3: the R1KH shall include the PMKR1Name in the PMKID field of the RSNIE. The PMKR1Name shall be as calculated by the R1KH according to the procedures of 8.5.1.5.4 and shall be the same as the PMKR1Name in Message #2; all other fields of the RSNIE shall be identical to the RSNIE present in the Beacon or Probe Response frames. The R1KH shall also include the FTIE, the MDIE, the reassociation deadline timeout in the TIE[ReassociationDeadline], and the PTK key lifetime in the TIE[KeyLifetime]. The FTIE and MDIE shall be the same as in the (Re)Association

Response frame. The reassociation deadline shall be set to the minimum of dot11FTReassociationDeadline and the key lifetime.

NOTE—"Data()" indicates the message is an IEEE 802.11 data frame.

It is assumed by this standard that the reassociation deadline is administered consistently across the mobility domain. The mechanism for such consistent administration is outside the scope of this standard.

The PTK shall be calculated by the R1KH and S1KH according to the procedures given in 8.5.1.5.5.

Upon completion of a successful FT 4-Way Handshake, the IEEE 802.1X Controlled Port shall be opened on both the non-AP STA and the AP. Subsequent EAPOL-Key frames shall use the key replay counter to detect replayed messages.

Upon completion of a successful FT 4-Way Handshake, the PTK key lifetime timer is initiated to ensure that the lifetime of the PTKSA is no longer than the value provided in the TIE[KeyLifetime] sent in Message 3.

Once the PTKSA key lifetime expires, as indicated by the TIE[KeyLifetime], to continue its association in the mobility domain the non-AP STA shall perform the FT initial mobility domain association procedures. If the AP sends a Deauthentication or Disassociation frame to the non-AP STA with reason code 2 (i.e., Previous authentication no longer valid), then to continue its association in the mobility domain, the non-AP STA shall perform the FT initial mobility domain association procedures with any AP in the mobility domain. If the Supplicant EAPOL state machines are triggered to send an EAPOL-Start packet after a successful initial mobility domain association, the non-AP STA shall perform the FT initial mobility domain association procedures.

11A.4.3 FT initial mobility domain association in a non-RSN

In this sequence, the STA utilizes the FT procedures by including the MDIE in the (Re)Association Request frame. The AP responds by including the MDIE in the (Re)Association Response frame. The message flow is shown in Figure 11A-3.

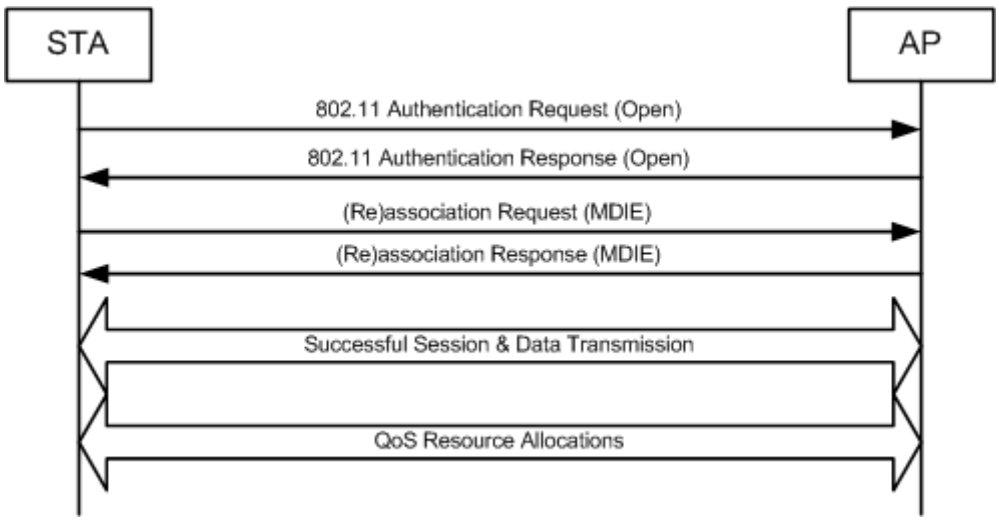

Figure 11A-3—FT initial mobility domain association in a non-RSN

The STA initiates the FT initial mobility domain association procedures by performing an IEEE 802.11 authentication using the Open System authentication algorithm.

STA→AP: Authentication-Request (Open System authentication algorithm)
AP→STA: Authentication-Response (Open System authentication algorithm, Status)

The SME of the STA initiates the authentication exchange through the use of the primitive MLME-AUTHENTICATE.request primitive, and the SME of the AP responds with MLME-AUTHENTICATE.response primitive. See 11.3.1.

Upon successful IEEE 802.11 Open System authentication, the STA shall send a (Re)Association Request frame to the AP and shall include the MDIE. The contents of the MDIE shall be the values advertised by the AP in its Beacon or Probe Response frames.

STA→AP: (Re)Association Request (MDIE)
AP→STA: (Re)Association Response (MDIE)

The SME of the STA initiates the (Re)association through the use of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive. The SME of the AP responds to the indication with MLME-ASSOCIATE.response or MLME-REASSOCIATE.response primitive. See 11.3.2.

If the contents of the MDIE received by the AP do not match the contents advertised in the Beacon and Probe Response frames, the AP shall reject the (Re)Association Request frame with status code 54 (i.e., Invalid MDIE).

The (Re)Association Response frame from the AP shall contain an MDIE, with contents as presented in Beacon and Probe Response frames.

On successful (re)association, the AP and the non-AP STA shall transition to State 3 (as defined in 11.3) to enable data frame transmission.

11A.5 FT Protocol

11A.5.1 Overview

STAs with dot11FastBSSTransitionEnabled set to TRUE shall support the FT Protocol.

The FT Protocol supports resource requests as part of the reassociation. The optional FT Resource Request Protocol (see 11A.6) supports resource requests prior to reassociation.

A STA shall not use any authentication algorithm except the FT authentication algorithm when using the FT Protocol.

11A.5.2 Over-the-air FT Protocol authentication in an RSN

The over-the-air FT Protocol in an RSN is shown in Figure 11A-4.

The STA and AP use the FT authentication sequence to specify the PMK-R1 security association and to provide values of SNonce and ANonce that enable a liveness proof, replay protection, and PTK key separation. This exchange enables a fresh PTK to be computed in advance of reassociation. The PTKSA is used to protect the subsequent reassociation transaction, including the optional RIC-Request.

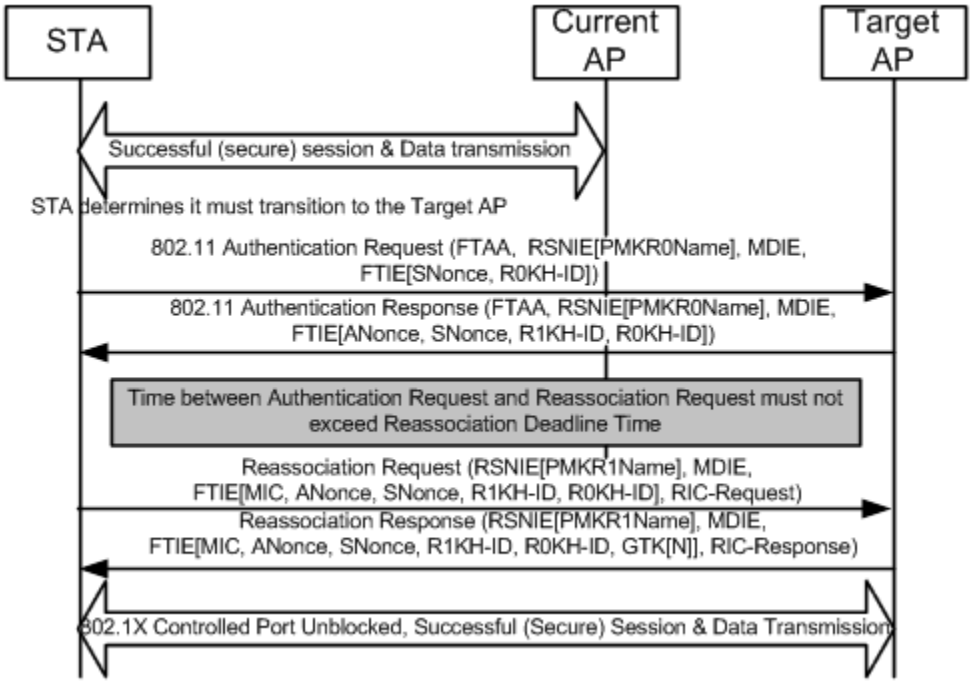

Figure 11A-4—Over-the-air FT Protocol in an RSN

To perform an over-the-air fast BSS transition to a target AP, the STA and target AP shall perform the following exchange:

STA→Target AP: Authentication-Request (FTAA, 0, RSNIE[PMKR0Name], MDIE, FTIE[SNonce, R0KH-ID])

Target AP→STA: Authentication-Response (FTAA, Status, RSNIE[PMKR0Name], MDIE, FTIE[ANonce, SNonce, R1KH-ID, R0KH-ID])

The SME of the STA initiates the authentication exchange, through the use of the MLME-AUTHENTICATE.request primitive, and the SME of the AP responds with an MLME-AUTHENTICATE.response primitive. See 11.3.1. The MLME primitives for Authentication when the FT authentication algorithm is selected use only Authentication transaction sequence number values 1 and 2.

In the Authentication Request frame, the SA field of the message header shall be set to the MAC address of the STA, and the DA field of the message header shall be set to the BSSID of the target AP. The information elements in the frame, and their required contents, shall be as given in 11A.8.2.

If the contents of the MDIE received by the AP do not match the contents advertised in the Beacon and Probe Response frames, the AP shall reject the Authentication Request with status code 54 (i.e., Invalid MDIE). If the Authentication Request frame contains an authentication algorithm set to FT authentication and the contents of the RSNIE do not indicate a negotiated AKM of Fast BSS Transition (suite type 00-0F-AC:3 or 00-0F-AC:4), the AP shall reject the Authentication Request with status code 43 (i.e., Invalid AKMP). If the FTIE in the FT Request frame contains an invalid R0KH-ID, the AP shall reject the FT Request frame with status code 55 (i.e., Invalid FTIE). If the RSNIE in the Authentication Request frame contains an invalid PMKR0Name and the AP has determined that it is an invalid PMKR0Name, the AP shall reject the Authentication Request with status code 53 (i.e., Invalid PMKID). If the requested R0KH is not reachable, the AP shall respond to the Authentication Request with status code 28 (i.e., R0KH unreachable). If the non-AP STA selects a pairwise cipher suite in the RSNIE that is different from the ones used in the Initial mobility domain association, then the AP shall reject the Authentication Request with status code 19

(i.e., Invalid Pairwise Cipher). Subsequent to a rejection of an Authentication Request, the STA may retry the Authentication Request.

In the Authentication Response frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the STA. The Status Code field shall be a value from the options listed in 7.3.1.9. The information elements in the frame, and their required contents, shall be as given in 11A.8.3.

The R1KH of the target AP uses the value of PMKR0Name and other information in the frame to calculate PMKR1Name. If the target AP does not have the key identified by PMKR1Name, it may retrieve that key from the R0KH identified by the STA. See 11A.2. Upon receiving a new PMK-R1 for a STA, the target AP shall delete the prior PMK-R1 security association and PTKSAs derived from the prior PMK-R1.

The STA and the target AP compute the PTK and PTKName using the PMK-R1, PMKR1Name, ANonce, and SNonce, as specified in 8.5.1.5.5. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the STA within the reassociation deadline timeout value.

If the STA does not receive a response to the Authentication Request frame, it may reissue the request following the restrictions given for Authentication frames in 11.3. If the Status Code field value returned by the target AP is 0, indicating success, the STA and target AP transition to State 2 (as defined in 11.3); the STA may continue with reassociation (11A.7.1). Handling of errors returned in the Status Code field shall be as specified in 11.3.

11A.5.3 Over-the-DS FT Protocol authentication in an RSN

A STA shall not initiate an over-the-DS FT authentication to a target AP whose MDIE contains the Fast BSS Transition over DS bit set to 0.

The over-the-DS FT Protocol in an RSN is shown in Figure 11A-5.

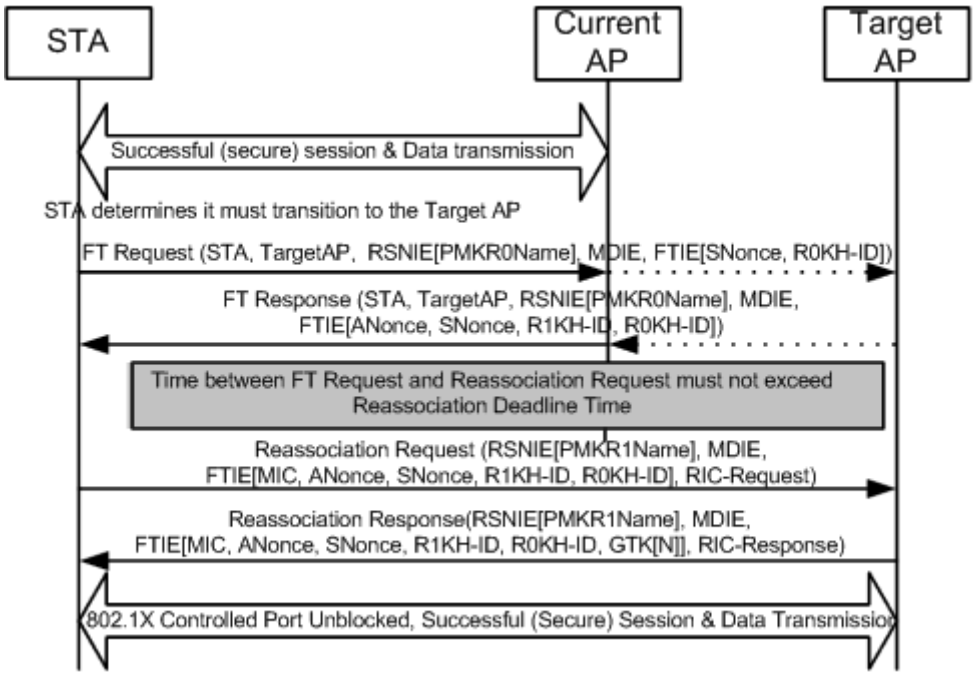

Figure 11A-5—Over-the-DS FT Protocol in an RSN

To perform an over-the-DS fast BSS transition to a target AP, the STA and the target AP (through the current AP) shall perform the following exchange:

STA→Target AP: FT Request (STA address, TargetAP address, RSNIE[PMKR0Name], MDIE, FTIE[SNonce, R0KH-ID])

Target AP→STA: FT Response (STA address, TargetAP address, Status, RSNIE[PMKR0Name], MDIE, FTIE[ANonce, SNonce, R1KH-ID, R0KH-ID])

The SME of the non-AP STA initiates the FT Request frame to the target AP by issuing a MLME-REMOTE_REQUEST.request primitive with parameters including the contents of the FT Request frame (FT Action frame with an Action field value indicating FT Request) to be sent. The MAC of the non-AP STA transmits this Action frame and issues a MLME-REMOTE_REQUEST.confirm primitive to signal that it has been sent. For processing at the current AP and target AP see 11A.10. When the MAC of the non-AP STA receives the FT Response frame (FT Action frame with an Action field value indicating FT Response), it passes it to the SME by use of MLME-REMOTE_REQUEST.indication primitive, with parameters including the contents of the received Action frame. The MLME interfaces on the non-AP STA, current AP, and the target AP for executing the over-the-DS fast BSS transition are shown in Figure 11A-6.

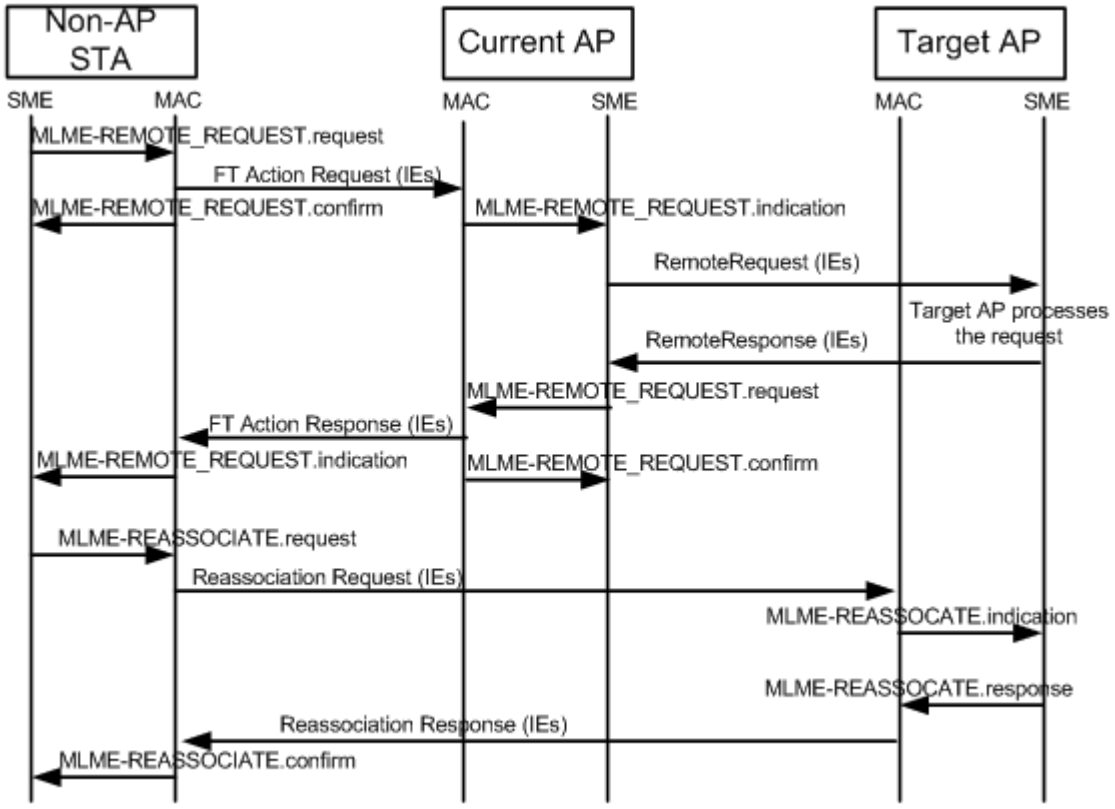

Figure 11A-6—MLME interfaces for over-the-DS FT Protocol messages

The STA Address field of the FT Request frame shall be set to the MAC address of the STA, and the Target AP Address field of the FT Request frame shall be set to the BSSID of the target AP. The information elements in the FT Request frame, and their required contents, shall be as given in 11A.8.2.

If the contents of the MDIE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the FT Request frame with status code 54 (i.e., Invalid MDIE). If the contents of the RSNIE do not indicate a negotiated AKM of Fast BSS Transition (suite type

00-0F-AC:3 or 00-0F-AC:4), the AP shall reject the FT Request frame with status code 43 (i.e., Invalid AKMP). If the FTIE in the FT Request frame contains an invalid R0KH-ID, the AP shall reject the FT Request frame with status code 55 (i.e., Invalid FTIE). If the RSNIE in the FT Request frame contains an invalid PMKR0Name, and the AP has determined that it is an invalid PMKR0Name, the AP shall reject the Authentication Request with status code 53 (i.e., Invalid PMKID). If the requested R0KH is not reachable, the AP shall respond to the FT Request frame with status code 28 (i.e., R0KH unreachable). The AP may reject the FT Request frame for limiting the non-AP STA's reassociation to this AP by using the status code 37 ("This request has been declined"). If the non-AP STA selects a pairwise cipher suite in the RSNIE that is different from the ones used in the initial mobility domain association, then the AP shall reject the FT Request frame with status code 19 (i.e., Invalid Pairwise Cipher).

The STA Address field of the FT Response frame shall be set to the MAC address of the non-AP STA, and the Target AP Address field of the FT Response frame shall be set to the BSSID of the target AP. The information elements in the FT Response frame, and their required contents, shall be as given in 11A.8.3. The Status Code field shall be a value from the options listed in 7.3.1.9.

The R1KH of the target AP uses the value of PMKR0Name and other information from the frame to calculate PMKR1Name. If the target AP does not have the key identified by PMKR1Name, it may retrieve that key from the R0KH identified by the non-AP STA. See 11A.2. Upon receiving a new PMK-R1 for a non-AP STA, the target AP shall delete the prior PMK-R1 security association and PTKSAs derived from the prior PMK-R1.

The non-AP STA and the target AP compute the PTK and PTKName using the PMK-R1, PMKR1Name, ANonce, and SNonce, as specified in 8.5.1.5.5. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the STA within the reassociation deadline timeout value.

If the non-AP STA does not receive a response to the FT Request frame, it may reissue the request following the restrictions given for Authentication frames in 11.3. If the Status Code field value returned by the target AP is 0, indicating success, the STA and target AP transition to State 2 (as defined in 11.3); the STA may continue with reassociation (11A.7.1). Handling of errors returned in the Status Code field shall be as specified for Authentication frames in 11.3.

11A.5.4 Over-the-air FT Protocol authentication in a non-RSN

The over-the-air FT Protocol in a non-RSN is shown in Figure 11A-7.

To perform an over-the-air fast BSS transition to a target AP in a non-RSN, the STA and target AP shall perform the following exchange:

STA→Target AP: Authentication-Request (FTAA, 0, MDIE)

Target AP→STA: Authentication-Response (FTAA, Status, MDIE)

In the Authentication Request frame, the SA field of the message header shall be set to the MAC address of the STA, and the DA field of the message header shall be set to the BSSID of the target AP. The information elements in the frame, and their required contents, shall be as given in 11A.8.2.

If the contents of the MDIE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the Authentication Request with status code 54 (i.e., Invalid MDIE).

In the Authentication Response frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the STA. The Status Code field shall be a value from the options listed in 7.3.1.9. The information elements in the frame, and their required contents, shall be as given in 11A.8.3.

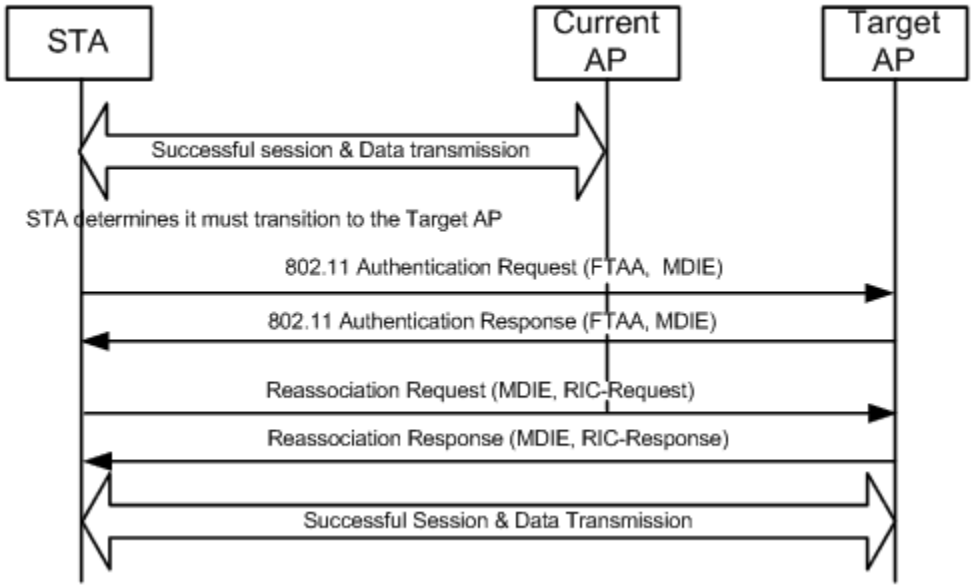

Figure 11A-7—Over-the-air FT Protocol in a non-RSN

If the STA does not receive a response to the Authentication Request frame, it may reissue the request following the restrictions given for Authentication frames in 11.3. If the Status Code field value returned by the target AP is 0, indicating success, the STA and target AP transition to State 2 (as defined in 11.3); the STA may continue with reassociation (11A.7.2). Handling of errors returned in the Status Code field shall be as specified in 11.3.

11A.5.5 Over-the-DS FT Protocol authentication in a non-RSN

The over-the-DS FT Protocol in a non-RSN is shown in Figure 11A-8.

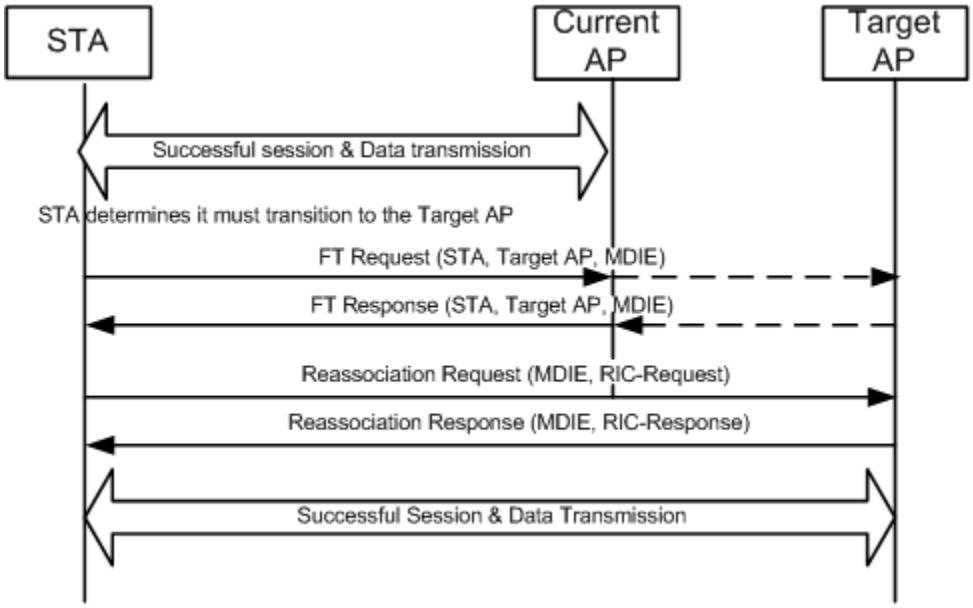

Figure 11A-8—Over-the-DS FT Protocol in a non-RSN

To perform an over-the-DS fast BSS transition to a target AP in a non-RSN, the STA and the target AP (through the current AP) shall perform the following exchange:

STA→Target AP: FT Request(STA, TargetAP, MDIE)

Target AP→STA: FT Response(STA, TargetAP, Status, MDIE)

The STA Address field of the FT Request frame shall be set to the MAC address of the STA, and the Target AP Address field of the FT Request frame shall be set to the BSSID of the target AP. The information elements in the FT Request frame, and their required contents, shall be as given in 11A.8.2.

If the contents of the MDIE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the FT Request frame with status code 54 (i.e., Invalid MDIE).

The STA Address field of the FT Response frame shall be set to the MAC address of the STA, and the Target AP Address field of the FT Response frame shall be set to the BSSID of the target AP. The information elements in the FT Response frame, and their required contents, shall be as given in 11A.8.3. The Status Code field shall be a value from the options listed in 7.3.1.9.

If the STA does not receive a response to the FT Request frame, it may reissue the request following the restrictions given for Authentication frames in 11.3. If the Status Code field value returned by the target AP is 0, indicating success, the STA and target AP transition to State 2 (as defined in 11.3); the STA may continue with reassociation (11A.7.2). Handling of errors returned in the Status Code field shall be as specified for Authentication frames in 11.3.

11A.6 FT Resource Request Protocol

11A.6.1 Overview

The FT Resource Request Protocol involves an additional message exchange after the Authentication Request/Response frame, or FT Request/Response frame, and prior to reassociation.

APs capable of fast BSS transition may allow STAs to request resources prior to reassociation. Availability of the FT Resource Request Protocol is advertised by the target AP in the MDIE. If the Resource Request Protocol Capability subfield is set to 0, then the STA shall not send an Authentication Confirm nor FT Confirm frame to the AP. An AP that receives an Authentication Confirm or FT Confirm frame from a STA and does not support the FT Resource Request Protocol shall respond with status code 38 (i.e., the request has not been successful as one or more parameters have invalid values).

The additional message exchange for the FT Resource Request Protocol shall be performed using the same method (over-the-air or over-the-DS) as was used for the Authentication Request/Response frame or FT Request/Response frame. An AP that receives an FT Confirm frame that did not previously receive an FT Request frame from the same STA shall reject the request with status code 52 (i.e., Invalid FT Action Frame Count). An AP that receives an Authentication Confirm frame that did not previously receive an Authentication Request frame from the same STA shall reject the request with status code 14 (i.e., Received an Authentication frame with authentication transaction sequence number out of expected sequence).

11A.6.2 Over-the-air fast BSS transition with resource request

The over-the-air FT Resource Request Protocol in an RSN is shown in Figure 11A-9.

The over-the-air FT Resource Request Protocol in a non-RSN is shown in Figure 11A-10.

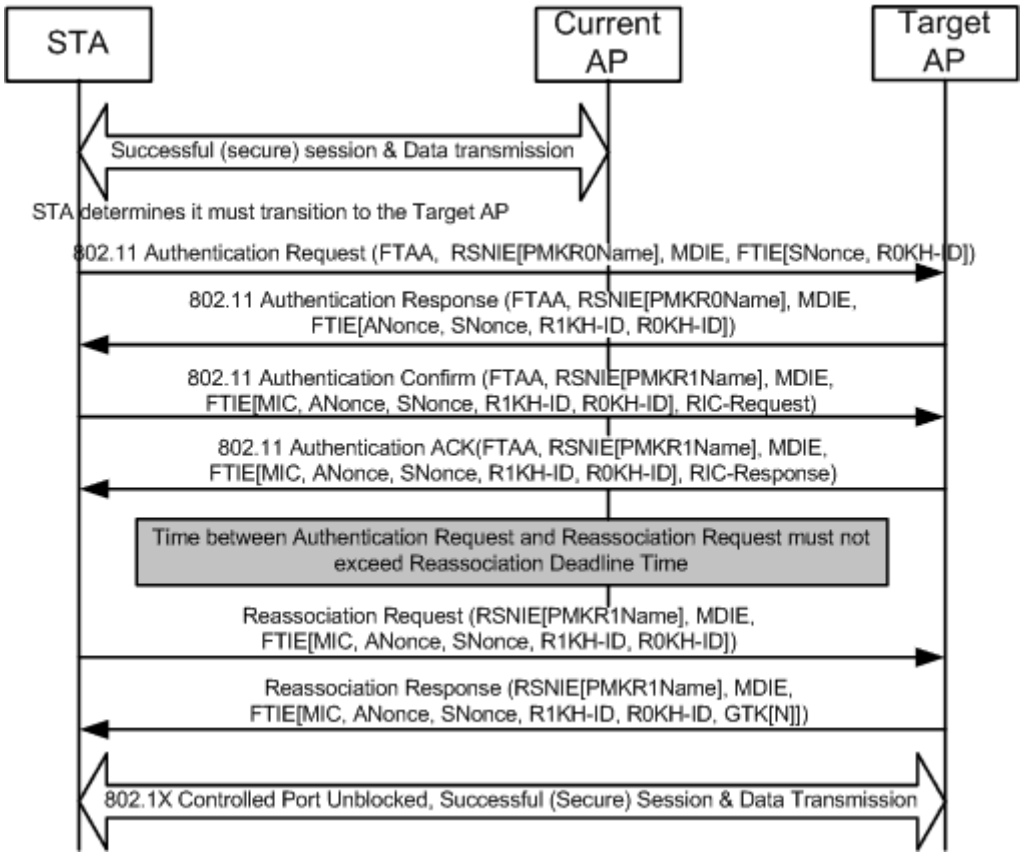

Figure 11A-9—Over-the-air FT Resource Request Protocol in an RSN

To perform an over-the-air FT Resource Request Protocol to a target AP, after completing the Authentication Request/Response exchange given in 11A.5.2 or 11A.5.4, the STA and target AP shall perform the following exchange:

STA→Target AP: Authentication-Confirm (FTAA, 0, RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)

Target AP→STA: Authentication-Ack (FTAA, Status, RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Response)

The SME of the STA initiates the resource request exchange through the use of the primitive MLME-RESOURCE_REQUEST.request primitive, and the SME of the AP responds with MLME-RESOURCE_ REQUEST.response primitive.

In the Authentication Confirm frame, the SA field of the message header shall be set to the MAC address of the STA, and the DA field of the message header shall be set to the BSSID of the target AP. In a non-RSN, the FTIE and RSNIE shall not be present. The information elements in the frame, the information element contents, and MIC calculation shall be as given in 11A.8.4.

If the contents of the MDIE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the Authentication Confirm frame with status code 54 (i.e., Invalid MDIE).

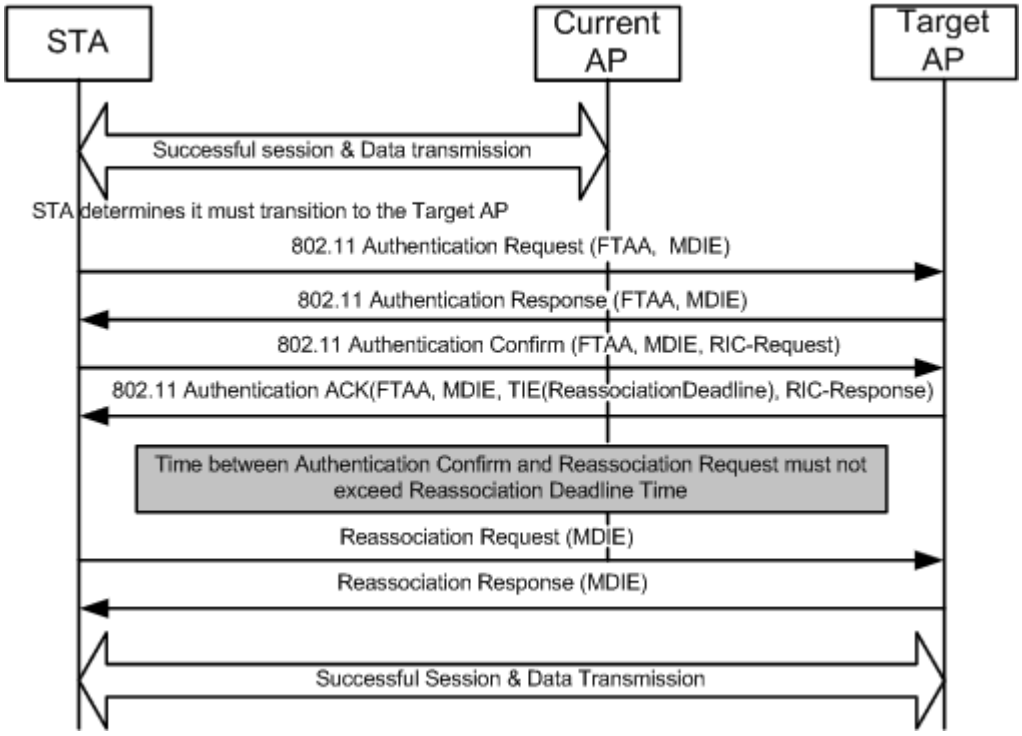

Figure 11A-10—Over-the-air FT Resource Request Protocol in a non-RSN

In an RSN, the R1KH of the target AP verifies the MIC in the FTIE in the Authentication Confirm frame and shall discard the request if it is incorrect. If the FTIE in the Authentication Confirm frame contains a different R0KH-ID, R1KH-ID, ANonce, or SNonce, the AP shall reject the Authentication Confirm frame with status code 55 (i.e., Invalid FTIE). If the RSNIE in the Authentication Confirm frame contains an invalid PMKR1Name, the AP shall reject the Authentication Confirm frame with status code 53 (i.e., Invalid PMKID).

In the Authentication Ack frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the STA. In a non-RSN, the FTIE and RSNIE shall not be present. The Status Code field shall be a value from the options listed in 7.3.1.9. The information elements in the frame, the information element contents, and MIC calculation shall be as given in 11A.8.5.

In an RSN, the S1KH of the STA verifies the MIC in the FTIE in the Authentication Ack frame and shall discard the response if the MIC is incorrect.

The STA may make a request for resources by including a RIC-Request (see 11A.11) in the Authentication Confirm frame. The RIC-Request is generated by the procedures of 11A.11.3.1, and the RIC-Response is generated by the procedures of 11A.11.3.2.

If the value of the Status Code field returned by the target AP in the Authentication Ack frame is nonzero, then the STA shall abandon this transition attempt.

In an RSN, on successful completion of the FT authentication exchange of the FT Resource Request Protocol, the PTKSA has been established and proven live. The key replay counter shall be initialized to zero, and the subsequent EAPOL-Key frames (e.g., GTK updates) shall use the key replay counter to ensure

they are not replayed. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the STA within the reassociation deadline timeout value.

In a non-RSN, the Authentication Ack frame contains a TIE with a reassociation deadline. If the STA does not send a Reassociation Request frame to the target AP within that interval, the STA shall abandon this transition attempt.

The exchange between the STA and the target AP may continue with reassociation (11A.7.1 or 11A.7.2).

11A.6.3 Over-the-DS fast BSS transition with resource request

The over-the-DS FT Resource Request Protocol in an RSN is shown in Figure 11A-11.

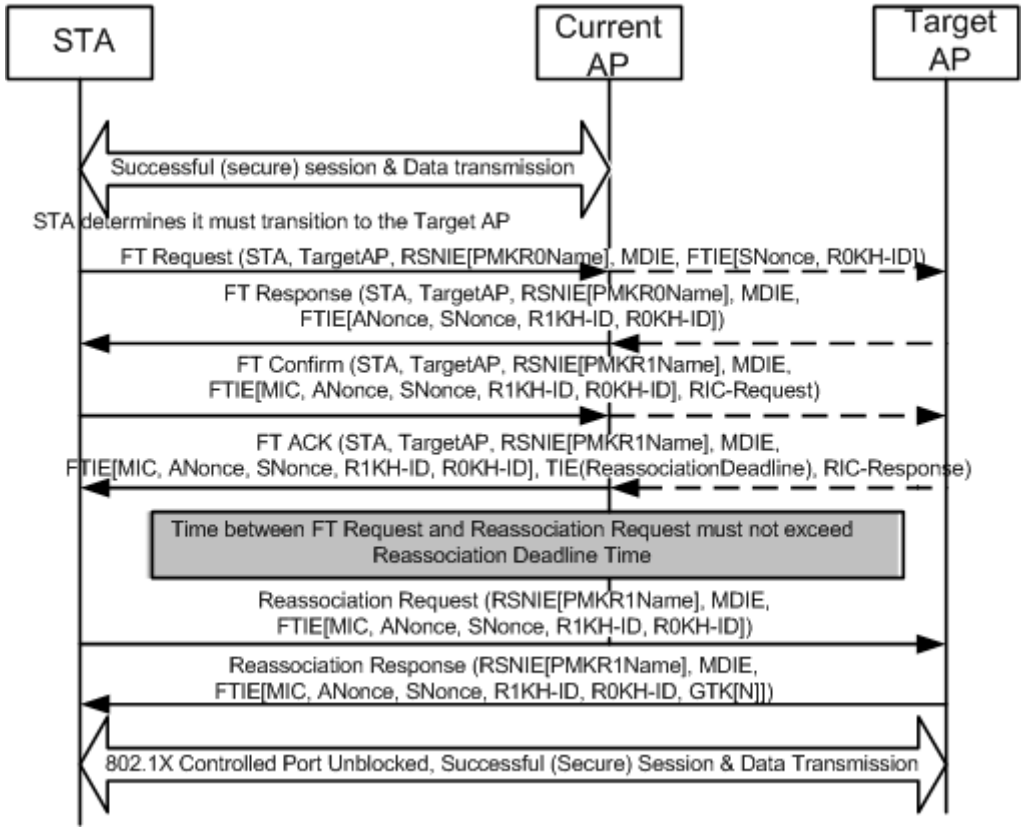

Figure 11A-11—Over-the-DS FT Resource Request Protocol in an RSN

The over-the-DS FT Resource Request Protocol in a non-RSN is shown in Figure 11A-12.

To perform an Over-the-DS FT Resource Request Protocol to a target AP, after completing the FT Request/ Response frame exchange given in 11A.5.3 or 11A.5.5, the STA and target AP (through the current AP) shall perform the following exchange, using the mechanism described in 11A.10:

STA→Target AP: FT Confirm (STA, TargetAP, RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)

Target AP→STA: FT Ack (STA, TargetAP, Status, RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], TIE[ReassociationDeadline], RIC-Response)

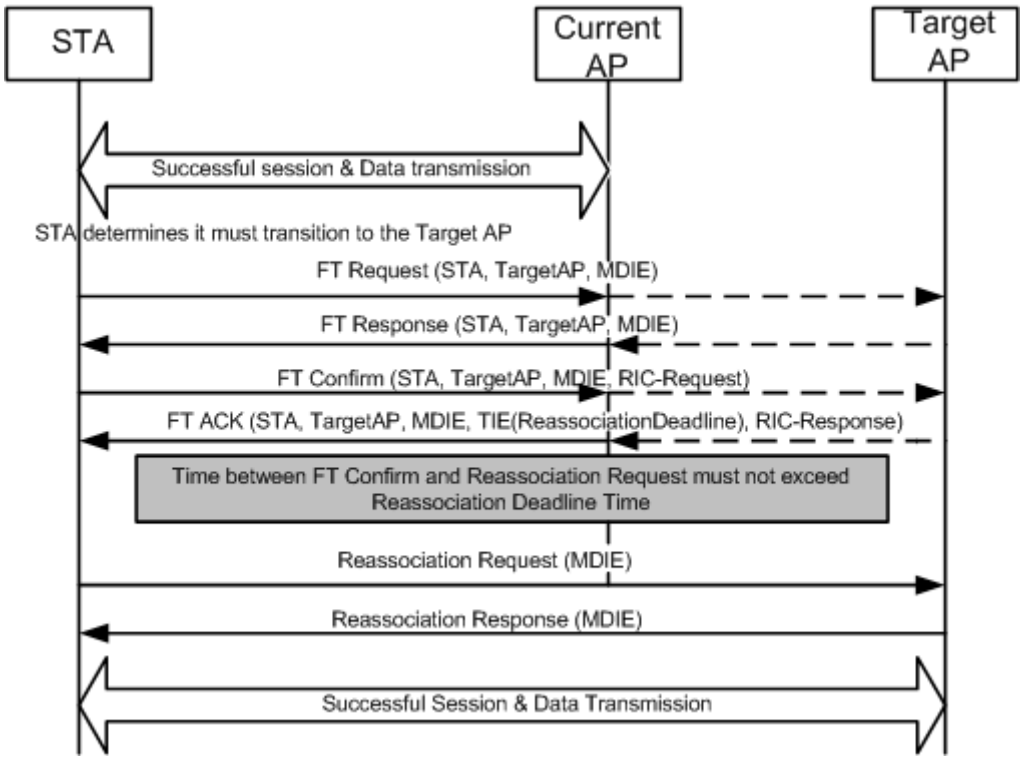

Figure 11A-12—Over-the-DS FT Resource Request Protocol in a non-RSN

The SME of the STA initiates the FT Confirm frame to the target AP by issuing a MLME-REMOTE_REQUEST.request primitive with parameters including the contents of the FT Confirm frame (FT Action frame with an Action field value indicating FT Confirm) to be sent. The MAC of the STA transmits this Action frame and issues a MLME-REMOTE_REQUEST.confirm primitive to signal that it has been sent. For processing at the current AP and target AP, see 11A.10. When the MAC of the STA receives the FT Ack frame (FT Action frame with an Action field value indicating FT Ack), it passes it to the SME by use of an MLME-REMOTE_REQUEST.indication primitive, with parameters including the contents of the received Action frame.

The STA Address field of the FT Confirm frame shall be set to the MAC address of the STA, and the Target AP Address field of the FT Confirm frame shall be set to the BSSID of the target AP. The information elements in the FT Confirm frame, the information element contents, and the MIC calculation shall be as given in 11A.8.4. In a non-RSN, the FTIE and RSNIE shall not be present.

If the contents of the MDIE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the FT Confirm frame with status code 54 (i.e., Invalid MDIE).

In an RSN, the R1KH of the target AP verifies the MIC in the FTIE and shall discard the request if it is incorrect. If the FTIE in the FT Confirm frame contains a different R0KH-ID, R1KH-ID, ANonce, or SNonce from the values sent in the FT Response frame, the AP shall reject the FT Confirm frame with status code 55 (i.e., Invalid FTIE). If the RSNIE in the FT Confirm frame contains an invalid PMKR1Name, the AP shall reject the FT Confirm frame with status code 53 (i.e., Invalid PMKID).

The STA Address field of the FT Ack frame shall be set to the MAC address of the STA, and the Target AP Address field of the FT Ack frame shall be set to the BSSID of the target AP. The information elements in

the FT Ack frame, the information element contents, and the MIC calculation shall be as given in 11A.8.5. In a non-RSN, the FTIE and RSNIE shall not be present. The Status Code field value shall be a value from the options listed in 7.3.1.9, and a TIE may appear.

In an RSN, the S1KH of the STA verifies the MIC in the FTIE in the FT Ack frame and shall discard the response if the MIC is incorrect.

The STA may make a request for resources by including a RIC-Request (see 11A.11) in the FT Confirm frame. The RIC-Request is generated by the procedures of 11A.11.3.1, and the RIC-Response is generated by the procedures of 11A.11.3.2.

In order to recover from over-the-DS packet losses, the STA may retransmit the FT Confirm frame until the reassociation deadline time is reached. If the STA does not receive a response to the FT Confirm frame or if the value of the Status Code field returned by the target AP in the FT Ack frame is nonzero, then the STA shall abandon this transition attempt.

In an RSN, on successful completion of the FT Confirm/Acknowledgment frame exchange, the PTKSA has been established and proven live. The key replay counter shall be initialized to zero, and the subsequent EAPOL-Key frames (e.g., GTK updates) shall use the key replay counter to ensure they are not replayed. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the STA within the reassociation deadline timeout value. Resource request procedures are specified in 11A.11.

In a non-RSN, the FT Ack frame contains a TIE with a reassociation deadline. If the STA does not send a Reassociation Request frame to the target AP within that interval, the STA shall abandon this transition attempt.

The exchange between the STA and the target AP may continue with reassociation (11A.7.1 or 11A.7.2).

11A.7 FT reassociation

11A.7.1 FT reassociation in an RSN

If the non-AP STA does not send a Reassociation Request frame to the target AP within the reassociation deadline interval received during the FT initial mobility domain association, the target AP may delete the PTKSA, and the non-AP STA shall abandon this transition attempt.

The non-AP STA shall perform a reassociation directly with the target AP via the following exchange:

STA→Target AP: Reassociation Request(RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)

Target AP→STA: Reassociation Response(RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]], RIC-Response)

The SME of the STA initiates the reassociation through the use of the MLME-REASSOCIATE.request primitive. The SME of the AP responds to the indication with MLME-REASSOCIATE.response primitive. See 11.3.2.

In the Reassociation Request frame, the SA field of the message header shall be set to the MAC address of the STA, and the DA field of the message header shall be set to the BSSID of the target AP. The information elements in the frame, the information element contents, and the MIC calculation shall be as given in 11A.8.4.

The R1KH of the target AP verifies the MIC in the FTIE in the Reassociation Request frame and shall discard the request if the MIC is incorrect. If the contents of the MDIE received by the target AP do not

match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the Reassociation Request frame with status code 54 (i.e., Invalid MDIE). If the FTIE in the Reassociation Request frame contains a different R0KH-ID, R1KH-ID, ANonce, or SNonce, the AP shall reject the Reassociation Request frame with status code 55 (i.e., Invalid FTIE). If the RSNIE in the Reassociation Request frame contains an invalid PMKR1Name, the AP shall reject the Reassociation Request frame with status code 53 (i.e., Invalid PMKID).

In the Reassociation Response frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the non-AP STA. The Status Code field shall be a value from the options listed in 7.3.1.9. The information elements in the frame, the information element contents, and the MIC calculation shall be as given in 11A.8.5.

The S1KH of the non-AP STA verifies the MIC in the FTIE in the Reassociation Response frame and shall discard the response if the MIC is incorrect.

If the non-AP STA is performing a reassociation exchange as part of the FT Resource Request Protocol, then the non-AP STA shall not include the RIC-Request in the Reassociation Request frame, and the AP shall not include the RIC-Response in the Reassociation Response frame. If the reassociation exchange is part of the FT Resource Request Protocol and the AP is unable to honor the resources that have been placed in the accepted state for that non-AP STA, then the AP shall reject the Reassociation Request frame and may use status code 33 (i.e., Association denied because QoS AP has insufficient bandwidth to handle another QoS STA).

If the non-AP STA did not utilize the FT Resource Request Protocol, the STA may make a request for resources by including a RIC-Request (see 11A.11) in the Reassociation Request frame. The RIC-Request is generated by the procedures of 11A.11.3.1, and the RIC-Response is generated by the procedures of 11A.11.3.2.

If the Status Code field value returned by the target AP in the response is 1 (i.e., Unspecified failure), 14 (i.e., Authentication transaction sequence number out of sequence), or 16 (i.e., Authentication rejected due to timeout waiting for next frame in sequence), then the non-AP STA shall abandon this transition attempt. Handling of other errors returned in the Status Code field shall be as specified in 11.3.

Upon a successful reassociation, the PTKSA has been established and proven live. The SME of the AP shall open the IEEE 802.1X Controlled Port. The non-AP STA shall transition to State 3 (as defined in 11.3). If the target AP is distinct from the previous AP, the non-AP STA shall enter State 1 with respect to the previous AP.

Upon a successful reassociation, the non-AP STA shall delete any corresponding PTKSA with its previous AP. The SME of the STA shall issue an MLME-DELETEKEYS.request primitive to delete the pairwise keys with the previous AP, and the STA and the AP shall issue a MLME-SETKEYS.request primitive and MLME-SETPROTECTION.request primitive to install the pairwise keys. The PTK key lifetime timer shall be initialized with the value calculated as the difference between the TIE[KeyLifetime] sent in Message 3 of the FT initial mobility domain association and the time since the completion of the FT 4-Way Handshake during the FT initial mobility domain association.

When the IEEE 802.1X Controlled Port is opened, the EAPOL-Key frame replay counter shall be initialized to zero. The R1KH shall increment the key replay counter on each successive EAPOL-Key frame that it transmits.

11A.7.2 FT reassociation in a non-RSN

The STA shall perform a reassociation with the target AP via the following exchange:

STA→Target AP: Reassociation Request(MDIE, RIC-Request)

Target AP→STA: Reassociation Response(MDIE, RIC-Response)

The non-AP SME of the STA initiates the reassociation through the use of the MLME-REASSOCIATE.request primitive. The SME of the AP responds to the indication with MLME-REASSOCIATE.response primitive. See 11.3.2.

In the Reassociation Request frame, the SA field of the message header shall be set to the MAC address of the non-AP STA, and the DA field of the message header shall be set to the BSSID of the target AP. The information elements in Reassociation Request frame, and their required contents, shall be as given in 11A.8.4.

If the contents of the MDIE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the Reassociation Request frame with status code 54 (i.e., Invalid MDIE).

In the Reassociation Response frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the non-AP STA. The information elements in Reassociation Response frame, and their required contents, shall be as given in 11A.8.5. The Status Code field shall be a value from the options listed in 7.3.1.9.

If the STA is performing a reassociation exchange as part of the FT Resource Request Protocol, then the STA shall not include the RIC-Request in the Reassociation Request frame, and the AP shall not include the RIC-Response in the Reassociation Response frame.

If the non-AP STA did not utilize the FT Resource Request Protocol, the STA may make a request for resources by including a RIC-Request (see 11A.11) in the Reassociation Request frame. The RIC-Request is generated by the procedures of 11A.11.3.1, and the RIC-Response is generated by the procedures of 11A.11.3.2.

If the Status Code field value returned by the target AP in the response is 1 (i.e., Unspecified failure), 14 (i.e., Authentication transaction sequence number out of sequence), or 16 (i.e., Authentication rejected due to timeout waiting for next frame in sequence), then the non-AP STA shall abandon this transition attempt. Handling of other errors returned in the Status Code field shall be as specified in 11.3.

If the AP has dot11RSNAEnabled set to TRUE, upon a successful reassociation, the SME shall open the IEEE 802.1X Controlled Port.

Upon a successful reassociation, the target AP and the non-AP STA shall transition to State 3 (as defined in 11.3). If the target AP is distinct from the previous AP, then the non-AP STA shall enter State 1 with respect to the previous AP.

11A.8 FT authentication sequence

11A.8.1 Overview

The FT authentication sequence comprises four sets of FT information elements. Each set of FT information elements is referred to in 11A.8 as a *message*. These messages are included in the FT Protocol frames or FT Resource Request Protocol frames to initiate a fast BSS transition. The FT authentication sequence is always initiated by the non-AP STA and responded to by the target AP.

In an RSN, the first two messages in the sequence allow the non-AP STA and target AP to provide association instance identifiers, SNonce and ANonce, respectively. SNonce and ANonce are chosen

randomly or pseudo-randomly and are used to generate a fresh PTK. The first two messages also enable the target AP to provision the PMK-R1 and the non-AP STA and target AP to compute the PTK. The third and fourth messages demonstrate liveness of the peer, authenticate the information elements, and enable an authenticated resource request.

When a non-AP STA invokes the FT Protocol, then the first two messages of the sequence are both carried in Authentication frames or both carried in Action frames, and these messages are described in 11A.8.2 and 11A.8.3. The third and fourth messages in the sequence are carried in the Reassociation Request and Reassociation Response frames and are described in 11A.8.4 and 11A.8.5.

When the non-AP STA invokes the FT Resource Request Protocol, then the first four messages of the sequence are all carried in Authentication frames or all carried in Action frames, and these messages are described in 11A.8.2 through 11A.8.5. The fifth and sixth frames of the FT Resource Request Protocol are carried in the Reassociation Request frame and Reassociation Response frame and are described in 11A.8.4 and 11A.8.5.

Regardless of the transport mechanism, the information contained in the FT authentication sequence consists of the set of information elements shown in Table 11A-1:

Table 11A-1—FT authentication information elements

Information	Presence in Authentication Sequence messages	Description
RSN	Present in all messages of the sequence if dot11RSNAEnabled is set to TRUE.	7.3.2.25
Mobility domain	Present in all messages of the sequence.	7.3.2.47
Fast BSS transition	Present in all messages of the sequence if dot11RSNAEnabled is set to TRUE.	7.3.2.48
Timeout interval (reassociation deadline)	May optionally appear in the fourth message of the sequence if dot11RSNAEnabled is not set to TRUE.	7.3.2.49
RIC	May appear in the third and fourth messages.	7.3.2.50

The first message is used by the non-AP STA to initiate a fast BSS transition. When RSNA is enabled, the STA shall include the R0KH-ID and the SNonce in the FTIE and the PMKR0Name in the RSNIE. The target AP can use the PMKR0Name to derive the PMKR1Name, and if the target AP does not have the PMK-R1 identified by PMKR1Name, it may attempt to retrieve that key from the R0KH identified by R0KH-ID. See 11A.2. The non-AP STA includes a fresh SNonce as its contribution to the association instance identifier and to provide key separation of the derived PTK; it is selected randomly to serve as a challenge that will demonstrate the liveness of the peer in the fourth message.

The second message is used by the target AP to respond to the requesting non-AP STA. The target AP provides the key holder identifiers and key names used to generate the PTK. The target AP also includes a fresh ANonce as its contribution to the association instance identifier and to provide key separation of the derived PTK. The response includes a status code.

In an RSN, the third message is used by the non-AP STA to assert to the target AP that it has a valid PTK. If no resources are required, then the STA omits inclusion of the RIC.

The fourth message is used by the target AP to respond to the requesting non-AP STA. This message serves as final confirmation of the transition, establishes that the AP possesses the PMK-R1 and is participating in this association instance, and protects against downgrade attacks. Note, however, that the RIC will be absent if no resources were requested in the third message. This also includes a status code and may include a reassociation deadline.

11A.8.2 FT authentication sequence: contents of first message

The RSNIE shall be present only if dot11RSNAEnabled is set to TRUE. If present, the RSNIE shall be set as follows:

— Version field shall be set to 1.
— PMKID Count field shall be set to 1.
— PMKID List field shall contain the PMKR0Name.
— All other fields shall be as specified in 7.3.2.25 and 8.4.3.

The MDIE shall contain the MDID field and the FT Capability and Policy field settings obtained from the target AP, as advertised by the target AP in Beacon and Probe Response frames. The MDID shall be identical to that obtained during the FT initial mobility domain association exchange.

The FTIE shall be present only if dot11RSNAEnabled is set to TRUE. If present, the FTIE shall be set as follows:

— R0KH-ID shall be the value of R0KH-ID obtained by the non-AP STA during its FT initial mobility domain association exchange.
— SNonce shall be set to a value chosen randomly by the non-AP STA, following the recommendations of 8.5.7.
— All other fields shall be set to 0.

11A.8.3 FT authentication sequence: contents of second message

If the status code is zero, then the following rules apply.

The RSNIE shall be present only if dot11RSNAEnabled is set to TRUE. If present, the RSNIE shall be set as follows:

— Version field shall be set to 1.
— PMKID Count field shall be set to 1.
— PMKID List field shall be set to the value contained in the first message of this sequence.
— All other fields shall be identical to the contents of the RSNIE advertised by the AP in Beacon and Probe Response frames.

The MDIE shall contain the MDID and FT Capability and Policy fields. This information element shall be the same as the MDIE advertised by the target AP in Beacon and Probe Response frames.

The FTIE shall be present only if dot11RSNAEnabled is set to TRUE. If present, the FTIE shall be set as follows:

— R0KH-ID shall be identical to the R0KH-ID provided by the non-AP STA in the first message.
— R1KH-ID shall be set to the R1KH-ID of the target AP, from the MIB variable dot11FTR1KeyHolderID.

— ANonce shall be set to a value chosen randomly by the target AP, following the recommendations of 8.5.7.
— SNonce shall be set to the value contained in the first message of this sequence.
— All other fields shall be set to 0.

11A.8.4 FT authentication sequence: contents of third message

The RSNIE shall be present only if dot11RSNAEnabled is set to TRUE. If present, the RSNIE shall be set as follows:

— Version field shall be set to 1.
— PMKID Count field shall be set to 1.
— PMKID field shall contain the PMKR1Name.
— All other fields shall be as specified in 7.3.2.25 and 8.4.3.

The MDIE shall contain the MDID and FT Capability and Policy fields. This information element shall be identical to the MDIE contained in the first message of this sequence.

The FTIE shall be present only if dot11RSNAEnabled is set to TRUE. If present, the FTIE shall be set as follows:

— ANonce, SNonce, R0KH-ID, and R1KH-ID shall be set to the values contained in the second message of this sequence.
— The Information Element Count field of the MIC Control field shall be set to the number of information elements protected in this frame (variable).
— When the negotiated AKM is 00-0F-AC:3 or 00-0F-AC:4, the MIC shall be calculated using the KCK and the AES-128-CMAC algorithm. The output of the AES-128-CMAC shall be 128 bits.
— The MIC shall be calculated on the concatenation of the following data, in the order given here:
  — non-AP STA MAC address (6 octets)
  — Target AP MAC address (6 octets)
  — Transaction sequence number (1 octet), which shall be set to the value 5 if this is a Reassociation Request frame and, otherwise, set to the value 3.
  — Contents of the RSNIE.
  — Contents of the MDIE.
  — Contents of the FTIE, with the MIC field of the FTIE set to 0.
  — Contents of the RIC-Request (if present)
— All other fields shall be set to 0.

If resources are being requested by the STA, then a sequence of information elements forming the RIC-Request shall be included.

11A.8.5 FT authentication sequence: contents of fourth message

If the status code is zero, then the following rules apply.

The RSNIE shall be present only if dot11RSNAEnabled is set to TRUE. If present, the RSNIE shall be set as follows:

— Version field shall be set to 1.

- PMKID Count field shall be set to 1.
- PMKID field shall contain the PMKR1Name
- All other fields shall be identical to the contents of the RSNIE advertised by the target AP in Beacon and Probe Response frames.

The MDIE shall contain the MDID and FT Capability and Policy fields. This information element shall be identical to the MDIE contained in the second message of this sequence.

The FTIE shall be present only if dot11RSNAEnabled is set to TRUE. If present, the FTIE shall be set as follows:

- ANonce, SNonce, R0KH-ID, and R1KH-ID shall be set to the values contained in the second message of this sequence.
- The Information Element Count field of the MIC Control field shall be set to the number of information elements protected in this frame (variable).
- When this message of the authentication sequence appears in a Reassociation Response frame, the Optional Parameter(s) field in the FTIE may include a GTK subelement. If a GTK is included, the Key field of the subelement shall be encrypted using KEK and the NIST AES key wrap algorithm. The Key field shall be padded before encrypting if the key length is less than 16 octets or if it is not a multiple of 8. The padding consists of appending a single octet 0xdd followed by zero or more 0x00 octets. When processing a received message, the receiver shall ignore this trailing padding. Addition of padding does not change the value of the Key Length field. Note: The length of the encrypted Key field can be determined from the length of the GTK subelement.
- When the negotiated AKM is 00-0F-AC:3 or 00-0F-AC:4, the MIC shall be calculated using the KCK and the AES-128-CMAC algorithm. The output of the AES-128-CMAC algorithm shall be 128 bits.
- The MIC shall be calculated on the concatenation of the following data, in the order given here:
  - Non-AP STA MAC address (6 octets)
  - Target AP MAC address (6 octets)
  - Transaction sequence number (1 octet), which shall be set to the value 6 if this is a Reassociation Response frame or, otherwise, set to the value 4.
  - Contents of the RSNIE.
  - Contents of the MDIE.
  - Contents of the FTIE, with the MIC field of the FTIE set to 0.
  - Contents of the RIC-Response (if present)
- All other fields shall be set to 0.

If this message is other than a Reassociation Response frame and dot11RSNAEnabled is set to FALSE, a TIE may appear. If this message is other than a Reassociation Response frame, includes a RIC-Response, and dot11RSNAEnabled is set to FALSE, then a timeout interval shall appear. If it appears, it shall be set as follows:

- Timeout Interval Type field shall be set to 1 (reassociation deadline)
- Timeout Interval Value field shall be set to the reassociation deadline time.

If resources were requested by the non-AP STA, then a RIC-Response shall be included.

11A.9 FT security architecture state machines

11A.9.1 Introduction

The FT state machines describe the interaction between the RSNA key management and 802.11 architectural components.

RSNA key management uses the MA-UNITDATA service primitives to send/receive EAPOL-Key frames for FT initial association; MLME interfaces described below for FT methods; and MLME-SETKEYS, MLME-DELETEKEYS, and MLME-SETPROTECTION primitives. FT key management uses the following primitives for key management delivery and reception:

— The MLME-REMOTE_REQUEST primitives for FT key management over the DS
— The MLME-AUTHENTICATE primitives for FT key management over the air
— The MLME-RESOURCE_REQUEST primitives for FT resource request over the air
— The MLME-REASSOCIATE primitives for FT key management over the air and over the DS

Some of the state machine design considerations are as follows:

— Details of error handling are not included in the state machines. See 11A.4, 11A.5, and 11A.6.
— Retransmission of FT Authentication and (Re)Association frames are not included in the state machines, see 11A.5 and 11A.6.

Various signals are used to communicate between the R0KH and the R1KH state machines. Note that these interactions may be between separate entities, rather than within a single SME. These interactions are as follows:

— In the R0KH state machine, FT-PMKR1-SA (PMKR1-SA) sends a PMK-R1 PMKSA to the R1KH.
— In the R1KH state machine, FT-FULL-AUTH (R1KH-ID) requests a key from the R0KH.
— In the R1KH state machine, FT-Transition-Auth (R1KH-ID) requests a key from the R0KH that was used for the initial mobility domain association.

The interactions between the R0KH and IEEE 802.1X, between the R1KH and IEEE 802.1X, and between the S1KH and IEEE 802.1X occur within the SME. At both the target AP and at the non-AP STA, the R1KH and S1KH initialize the IEEE 802.1X EAPOL state machines in the respective SMEs. The Controlled Port is opened without an EAP exchange when the reassociation completes.

11A.9.2 R0KH state machine

There is one R0KH state machine, which includes FT key management.

The state diagram in Figure 11A-13 consists of a set of states that handle R0KH functions including key hierarchy instantiation, key generation, and cleanup. This state machine interacts with the R1KH state machine.

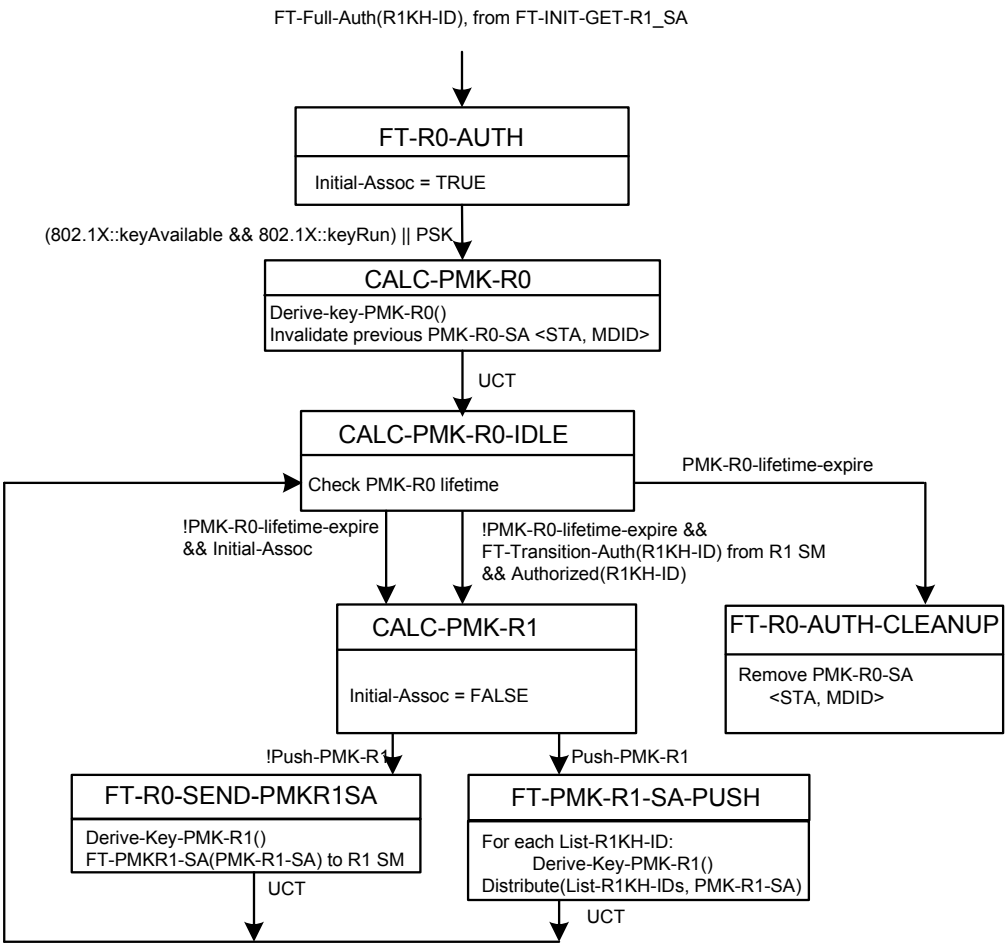

Figure 11A-13—R0KH state machine

11A.9.2.1 R0KH state machine states

The following list summarizes the states of the R0KH state machine:

— **CALC-PMK-R0**: This state is entered after the MSK from EAP authentication or PSK is available.

— **CALC-PMK-R0-IDLE**: This state is an intermediate state for the R0KH to wait for new requests from R1KHs.

— **CALC-PMK-R1**: For FT initial association, this state is entered as an unconditional transfer. For FT methods, this state is entered through the event from an R1KH state machine.

— **FT-PMK-R1-SA-PUSH**: This state is entered if Push-PMK-R1 is set to TRUE. PMK-R1s are derived and distributed to all the configured R1KHs.

— **FT-R0-AUTH-CLEANUP**: This state is entered when the PMK-R0 lifetime expires.

— **FT-R0-AUTH**: This state is entered through the event from the R1KH state machine. The R1KH state machine sends this event when it determines that a new PMK-R0 is needed.

— **FT-R0-SEND-PMKR1SA**: This state is entered from CALC-PMK-R1 when a request for the PMK-R1 security association is received from an R1KH. PMK-R1 security association is derived and distributed to the requesting R1KH.

11A.9.2.2 R0KH state machine variables

The following list summarizes the variables used by the R0KH state machine:

— *Initial-Assoc* – This variable is used to indicate whether the current authentication is the initial association, in order to trigger the initial derivation of PMK-R1.

— *List-R1KH-IDs* – This variable contains a list of all of the R1KH-IDs in the mobility domain. This list is populated by the key distribution protocol as required in 11A.2.

— *PMK-R0-lifetime-expire* – This variable is set to TRUE when PMK-R0 lifetime is deemed expired.

— *PSK* – This variable is set to TRUE when authentication is performed by use of a preshared key.

— *Push-PMK-R1* – This variable is set to TRUE when R0KH can push the PMK-R1 security associations to R1KHs.

11A.9.2.3 R0KH state machine procedures

The following list summarizes the procedures used by the R0KH state machine:

— **Authorized(R1KH-ID)** – This procedure returns a value of true if the R1KH is a known key holder in the mobility domain.

— **Distribute (List-R1KH-IDs, PMK-R1 PMKSA)** – Distributes the PMK-R1-SAs for the current instance of the key hierarchy to the list of R1KH-IDs.

— **Derive-Key-PMK-R0()** – This procedure derives the PMK-R0 from the MSK or PSK, derives the PMKR0Name (as described in 8.5.1.5.3), and creates PMK-R0 security association.

— **Derive-Key-PMK-R1 (R1KH-ID)** – This procedure derives the PMK-R1 from PMK-R0, and R1KH-ID (as described in 8.5.1.5.4), for the R1KH identified by R1KH-ID, and creates PMK-R1 security association.

11A.9.3 R1KH state machine

The R1KH state machine includes functions for FT initial association and FT protocols. The R1KH states performing FT initial association and the R1KH states performing FT protocol exchanges interact differently with the R0KH state machine.

The R1KH state machine and other portions of the SME are defined in Figure 11A-14 and Figure 11A-15 and consist of a set of states that handle FT initial mobility domain association, PMK-R1 reception, PTK handshake and session establishment, FT protocols (including resource requests), and cleanup. This state machine interacts with the R0KH state machine to generate a fresh FT key hierarchy for the initial mobility domain association and to get the PMK-R1 security association (PMK-R1 PMKSA) for the FT protocols. While the figures show the over-the-air message exchanges, the over-the-DS exchanges are handled similarly.

A new instance of the R1KH state machine is created each time initial mobility domain association or fast BSS transition is initiated.

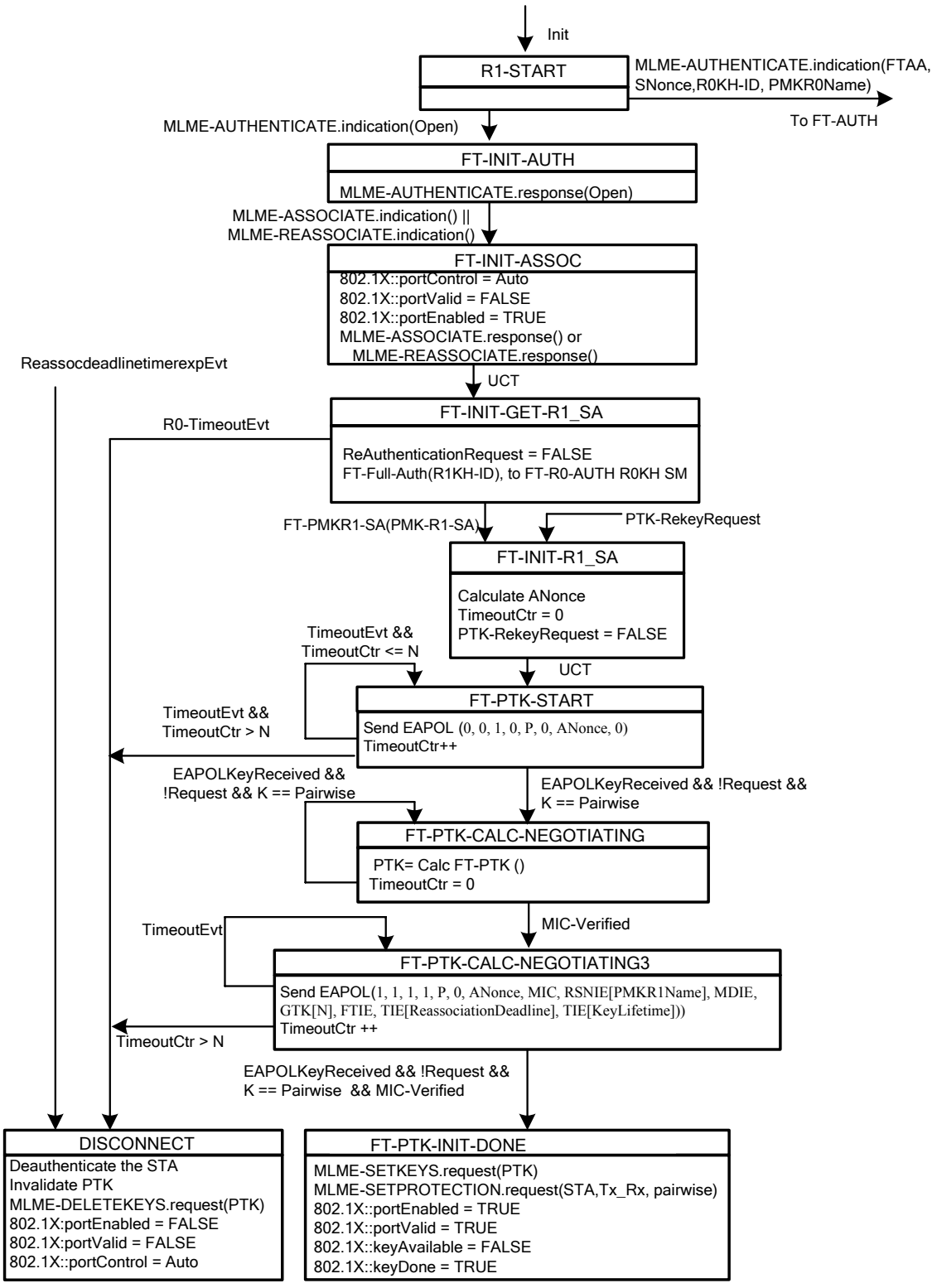

Figure 11A-14—R1KH state machine, including portions of the SME (part 1)

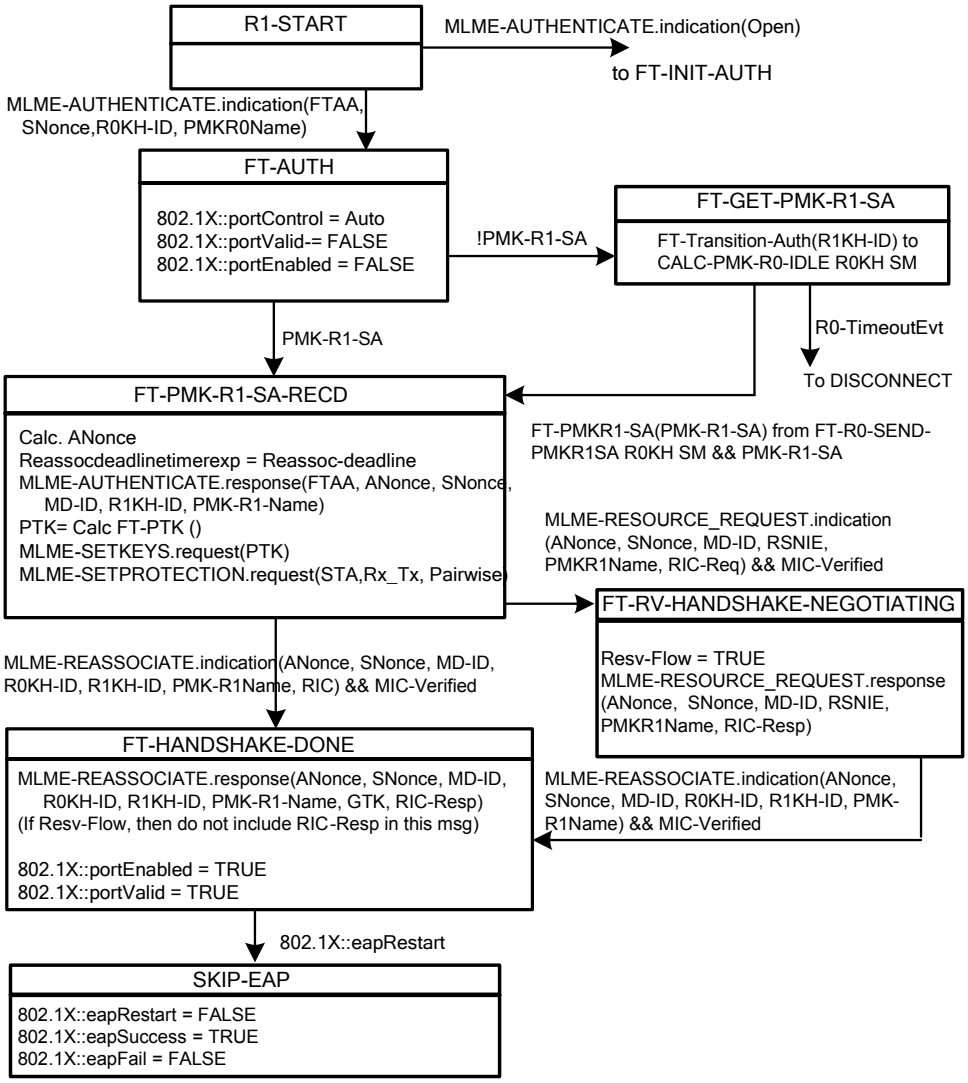

Figure 11A-15—R1KH state machine, including portions of the SME (part 2)

11A.9.3.1 R1KH state machine states

The following list summarizes the states of the R1KH state machine:

— **DISCONNECT**: This state is entered when the current session ends or when errors occur.

— **FT-AUTH**: This state is entered upon receipt of an indication that an FT Protocol or FT Resource Request Protocol is invoked.

— **FT-GET-PMK-R1-SA**: This state is entered when R1KH sends a message to the R0KH to get the PMK-R1-SA.

— **FT-HANDSHAKE-DONE**: This state is entered when reassociation indication parameters are validated. The Reassociation Response frame is then sent.

— **FT-INIT-ASSOC**: This state is entered upon receipt of a (Re)Association Request frame during initial mobility domain association.

— **FT-INIT-AUTH**: This state is entered upon receipt of an indication that initial association is invoked.
— **FT-INIT-GET-R1_SA**: This state is entered when the R1KH determines that a new key hierarchy is required.
— **FT-INIT-R1_SA**: This state is entered on receiving the PMK-R1-SA from the R0KH and when rekeying the PTK.
— **FT-PMK-R1-SA-RECD**: This state is entered on receiving the PMK-R1-SA from the R0KH. An FT Authenticate response is sent in this state. This state then calculates the PTK and delivers the key to the MAC.
— **FT-PTK-INIT-DONE**: This state is entered on successful validation of the fourth EAPOL-Key message. In this state, keys are provided to the MAC.
— **FT-PTK-CALC-NEGOTIATING**: This state is entered when a second EAPOL-Key message is received.
— **FT-PTK-CALC-NEGOTIATING3**: This state is entered on successful validation of the second EAPOL-Key message. In this state, the third EAPOL-Key message is sent.
— **FT-PTK-START**: This state is entered when the PMK-R1-SA is present. This state is the beginning of the 4-Way Handshake to derive a fresh PTK.
— **FT-RV-HANDSHAKE-NEGOTIATING**: This state is entered when an FT resource request is received. The FT resource response is sent.
— **R1-START**: This is the start of the R1KH state machine.
— **SKIP-EAP**: This state is entered after successful completion of the FT Protocol. In this state, the EAPOL state machine is triggered to open the IEEE 802.1X port.

11A.9.3.2 R1KH state machine variables

The following list summarizes the variables used by the R1KH state machine:

— *Init* – This variable is set to TRUE to initialize the R1KH the state machine
— *EAPOLKeyReceived* – This variable is set to TRUE when an EAPOL-Key message is received.
— *K* – This variable is one of the values of the Key Type bit in the EAPOL-Key frame received and can be either Pairwise or Group
— *MIC-Verified* – This variable is set to TRUE when the message authentication code integrity check passes.
— *Pairwise* – This variable is one of the values of the Key Type bit in the EAPOL-Key frame
— *PMK-R1-SA* – This variable is set to TRUE when a valid PMK-R1-SA is present at the R1KH.
— *PTK-RekeyRequest* – This variable is set to TRUE when a PTK Key request is received.
— *Reassocdeadlinetimerexp* – This variable contains the reassociation deadline timer value.
— *ReassocdeadlinetimerexpEvt* – This variable is set to TRUE when the reassociation deadline timer expires.
— *Request* – This variable is the value of the Request bit in the Key Information field in the EAPOL-Key frame.
— *Resv-flow* – This variable is set to TRUE when an indication of an FT Resource Request Protocol is received.
— *R0-TimeoutEvt* – This variable is set to TRUE when the timeout for R0KH authentication expires (e.g., when the EAP authentication session timeout expires).
— *TimeoutCtr* – This variable contains the number of successive timeouts waiting for protocol responses.
— *TimeoutEvt* – This variable is set to TRUE when a timeout for receiving EAPOL-Key response expires.

11A.9.3.3 R1KH state machine procedures

The following list summarizes the procedure used by the R1KH state machine:

— **Calc-FT-PTK()** – This procedure calculates the PTK.

11A.9.4 S0KH state machine

There is one S0KH state machine within the Supplicant, defined in Figure 11A-16, which incorporates the FT initial association and key management.

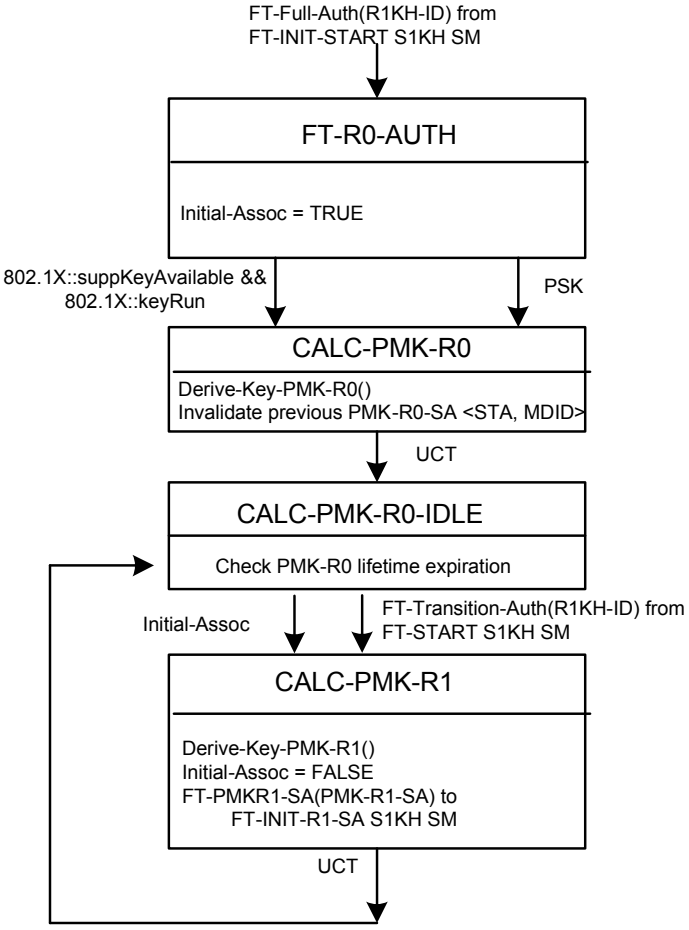

Figure 11A-16—S0KH state machine

11A.9.4.1 S0KH state machine states

The following list summarizes the states of the S0KH state machine:

— **CALC-PMK-R0**: This state is entered after the key is received, either from the EAP authentication or from the PSK.

— **CALC-PMK-R0-IDLE**: This state is entered after the PMK-R0 has been calculated and either continues with initial association or waits for requests from an S1KH for a PMK-R1.

— **CALC-PMK-R1**: For FT initial association, this state is entered as an unconditional transfer. For FT protocols, this state is entered through the event from the S1KH state machine. In this state, the PMK-R1 is sent to the S1KH.

— **FT-R0-AUTH**: This state is entered when the FT-Full-Auth event occurs during initial association in the S1KH state machine. The S1KH state machine sends this event when it determines that a new PMK-R0 is needed.

11A.9.4.2 S0KH state machine variables

The following list summarizes the variables used by the S0KH state machine:

— *Initial-Assoc* – This variable is used to indicate whether the current authentication is the initial association, in order to trigger the initial derivation of PMK-R1.

— *PSK* – This variable is set to TRUE when authentication is performed by use of a preshared key.

11A.9.4.3 S0KH state machine procedures

The following list summarizes the procedures used by the S0KH state machine:

— **Derive-Key-PMK-R0()** – This procedure derives the PMK-R0 and PMKR0Name and creates PMK-R0 security association.

— **Derive-Key-PMK-R1 (R1KH-ID)** – This procedure derives the PMK-R1 and PMKR1Name from PMK-R0 for the indicated R1KH and creates PMK-R1 security association.

11A.9.5 S1KH state machine

The S1KH state machine includes functions for fast BSS transitions, including initial association. The S1KH state machine and other portions of the SME are defined in Figure 11A-17 and Figure 11A-18 and consist of a set of states that handle FT initial association, PTK handshake and session establishment, resource requests, cleanup, and teardown. This state machine interacts with the S0KH state machine to generate a fresh key hierarchy.

11A.9.5.1 S1KH state machine states

The following list summarizes the states of the S1KH state machine:

— **DISCONNECT**: This state is entered when the current session expires.

— **FT-AIR-REQUEST**: This state is entered when it is determined that an over-the-air FT method will be executed. This state sends the FT Authentication Request frame over the air.

— **FT-DONE**: This state is entered when a Reassociation Response frame is received.

— **FT-DS-REQUEST**: This state is entered when it is determined that an over-the-DS FT method will be executed. This state sends the FT Authentication Request frame over the DS.

— **FT-INIT**: This state is entered when an FT method is initiated.

— **FT-INIT-ASSOC:** This state is entered when authentication for initial mobility domain association has been completed.

— **FT-INIT-AUTH**: This state is entered when an FT initial association event is initiated.

— **FT-INIT-START:** This state is entered when association for initial mobility domain association has been completed.

— **FT-INIT-R1-SA**: This state is entered on receiving the PMK-R1-SA from the S0KH and when the Authenticator is starting PTK rekeying by sending out EAPOL-Key Message 1.

— **FT-NO-RV-CONFIRM**: This state is entered when performing FT Protocol (i.e., not the FT Resource Request Protocol). This state is entered for both over-the-air and over-the-DS processing. This state sends the Reassociation Request frame.

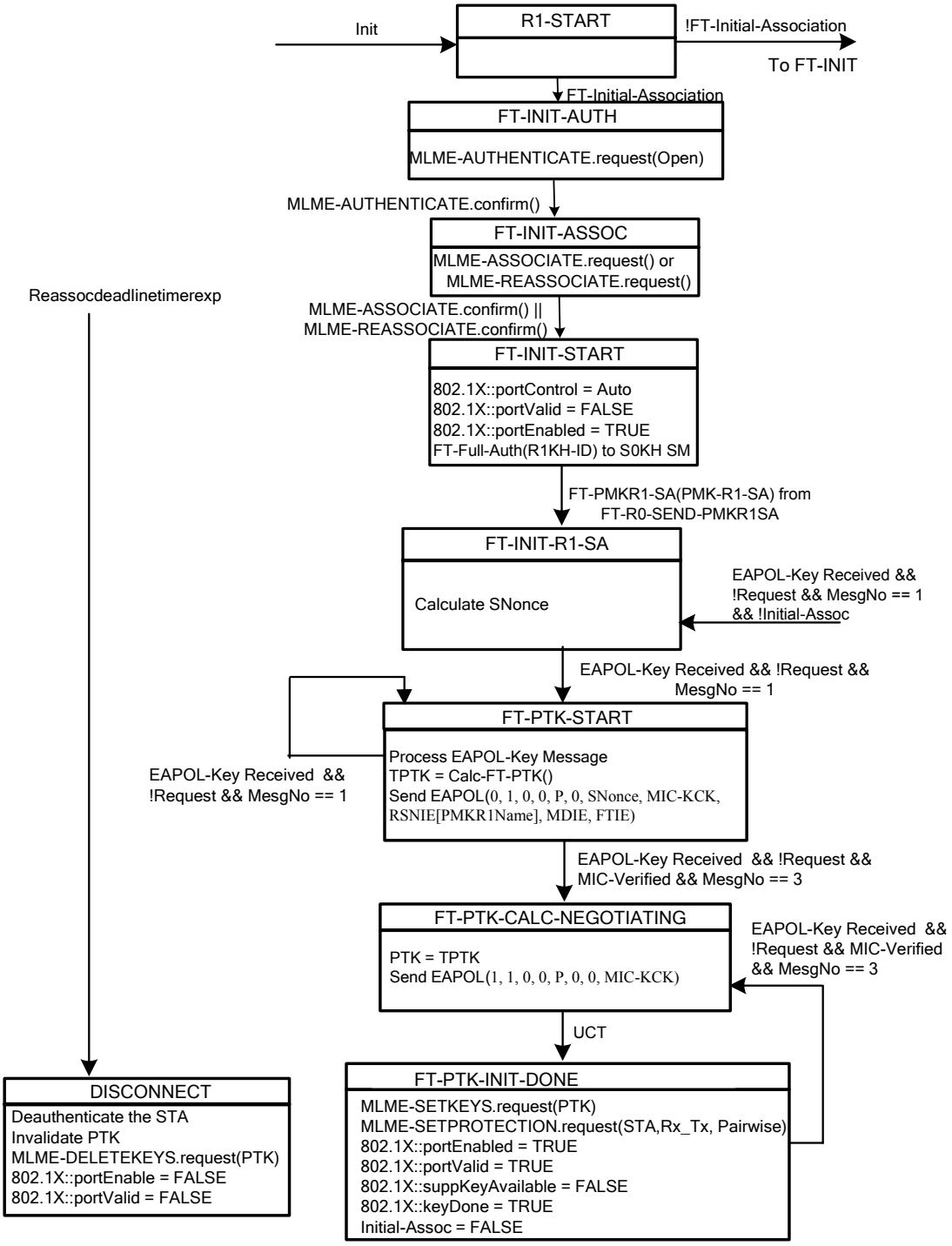

Figure 11A-17—S1KH state machine, including portions of the SME (part 1)

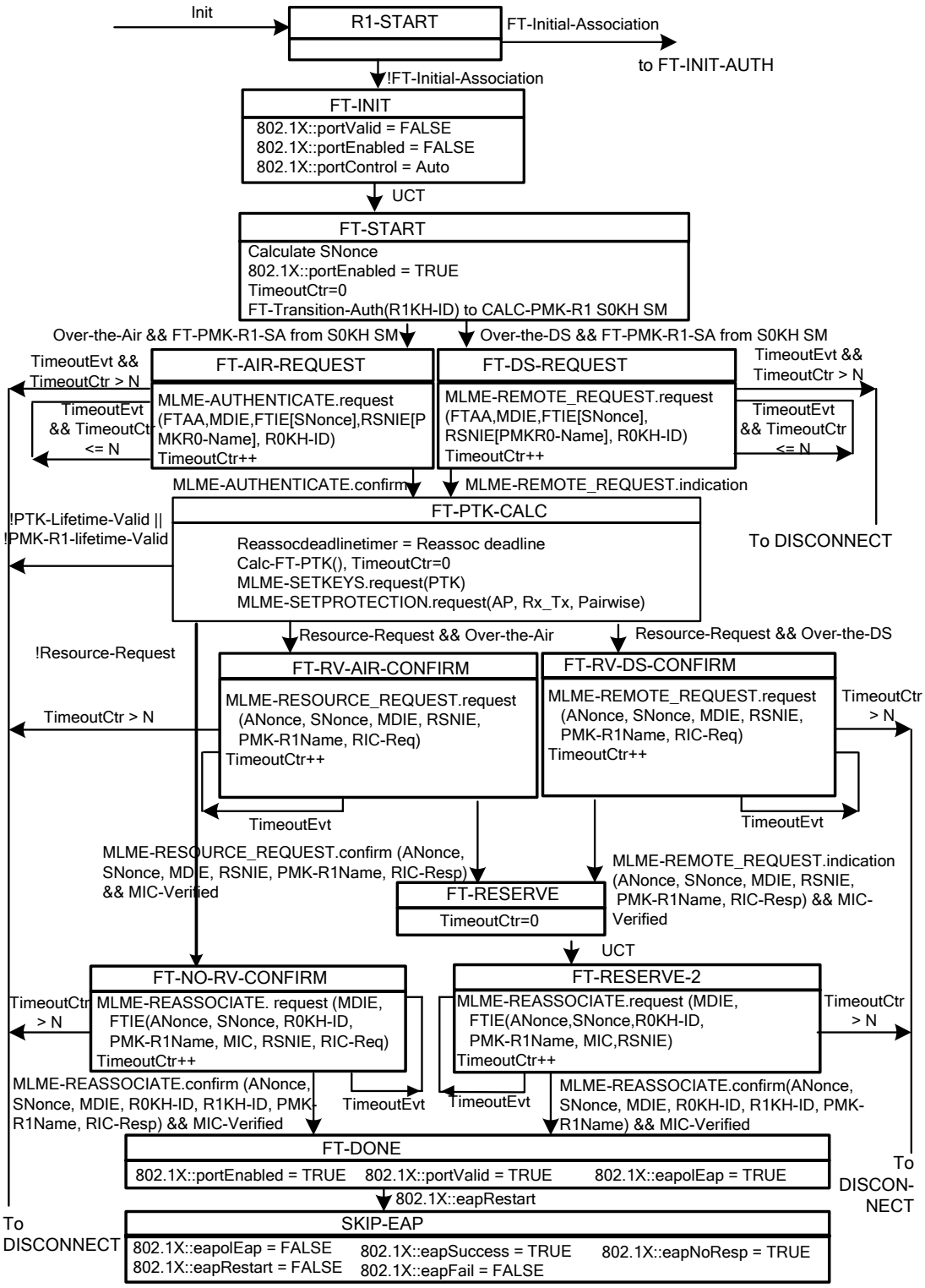

Figure 11A-18—S1KH state machine, including portions of the SME (part 2)

— **FT-PTK-CALC**: This state is entered when the over-the-air FT Authentication Response frame is received if an over-the-air FT Authentication Request frame was sent or when over-the-DS FT Response frame is received if an over-the-DS FT Request frame was sent. The PTK is calculated and installed in the MAC.

— **FT-PTK-INIT-DONE**: This state is entered after sending the fourth EAPOL-Key message. This state establishes the PTK keys into the MAC.

— **FT-PTK-CALC-NEGOTIATING**: This state is entered when a valid, third EAPOL-Key message is received. This state sends the fourth EAPOL-Key message.

— **FT-PTK-START**: This state is entered to derive a new PTK when the PMK-R1-SA is present and when the EAPOL-Key 4-Way Handshake Message 1 is received. This state sends the EAPOL-Key 4-Way Handshake Message 2.

— **FT-RESERVE**: This state is entered when the over-the-air FT Authentication Ack frame is received if an over-the-air FT Authentication Confirm frame was sent or when over-the-DS FT Ack frame is received if an over-the-DS FT Confirm frame was sent.

— **FT-RESERVE-2**: The Reassociation Request frame is sent in this state after completion of FT resource request.

— **FT-RV-AIR-CONFIRM**: This state is entered for over-the-air FT Resource Request Protocol processing. The FT Authentication Confirm frame containing the FT resource request is sent.

— **FT-RV-DS-CONFIRM**: This state is entered for over-the-DS FT Resource Request Protocol processing. The FT Authentication Confirm frame containing the FT resource request is sent.

— **FT-START**: This state is entered when all FT parameters are validated and FT needs to be initiated.

— **R1-START**: This is the start of the S1KH state machine.

— **SKIP-EAP**: This state is entered after successful completion of the FT Protocol. In this state, the EAPOL state machine is triggered to open the IEEE 802.1X port.

11A.9.5.2 S1KH state machine variables

The following list summarizes the variables used by the S1KH state machine:

— *EAPOLKeyReceived* – This variable is set to TRUE when an EAPOL-Key message is received.

— *FT-Initial-Association* – This variable is set to TRUE when the S1KH is performing an initial association.

— *Init* – This variable is set to TRUE to initialize the S1KH state machine. In addition, this variable can be used to restart the state machine when transitioning to a new AP.

— *MesgNo* – In conjunction with *EAPOLKeyReceived*, this variable indicates which message in the 4-Way Handshake has been received.

— *MIC-Verified* – This variable is set to TRUE when the message authentication integrity check is valid.

— *N* – This variable contains the limit of timeout events before considering the transition a failure.

— *Over-the-Air* – This variable is set to TRUE when the FT Protocol will be exchanged over the air. Note that both *Over-the-Air* and *Over-the-DS* cannot be set to TRUE at the same time.

— *Over-the-DS* – This variable is set to TRUE when the FT Protocol will be exchanged over the DS. Note that both *Over-the-Air* or *Over-the-DS* cannot be set to TRUE at the same time.

— *PMK-R1-Lifetime-Valid* – This variable is set to TRUE when the PMK-R1 lifetime is valid.

— *PTK-Lifetime-Valid* – This variable is set to TRUE when the PTK lifetime is valid.

— *Reassocdeadlinetimer* – This variable contains the reassociation deadline timer value.

— *ReassocdeadlinetimerExp* – This variable is set to TRUE when the reassociation deadline timer expires.

— *Resource-request* – This variable is set to TRUE when the FT Resource Request Protocol will be executed.

— *TimeoutCtr* – This variable contains the number of successive timeouts waiting for protocol responses.

— *TimeoutEvt* – This variable is set to TRUE when a timeout event occurs.

— *TPTK* – This variable contains the newly calculated PTK, which will be installed after receipt of Message 3 of the 4-Way Handshake.

11A.9.5.3 S1KH state machine procedures

The following list summarizes the procedure used by the S1KH state machine:

— **Calc-FT-PTK()** – This procedure calculates the PTK.

11A.10 Remote request broker (RRB) communication

11A.10.1 Overview

The RRB mechanism allows the non-AP STA to communicate with a target AP through the non-AP STA's existing association (with the current AP). The non-AP STA transmits an FT Action frame (including the address of the STA and the BSSID of the target AP) to the current AP. The current AP encapsulates the FT Action frame (Request or Confirm) inside a Remote Request frame and transmits it to the target AP over the DS. The target AP processes the remote request and responds to the non-AP STA by sending an FT Action frame (Response or Acknowledgment) through the current AP.

The SME of the non-AP STA initiates an exchange with a target AP by issuing an MLME-REMOTE_REQUEST.request primitive with parameters including the contents of the FT Action frame to be sent. The MAC of the non-AP STA transmits this Action frame and issues an MLME-REMOTE_REQUEST.confirm primitive to signal that it has been sent. When the MAC of the current AP receives an FT Action frame, it passes it to the RRB by use of an MLME-REMOTE_REQUEST.indication primitive, with parameters including the contents of the received Action frame.

When the RRB of the current AP has received a response from the target AP, it uses the MLME-REMOTE_REQUEST.request primitive to send the response, as an FT Action frame, to the requesting non-AP STA. The MAC of the current AP transmits this Action frame and issues a MLME-REMOTE_REQUEST.confirm primitive to signal that it has been sent. When the MAC of the non-AP STA receives an FT Action frame, the MAC passes the Action frame to the SME by use of an MLME-REMOTE_REQUEST.indication primitive, with parameters including the contents of the received Action frame.

11A.10.2 Remote request broker (RRB)

The RRB resides in the SME on the APs and acts as a forwarding agent (at the current AP) and termination point (at the target AP) for protocol messages over the DS.

The RRB allows APs that are part of the same mobility domain to exchange information over the DS. APs that advertise the same MDID shall be reachable over the DS and support the over-the-DS communication.

As a termination point, when the RRB at the target AP receives a request frame from the current AP, it will interact with the MAC and other parts of the SME to process the request and respond with a Remote Response frame, through the RRB on the current AP, back to the requesting non-AP STA.

As a forwarding agent, when the RRB at the current AP receives a request from a non-AP STA directed to another AP in the same mobility domain, the current AP will forward the request to that target AP. The RRB on the current AP converts Action frames into Remote Request frames and converts Remote Response frames into Action frames.

The target AP and the current AP need to reside in the same mobility domain to successfully exchange Remote Request frames. The RRB on the current AP shall transmit Remote Request frames to the target AP based on the BSSID of the target AP (supplied in the FT Action frames) using the same procedures as preauthentication, as described in 8.4.6.1.

The message flow for a resource request over the DS is given in Figure 11A-19. The non-AP STA indicates the destination target AP BSSID as part of the FT Action frame. The RRB on the current AP encapsulates the FT Action frame and supplies the current AP BSSID in the Remote Request frame.

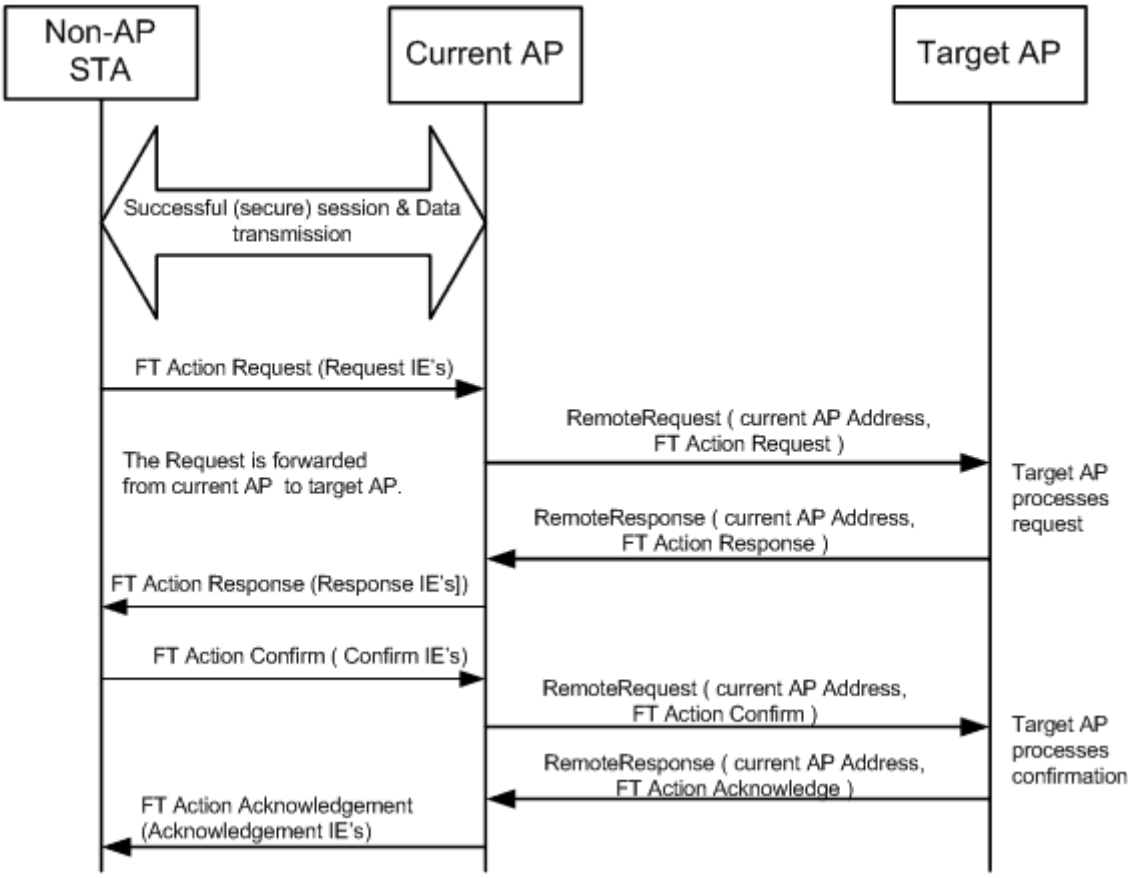

Figure 11A-19—Sample message flow for over-the-DS resource request

11A.10.3 Remote Request/Response frame definition

This subclause defines a mechanism to transport the remote request and remote response between the current AP and the target AP. Any other mechanism may be used.

The Remote Request frame is transmitted over the DS from the current AP to the target AP. The frame format for the Remote Request/Response frame is given in Figure 11A-20. Remote Request/Response

frames will use an Ethertype of 89-0d. The Remote Request/Response frame contains version, type, and length fields, along with the AP Address.

Size	Information
1	Remote Frame Type
1	FT Packet Type
2	FT Action Length
6	AP Address
Variable	FT Action Frame

Figure 11A-20—Remote Request/Response frame format

The Remote Frame Type field for FT Remote Request/Response frame shall be set to 1. Received messages with remote frame type other than 1 shall be discarded.

The FT Packet Type field shall be set to 0 for remote request and to 1 for remote response.

The FT Action Length field shall be set to an unsigned number representing the length in octets of the FT Action Frame field, following the bit ordering conventions of 7.1.1.

The AP Address field shall be set to the BSSID of the current AP. The target AP shall use this address as the destination address when sending the Remote Response frame as a response to the Remote Request.

The FT Action Frame field shall be set to the contents of the FT Action frame, from the Category field to the end of the Action frame body.

11A.11 Resource request procedures

11A.11.1 General

When using the resource request procedure, the non-AP STA has the option to request a resource allocation at the target AP. To request resources, the non-AP STA creates a resource information container (RIC) and inserts it in an appropriate request message to the target AP. The request message is sent to the target AP either directly (over the air), or via the current AP (over the DS), according to the FT procedures described in 11A.5 and 11A.6. In an RSNA, resource requests and responses are exchanged only after the establishment of the PTK and are protected by MICs.

The RIC contains a complete list of resources requested by the non-AP STA. An AP that receives a resource request from a non-AP STA shall discard any previous resource request from that STA. In an RSN, this resource request shall first be authenticated by the AP through checking of the MIC before the AP discards any previous resource request.

If the non-AP STA is performing a fast BSS transition according to the FT Protocol, described in 11A.5, it shall generate a RIC and process the RIC-Response according to the procedures of 11A.11.3.1, performing the exchange in the Reassociation Request/Response frames.

If the non-AP STA is performing a fast BSS transition according to the FT Resource Request Protocol, described in 11A.6, it shall generate a RIC and process the RIC-Response according to the procedures of

11A.11.3.1, performing the exchange in the Authentication Confirm/Authentication Ack frames (over the air) or FT Confirm/FT Ack frames (over the DS).

11A.11.2 Resource information container (RIC)

The RIC refers to a collection of information elements that are used to express a resource request or response.

When used in making a request, a RIC has one or more Resource Requests, as shown in Figure 11A-21.

Figure 11A-21—RIC-Request format

Each Resource Request consists of an RDIE followed by one or more alternative Resource Descriptors. An example of a Resource Request is shown in Figure 11A-22.

Figure 11A-22—Resource Request format

Each Resource Descriptor consists of one or more information elements. The possible Resource Descriptors that may appear in a RIC, and the information elements that they contain, are given in Table 11A-2.

Table 11A-2—Resource types and resource descriptor definitions

Resource type	Resource Descriptor definition	Notes
802.11 QoS	In a request: TSPEC (see 7.3.2.30), followed by zero or more TCLAS (see 7.3.2.31), followed by zero or one TCLAS Processing (See 7.3.2.33) In a response: TSPEC (see 7.3.2.30), followed by zero or one Schedule (See 7.3.2.34)	May be sent by a QoS non-AP STA to a QoS AP. Definition of TSPEC information elements shall be as given in 11.4. Definition of TCLAS, TCLAS Processing, and Schedule information elements, and the rules for including them in requests and responses, shall be as given in 11.4. Resource request procedures shall be as given in 11.4.
Block Ack Parameters	In a request: RIC Descriptor (see 7.3.2.51), containing a Resource Type field identifying Block Ack. In a response: RIC Descriptor (see 7.3.2.51), containing a Resource Type field identifying Block Ack.	Resource request procedures shall be as given in 11.5.
Vendor Specific	RDIE is followed by any vendor-specific information elements required to specify this resource	

If there are multiple Resource Descriptors, then they are treated as choices by the target AP. The AP attempts to allocate whatever is specified in the first Resource Descriptor; if this fails, the AP attempts to allocate whatever is specified in the next Resource Descriptor instead, and so on until a successful allocation or the AP reaches the end of the Resource Descriptor list. Thus, an OR relationship exists between Resource Descriptors that follow an RDIE, with the Resource Descriptors appearing in order of preference.

An example of a Resource Request consisting of two alternative Resource Descriptors is shown in Figure 11A-23.

Figure 11A-23—Resource Request example #1

For example, when the resource being requested is QoS for downstream traffic, a TSPEC information element may be followed by one or more TCLAS information elements and, when multiple TCLAS information elements are present, a TCLAS Processing element. Such an example Resource Request with two alternative TSPECs is shown in Figure 11A-24.

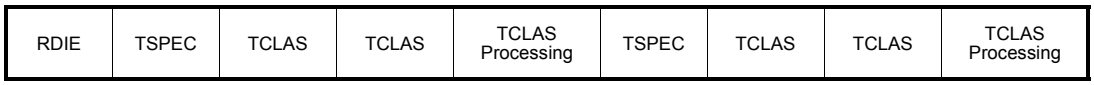

Figure 11A-24—Resource Request example #2

An example of a RIC with two resource requests, each with a single TSPEC, is given in Figure 11A-25.

Figure 11A-25—RIC-Request example #1

An example of a RIC with one resource request, with a choice of two TSPECs, is given in Figure 11A-26. This indicates that the target AP can select one of the two TSPECs.

Figure 11A-26—RIC-Request example #2

An example of a RIC with a RIC Descriptor is given in Figure 11A-27. The target AP can acknowledge if the resource specified in the RIC Descriptor is available.

Figure 11A-27—RIC-Request example #3

When sent by an AP in response to a RIC-Request, the RIC-Response consists of a list of one or more Resource Responses including one response for each of the Resource Requests that was contained in the RIC-Request. The basic format of a RIC-Response is shown in Figure 11A-28.

Figure 11A-28—RIC-Response format

Each Resource Response consists of an RDIE with the RDIE identifier matching the RDIE identifier in the request, in the same order as the RDIEs appeared in the request. The RDIE is followed by zero or one Resource Descriptors. If the request was not successful (as indicated in the RDIE status), then the AP may include a suggestion that could have been successful. If the resource request was successful, then the particular Resource Descriptor (of the alternatives given by the STA) is included in the response, as modified by the AP during the processing of the resource request. For example, when the resource being requested is QoS for upstream traffic, the TSPEC information element may be followed by a Schedule element.

An example of a RIC-Response with two QoS resource responses, each with a single TSPEC and Schedule element, is given in Figure 11A-29.

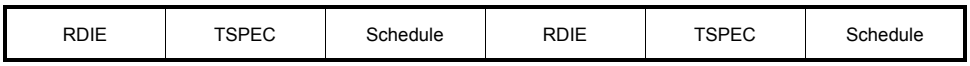

Figure 11A-29—Example QoS RIC-Response

11A.11.3 Creation and handling of a resource request

11A.11.3.1 STA procedures

The resource request enables a non-AP STA to request resources based on specified Resource Descriptors (e.g., TSPECs) before or at the time the non-AP STA associates with the target AP. In using TSPECs for requesting QoS resources, the TSPECs in the request need not belong to only active TSs; the non-AP STA can send TSPECs for any TS that it intends to use after the transition and request the same resources that would be requested by a later ADDTS exchange. For each resource, the STA may provide the AP with a choice of Resource Descriptors in order of preference, any one of which will meet the needs of the application.

The non-AP STA shall construct the RIC with a number of Resource Requests, each delineated by an RDIE.

The STA shall indicate the resources required at the target AP. For QoS resources, each TS shall be requested by a separate RDIE and associated TSPEC(s). The RDIE Identifier field in the RDIE shall be an arbitrary value chosen by the non-AP STA that uniquely identifies the RDIE within the RIC. The Status Code field shall be set to 0, and the Resource Count field shall be set to the number of alternative Resource Descriptors that follow.

Following each RDIE, the non-AP STA shall include one or more Resource Descriptors that define the resources required for this TS. When multiple TSPECs follow an RDIE as part of a single QoS resource request, a logical "OR" relationship exists between them, and at most one of these TSPECs will be accepted by the AP. The non-AP STA shall order the Resource Descriptors in decreasing order of preference.

In generating the RDIE for QoS resources for a TS, the procedures of 11.4 shall be followed for the generation of TSPECs and inclusion of TCLAS and TCLAS Processing elements. If the TS is a downstream flow, then the RDIE may also include one or more TCLAS element(s) (defined in 7.3.2.31) and (if multiple TCLAS elements are included) a TCLAS Processing element (defined in 7.3.2.33). If present, the TCLAS shall appear after the corresponding TSPEC.

A resource request is considered successful by a non-AP STA if the status code 0 is returned in each RDIE.

If the frame containing the response to the resource request contains a status code other than 0, the non-AP STA considers that the request has failed and that no resources are being held at the target AP.

The response from the target AP contains a RIC-Response, with the RDIEs in the response indicating which resources were considered by the target AP and the setting of the status code indicating which Resource Descriptors were accepted by the AP.

The RDIE Identifier field in the RDIE enables the STA to match the response with the RDIE in the request. The value of the Status Code field is interpreted as follows:

— Status code = 0 indicates that the request has been accepted. The RDIE may be followed by the Resource Descriptor that was accepted.

— Status code = nonzero (one of the values from 7.3.1.9) indicates that the resources could not be accepted. The RDIE may be followed by a suggested Resource Descriptor that could have been accepted.

A response to a successful resource request (other than in a Reassociation Request frame) may contain a reassociation deadline. If the non-AP STA does not initiate a Reassociation Request frame with the target AP within the reassociation deadline (if appropriate), then the AP will release resources held for that non-AP STA.

11A.11.3.2 AP procedures

When a RIC appears in a request message, the AP shall check its ability to allocate one resource for each RDIE in the RIC in the order appearing in the RIC. In a Reassociation Request frame, the QoS Capability information element shall be processed prior to the QoS resource requests in the RIC.

The behavior of the AP will be identical to that described in the flowchart in Figure 11A-30.

As shown in Figure 11A-30, the Resource Descriptors are examined by the AP in the order presented, and the first that could have been allocated is accepted. Thus the preference ordering by the non-AP STA is honored.

The target AP's SME examines the resource requests in the RIC. For requests that require processing by the MAC sublayer, the SME generates an MLME-RESOURCE_REQUEST_LOCAL.request primitive. The MAC shall respond with MLME-RESOURCE_REQUEST_LOCAL.confirm primitive that will indicate whether the MAC has accepted the resource request. The SME may also send these resource requests to an external entity such as a back-end QoS module for its consideration; these procedures are beyond the scope of this standard. The acceptance of a TSPEC by the target AP results in the resource allocation for a TS at the target AP.

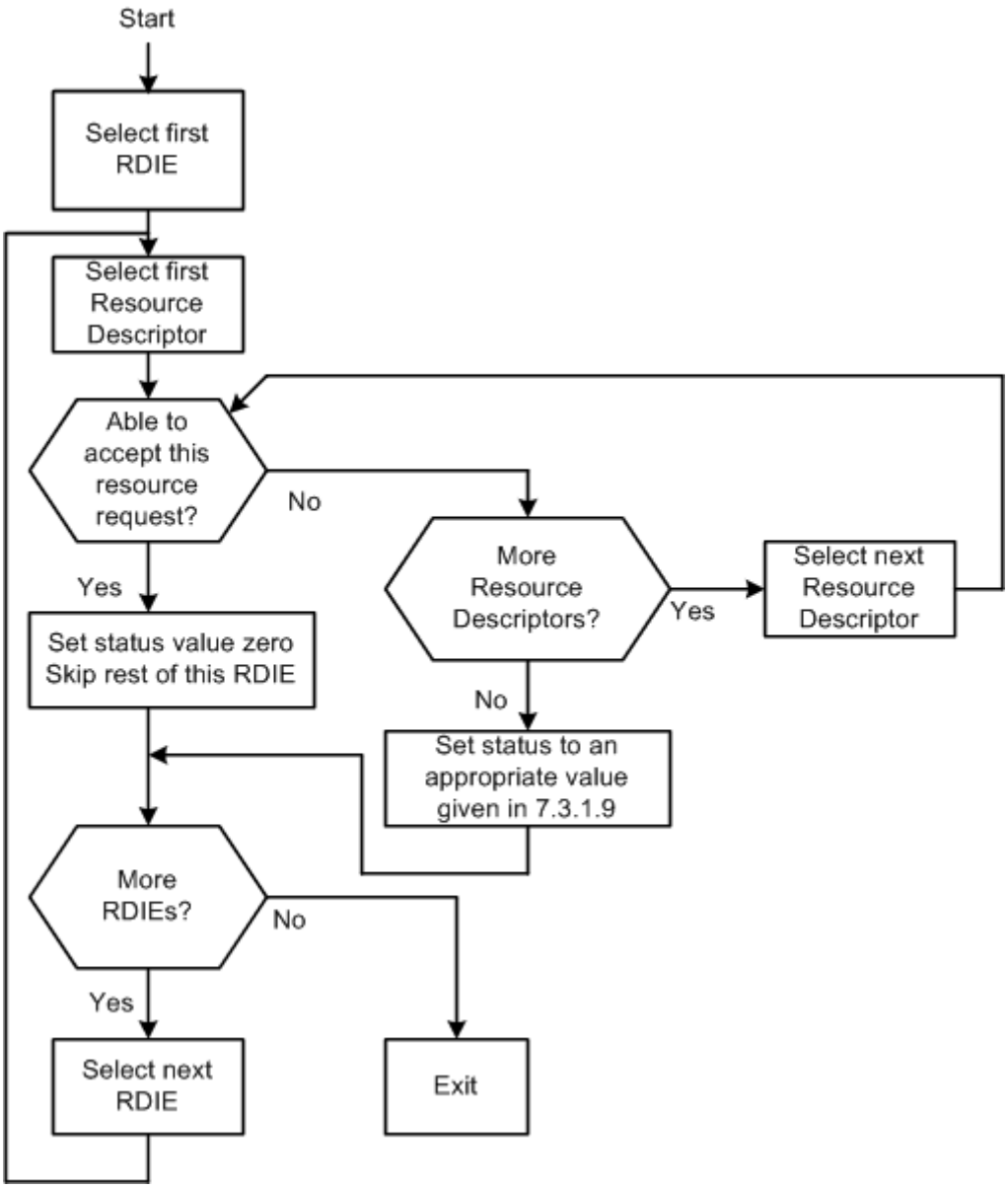

Figure 11A-30—Overview of RIC processing at an AP

In response to a RIC-Request, the AP shall construct a RIC-Response. The RIC-Response shall contain one RDIE for each RDIE in the RIC-Request. The RDIEs shall be in the same order as in the request and the RDIE Identifier field in each RDIE shall be the value of the RDIE Identifier field in the corresponding RDIE in the request. The Status Code field in the RDIE shall be set according to the result of the allocation request as follows:

— Status code = 0 indicates that the resource request has been accepted. The RDIE shall also be followed by the Resource Descriptor that was accepted.

— Status code = nonzero indicates that the resources could not be accepted. The Status Code field contains a value from 7.3.1.9 indicating the reason for the failure. In this case, the AP may include a single Resource Descriptor following the RDIE indicating a suggested resource that could have been accepted. The Resource Count field shall be set to 0 or 1 depending whether the suggested Resource

Descriptor is attached. A nonzero status code in an RDIE shall not cause a nonzero status code in the frame containing the RIC.

If the resource request included QoS resources and is successful, then the procedures for handling of TSPEC, TCLAS, and TCLAS Processing elements shall be as specified in 11.4, and the AP shall place the TSs into the accepted state. The RIC-Response shall contain the updated accepted TSPEC. Each RDIE may also include a Schedule information element (as defined in 7.3.2.34) after the accepted TSPEC. Upon reassociation, AP shall move all of the TSs from the accepted state into the active state.

If the STA does not invoke a reassociation within the reassociation deadline, then the TSs that had been accepted shall become inactive, and the resources shall be released. At the point that the STA reassociates with the target AP (within the reassociation deadline, if appropriate), the TSs are put into the active state. This may be immediate if the RIC-Request was part of a Reassociation Request frame.

Annex A

(normative)

Protocol Implementation Conformance Statement (PICS) proforma

A.4 PICS proforma—IEEE Std 802.11-2007

A.4.3 Implementation under test (IUT) configuration

Insert the following row at the end of the table in A.4.3:

Item	IUT configuration	References	Status	Support
*CF14	Is infrastructure mode implemented?	5.2.2	O	Yes ☐ No ☐

A.4.4 MAC protocol

A.4.4.1 MAC protocol capabilities

Change entry for PC34, and insert the following rows at the end of the table in A.4.4.1:

Item	Protocol capability	References	Status	Support
*PC34	Robust security network association (RSNA)	7.2.2, 7.3.1.4, 5.4.3.3, 8.7.2, 11.3.1, 11.3.2, 8.3.3	O	Yes ☐ No ☐
*PC35	Fast basic service set (BSS) transition (FT)	11A	CF14:O	Yes ☐ No ☐ N/A ☐
PC35.1	Mobility Domain information element (MDIE)	7.3.2.47	PC35:M	Yes ☐ No ☐ N/A ☐
PC35.2	Fast basic service set (BSS) Transition information element (FTIE)	7.3.2.48	PC35&PC34:M	Yes ☐ No ☐ N/A ☐
PC35.3	Timeout Interval information element (TIE)	7.3.2.49	PC35:M	Yes ☐ No ☐ N/A ☐
PC35.4	Fast basic service set (BSS) Transition (FT) authentication algorithm	7.3.1.1	PC35:M	Yes ☐ No ☐ N/A ☐
PC35.5	Fast basic service set (BSS) Transition (FT) Action frames	7.4.8	PC35:M	Yes ☐ No ☐ N/A ☐
PC35.6	Fast basic service set (BSS) Transition (FT) key management based on IEEE 802.1X	8.5.1.5, 7.3.2.25	PC35&PC34:M	Yes ☐ No ☐ N/A ☐

Item	Protocol capability	References	Status	Support
PC35.7	Fast basic service set (BSS) Transition (FT) key management based on preshared keys (PSKs)	8.5.1.5, 7.3.2.25	PC35&PC34:M	Yes ☐ No ☐ N/A ☐
PC35.8	Fast basic service set (BSS) Transition (FT) key hierarchy	8.5.1.5	PC35&PC34:M	Yes ☐ No ☐ N/A ☐
PC35.9	FT initial mobility domain association	11A.4	PC35&PC34:M	Yes ☐ No ☐ N/A ☐
PC35.10	Fast Basic Service Set (BSS) Transition (FT) Protocol	11A.5	PC35:M	Yes ☐ No ☐ N/A ☐
PC35.10.1	Fast Basic Service Set (BSS) Transition (FT) Protocol in robust security network (RSN)	11A.5.2, 11A.5.3, 11A.7.1	PC35&PC34:M	Yes ☐ No ☐ N/A ☐
PC35.10.2	Fast Basic Service Set (BSS) Transition (FT) Protocol in non-robust security network (non-RSN)	11A.5.4, 11A.5.5, 11A.7.2	PC35:M	Yes ☐ No ☐ N/A ☐
*PC35.11	Fast Basic Service Set (BSS) Transition (FT) Resource Request Protocol	11A.6	PC35:O	Yes ☐ No ☐ N/A ☐
PC35.11.1	Resource Request protocol over the air	11A.6.2	PC35.11:M	Yes ☐ No ☐ N/A ☐
PC35.11.2	Resource Request protocol over the distribution system (DS)	11A.6.3, 11A.10	PC35.11:M	Yes ☐ No ☐ N/A ☐
PC35.12	QoS procedures for fast basic service set (BSS) transition	11A.11	CF12& PC35:M	Yes ☐ No ☐ N/A ☐
*PC35.13	Resource Information Container (RIC) Data information element (RDIE)	11A.11, 7.3.2.50	PC35:M	Yes ☐ No ☐ N/A ☐
PC35.13.1	Resource Request Procedures at the station (STA)	11A.11.3.1	PC35.13:M	Yes ☐ No ☐ N/A ☐
PC35.13.2	Resource Request Procedures at the target access point (AP)	11A.11.3.2	PC35.13:M	Yes ☐ No ☐ N/A ☐
*PC35.14	Remote Request Procedures at the current access point (AP)	11A.10	PC35:M	Yes ☐ No ☐ N/A ☐
PC35.14.1	Remote Request/Response frame support	11A.10.3	PC35.14:O	Yes ☐ No ☐ N/A ☐
PC35.14.2	Vendor-specific remote request broker (RRB) mechanism	11A.10.3	PC35.14:O	Yes ☐ No ☐ N/A ☐

Annex D

(normative)

ASN.1 encoding of the MAC and PHY MIB

In "Major sections" in Annex D, add the following text at the end of the Station Management attributes:

```
--*********************************************************************
--* 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 part of an IEEE 802.11
    -- network."

    dot11smt OBJECT IDENTIFIER ::= { ieee802dot11 1 }

        -- dot11smt GROUPS
        -- dot11StationConfigTable                    ::= { dot11smt 1 }
        -- dot11AuthenticationAlgorithmTable          ::= { dot11smt 2 }
        -- dot11WEPDefaultKeysTable                   ::= { dot11smt 3 }
        -- dot11WEPKEYMappingsTable                   ::= { dot11smt 4 }
        -- dot11PrivacyTable                          ::= { dot11smt 5 }
        -- dot11SMTnotification                       ::= { dot11smt 6 }
        -- dot11MultiDomainCapabilityTable            ::= { dot11smt 7 }
        -- dot11SpectrumManagementTable               ::= { dot11smt 8 }
        -- dot11RSNAConfigTable                       ::= { dot11smt 9 }
        -- dot11RSNAConfigPairwiseCiphersTable        ::= { dot11smt 10 }
        -- dot11RSNAConfigAuthenticationSuitesTable   ::= { dot11smt 11 }
        -- dot11RSNAStatsTable                        ::= { dot11smt 12 }
        -- dot11RegulatoryClassesTable                ::= { dot11smt 13 }
        -- dot11RadioResourceManagement               ::= { dot11smt 14 }
        -- dot11FastBSSTransitionConfigTable          ::= { dot11smt 15 }
    --
```

In "dotStationConfig TABLE" in Annex D, change Dot11StationConfigEntry as follows:

```
-- ********************************************************************
-- * dotStationConfig TABLE
-- ********************************************************************

Dot11StationConfigEntry ::=
   SEQUENCE {
       dot11StationID                                  MacAddress,
       dot11MediumOccupancyLimit                       INTEGER,
       dot11CFPollable                                 TruthValue,
```

```
        dot11CFPeriod                                   INTEGER,
        dot11CFPMaxDuration                             INTEGER,
        dot11AuthenticationResponseTimeOut              Unsigned32,
        dot11PrivacyOptionImplemented                   TruthValue,
        dot11PowerManagementMode                        INTEGER,
        dot11DesiredSSID                                OCTET STRING,
        dot11DesiredBSSType                             INTEGER,
        dot11OperationalRateSet                         OCTET STRING,
        dot11BeaconPeriod                               INTEGER,
        dot11DTIMPeriod                                 INTEGER,
        dot11AssociationResponseTimeOut                 Unsigned32,
        dot11DisassociateReason                         INTEGER,
        dot11DisassociateStation                        MacAddress,
        dot11DeauthenticateReason                       INTEGER,
        dot11DeauthenticateStation                      MacAddress,
        dot11AuthenticateFailStatus                     INTEGER,
        dot11AuthenticateFailStation                    MacAddress,
        dot11MultiDomainCapabilityImplemented           TruthValue,
        dot11MultiDomainCapabilityEnabled               TruthValue,
        dot11CountryString                              OCTET STRING,
        dot11SpectrumManagementImplemented              TruthValue,
        dot11SpectrumManagementRequired                 TruthValue,
        dot11RSNAOptionImplemented                      TruthValue,
        dot11RSNAPreauthenticationImplemented           TruthValue,
        dot11RegulatoryClassesImplemented               TruthValue,
        dot11RegulatoryClassesRequired                  TruthValue,
        dot11QosOptionImplemented                       TruthValue,
        dot11ImmediateBlockAckOptionImplemented         TruthValue,
        dot11DelayedBlockAckOptionImplemented           TruthValue,
        dot11DirectOptionImplemented                    TruthValue,
        dot11APSDOptionImplemented                      TruthValue,
        dot11QAckOptionImplemented                      TruthValue,
        dot11QBSSLoadOptionImplemented                  TruthValue,
        dot11QueueRequestOptionImplemented              TruthValue,
        dot11TXOPRequestOptionImplemented               TruthValue,
        dot11MoreDataAckOptionImplemented               TruthValue,
        dot11AssociateinNQBSS                           TruthValue,
        dot11DLSAllowdInQBSS                            TruthValue,
        dot11DLSAllowed                                 TruthValue,
        dot11AssociateStation                           MacAddress,
        dot11AssociateID                                INTEGER,
        dot11AssociateFailStation                       MacAddress,
        dot11AssociateFailStatus                        INTEGER,
        dot11ReassociateStation                         MacAddress,
        dot11ReassociateID                              INTEGER,
        dot11ReassociateFailStation                     MacAddress,
        dot11ReassociateFailStatus                      INTEGER,
        dot11RadioMeasurementCapable                    TruthValue,
        dot11RadioMeasurementEnabled                    TruthValue,
        dot11RRMMeasurementProbeDelay                   INTEGER,
        dot11RRMMeasurementPilotPeriod                  INTEGER,
        dot11RRMLinkMeasurementEnabled                  TruthValue,
        dot11RRMNeighborReportEnabled                   TruthValue,
        dot11RRMParallelMeasurementsEnabled             TruthValue,
```

```
        dot11RRMRepeatedMeasurementsEnabled                 TruthValue,
        dot11RRMBeaconPassiveMeasurementEnabled             TruthValue,
        dot11RRMBeaconActiveMeasurementEnabled              TruthValue,
        dot11RRMBeaconTableMeasurementEnabled               TruthValue,
        dot11RRMBeaconMeasurementReportingConditionsEnabled
                                                            TruthValue,
        dot11RRMFrameMeasurementEnabled                     TruthValue,
        dot11RRMChannelLoadMeasurementEnabled               TruthValue,
        dot11RRMNoiseHistogramMeasurementEnabled            TruthValue,
        dot11RRMStatisticsMeasurementEnabled                TruthValue,
        dot11RRMLCIMeasurementEnabled                       TruthValue,
        dot11RRMLCIAzimuthEnabled                           TruthValue,
        dot11RRMTransmitStreamCategoryMeasurementEnabled
                                                            TruthValue,
        dot11RRMTriggeredTransmitStreamCategoryMeasurementEnabled
                                                            TruthValue,
        dot11RRMAPChannelReportEnabled                      TruthValue,
        dot11RRMMIBEnabled                                  TruthValue,
        dot11RRMMaxMeasurementDuration                      Unsigned32,
        dot11RRMNonOperatingChannelMaxMeasurementDuration
                                                            Unsigned32,
        dot11RRMMeasurementPilotTransmissionInformationEnabled
                                                            TruthValue,
        dot11RRMMeasurementPilotCapability                  Unsigned32,
        dot11RRMNeighborReportTSFOffsetEnabled              TruthValue,
        dot11RRMRCPIMeasurementEnabled                      TruthValue,
        dot11RRMRSNIMeasurementEnabled                      TruthValue,
        dot11RRMBSSAverageAccessDelayEnabled                TruthValue,
        dot11RRMBSSAvailableAdmissionCapacityEnabled
                                                            TruthValue,
        dot11RRMAntennaInformationEnabled                   TruthValue,
        dot11FastBSSTransitionImplemented                   TruthValue  }
```

In "dotStationConfig TABLE" in Annex D, insert the following attribute after dot11RRMAntennaInformationEnabled { dot11StationConfigEntry 82 }:

```
dot11FastBSSTransitionImplemented OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-only
          STATUS current
       DESCRIPTION
              "This object indicates if the entity is fast BSS transition
              capable."
       ::= { dot11StationConfigEntry 83 }
```

In "dot11AuthenticationAlgorithms TABLE" in Annex D, change dot11AuthenticationAlgorithmsTable as follows:

```
-- ************************************************************************
-- * dot11AuthenticationAlgorithms 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
                Value = 3: Fast BSS Transition (FT)"
       REFERENCE "IEEE Std 802.11-2007, 7.3.1.1"
   ::= { dot11smt 2 }
```

IN "dot11AuthenticationAlgorithms TABLE" in Annex D, change dot11AuthenticationAlgorithm as follows:

```
dot11AuthenticationAlgorithm OBJECT-TYPE
          SYNTAX INTEGER { openSystem(1), sharedKey(2),
                fastBSSTransition(3) }
          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~~the
             authentication algorithm described by this entry in the
             table. The following values can be used here.
                Value = 1: Open System
                Value = 2: Shared Key"
                Value = 3: Fast BSS Transition (FT)"
   ::= { dot11AuthenticationAlgorithmsEntry 2 }
```

After "dot11RegulatoryClasses TABLE" and before "MAC Attribute Templates" in Annex D, insert the following dot11FastBSSTransitionConfig TABLE:

```
-- ************************************************************************
-- * dot11FastBSSTransitionConfig TABLE
-- ************************************************************************
dot11FastBSSTransitionConfigTable OBJECT-TYPE
          SYNTAX SEQUENCE OF Dot11FastBSSTransitionConfigEntry
```



```
          MAX-ACCESS not-accessible
          STATUS current
       DESCRIPTION
             "The table containing fast BSS transition configuration
             objects."
   ::= { dot11smt 15 }

dot11FastBSSTransitionConfigEntry OBJECT-TYPE
          SYNTAX Dot11FastBSSTransitionConfigEntry
          MAX-ACCESS not-accessible
          STATUS current
       DESCRIPTION
             "An entry in the dot11FastBSSTransitionConfigTable."
       INDEX { ifIndex }
   ::= { dot11FastBSSTransitionConfigTable 1 }

Dot11FastBSSTransitionConfigEntry ::=
   SEQUENCE {
       dot11FastBSSTransitionEnabled         TruthValue,
       dot11FTMobilityDomainID               OCTET STRING,
       dot11FTOverDSEnabled                  TruthValue,
       dot11FTResourceRequestSupported       TruthValue,
       dot11FTR0KeyHolderID                  OCTET STRING,
       dot11FTR0KeyLifetime                  Unsigned32,
       dot11FTR1KeyHolderID                  OCTET STRING,
       dot11FTReassociationDeadline          Unsigned32 }

dot11FastBSSTransitionEnabled OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
       DESCRIPTION
             "When this object is set to TRUE, this shall indicate that
             fast BSS transition (FT) is enabled on this entity. The
             entity will advertise the FT-related information elements
             in its Beacon and Probe Response frames. This object
             requires that dot11FastBSSTransitionImplemented also be set
             to TRUE."
   ::= { dot11FastBSSTransitionConfigEntry 1 }

dot11FTMobilityDomainID OBJECT-TYPE
          SYNTAX OCTET STRING (SIZE(2))
          MAX-ACCESS read-write
          STATUS current
       DESCRIPTION
             "This attribute shall specify the Mobility Domain
             identifier (MDID) of this entity.
             The MDID is used to indicate a group of APs, within an ESS,
             between which a STA can use fast BSS transition services.
             Fast BSS transitions are allowed only between APs that have
             the same MDID and are within the same ESS. They are not
             allowed between APs with different MDIDs or in different
             ESSs.
```

```
            Since fast BSS transition services are defined only within
            the scope of an ESS, there is no requirement that MDIDs be
            unique across ESSs."
    ::= { dot11FastBSSTransitionConfigEntry 2 }

dot11FTOverDSEnabled OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
       DESCRIPTION
            "When this object is set to TRUE, this shall indicate that
            fast BSS transition via the over-the-DS protocol as
            described in Clause 11A is enabled on this AP entity."
    ::= { dot11FastBSSTransitionConfigEntry 3 }

dot11FTResourceRequestSupported OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
       DESCRIPTION
            "When this object is set to TRUE, this shall indicate that
            the fast BSS transition (FT) resource request procedures of
            11A.10 are supported on this AP entity."
    ::= { dot11FastBSSTransitionConfigEntry 4 }

dot11FTR0KeyHolderID OBJECT-TYPE
          SYNTAX OCTET STRING (SIZE(1..48))
          MAX-ACCESS read-write
          STATUS current
       DESCRIPTION
            "This attribute shall specify the PMK-R0 key holder
            identifier (R0KH-ID) of the Authenticator of this AP.
            NOTE: Backend protocol may allow longer NAS Client
            identifiers (e.g., RADIUS allows up to 253-octet NAS-
            Identifier), but when used with fast BSS transition, the
            maximum length is limited to 48 octets. The same value must
            be used for the NAS Client identifier and
            dot11FTR0KeyHolderID."
    ::= { dot11FastBSSTransitionConfigEntry 5 }

dot11FTR0KeyLifetime OBJECT-TYPE
          SYNTAX Unsigned32 (60..4294967295)
          MAX-ACCESS read-write
          STATUS current
       DESCRIPTION
            "This attribute shall specify the default lifetime of the
            PMK-R0, in seconds, when a Session-Timeout attribute is not
            provided during the EAP authentication. This attribute
            shall also apply when the PMK-R0 is derived from a PSK."
          DEFVAL { 1209600 }
    ::= { dot11FastBSSTransitionConfigEntry 6 }
```



```
dot11FTR1KeyHolderID OBJECT-TYPE
          SYNTAX OCTET STRING (SIZE(6))
          MAX-ACCESS read-write
          STATUS current
       DESCRIPTION
             "This attribute shall specify the PMK-R1 key holder
             identifier (R1KH-ID) of the Authenticator of this AP. It
             shall be set to a MAC address of the entity holding the
             PMK-R1 in the Authenticator."
   ::= { dot11FastBSSTransitionConfigEntry 7 }

dot11FTReassociationDeadline OBJECT-TYPE
          SYNTAX Unsigned32 (1000..4294967295)
          MAX-ACCESS read-write
          STATUS current
       DESCRIPTION
             "This attribute shall specify the number of time units
             (TUs) that this target AP entity shall retain a PTKSA and
             reserve any specified resources for a STA while waiting for
             a reassociation from that STA. It is assumed that this
             value will be administered consistently across the mobility
             domain"
          DEFVAL { 1000 }
   ::= { dot11FastBSSTransitionConfigEntry 8 }

-- ****************************************************************
-- * End of dot11FastBSSTransitionConfig TABLE
-- ****************************************************************
```

In "Compliance Statements" in Annex D, change dot11Compliance MODULE-COMPLIANCE as follows:

```
-- ****************************************************************
-- * 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 {
          dot11SMTbase~~7~~8,dot11MACbase2, dot11CountersGroup2,
          dot11SmtAuthenticationAlgorithms, dot11ResourceTypeID,
          dot11PhyOperationComplianceGroup }
```

In "Compliance Statements" in Annex D, change OPTIONAL-GROUPS as follows:

```
-- OPTIONAL-GROUPS { dot11SMTprivacy, dot11MACStatistics,
--    dot11PhyAntennaComplianceGroup, dot11PhyTxPowerComplianceGroup,
--    dot11PhyRegDomainsSupportGroup,
--    dot11PhyAntennasListGroup, dot11PhyRateGroup,
--    dot11SMTbase3, dot11MultiDomainCapabilityGroup,
--    dot11PhyFHSSComplianceGroup2, dot11RSNAadditions,
--    dot11RegulatoryClassesGroup, dot11Qosadditions,
--    dot11RRMCompliance, dot11FTComplianceGroup }
--
::= { dot11Compliances 1 }
```

In "Groups - units of conformance" in Annex D, change dot11SMTbase7 as follows:

```
dot11SMTbase7 OBJECT-GROUP
       OBJECTS {dot11MediumOccupancyLimit,
                dot11CFPollable,
                dot11CFPPeriod,
                dot11CFPMaxDuration,
                dot11AuthenticationResponseTimeOut,
                dot11PrivacyOptionImplemented,
                dot11PowerManagementMode,
                dot11DesiredSSID, dot11DesiredBSSType,
                dot11OperationalRateSet,
                dot11BeaconPeriod, dot11DTIMPeriod,
                dot11AssociationResponseTimeOut,
                dot11DisassociateReason,
                dot11DisassociateStation,
                dot11DeauthenticateReason,
                dot11DeauthenticateStation,
                dot11AuthenticateFailStatus,
                dot11AuthenticateFailStation,
                dot11MultiDomainCapabilityImplemented,
                dot11MultiDomainCapabilityEnabled,
                dot11CountryString,
                dot11SpectrumManagementImplemented,
                dot11SpectrumManagementRequired,
                dot11RSNAOptionImplemented,
                dot11RegulatoryClassesImplemented,
                dot11RegulatoryClassesRequired,
                dot11QosOptionImplemented,
                dot11ImmediateBlockAckOptionImplemented,
                dot11DelayedBlockAckOptionImplemented,
                dot11DirectOptionImplemented,
                dot11APSDOptionImplemented,
                dot11QAckOptionImplemented,
                dot11QBSSLoadOptionImplemented,
                dot11QueueRequestOptionImplemented,
                dot11TXOPRequestOptionImplemented,
                dot11MoreDataAckOptionImplemented,
```



```
                dot11AssociateinNQBSS,
                dot11DLSAllowedInQBSS,
                dot11DLSAllowed,
                dot11AssociateStation,
                dot11AssociateID,
                dot11AssociateFailStation,
                dot11AssociateFailStatus,
                dot11ReassociateStation,
                dot11ReassociateID,
                dot11ReassociateFailStation,
                dot11ReassociateFailStatus,
                dot11RadioMeasurementCapable,
                dot11RadioMeasurementEnabled,
                dot11RRMMeasurementProbeDelay,
                dot11RRMMeasurementPilotPeriod,
                dot11RRMLinkMeasurementEnabled,
                dot11RRMNeighborReportEnabled,
                dot11RRMParallelMeasurementsEnabled,
                dot11RRMRepeatedMeasurementsEnabled,
                dot11RRMBeaconPassiveMeasurementEnabled,
                dot11RRMBeaconActiveMeasurementEnabled,
                dot11RRMBeaconTableMeasurementEnabled,
                dot11RRMBeaconMeasurementReportingConditionsEnabled,
                dot11RRMFrameMeasurementEnabled,
                dot11RRMChannelLoadMeasurementEnabled,
                dot11RRMNoiseHistogramMeasurementEnabled,
                dot11RRMStatisticsMeasurementEnabled,
                dot11RRMLCIMeasurementEnabled,
                dot11RRMLCIAzimuthEnabled,
                dot11RRMTransmitStreamCategoryMeasurementEnabled,
                dot11RRMTriggeredTransmitStreamCategoryMeasurementEnabled,
                dot11RRMAPChannelReportEnabled,
                dot11RRMMIBEnabled,
                dot11RRMMaxMeasurementDuration,
                dot11RRMNonOperatingChannelMaxMeasurementDuration,
                dot11RRMMeasurementPilotTransmissionInformationEnabled,
                dot11RRMMeasurementPilotCapability,
                dot11RRMNeighborReportTSFOffsetEnabled,
                dot11RRMRCPIMeasurementEnabled,
                dot11RRMRSNIMeasurementEnabled,
                dot11RRMBSSAverageAccessDelayEnabled,
                dot11RRMBSSAvailableAdmissionCapacityEnabled,
                dot11RRMAntennaInformationEnabled}
                STATUS ~~current~~deprecated
    DESCRIPTION
          "The SMTbase7 object class provides the necessary support
          at the STA to manage the processes in the STA so that the
          STA may work cooperatively as a part of an IEEE 802.11
          network, when the STA is capable of multidomain operation.
          This object group should be implemented when the
          multidomain capability option is implemented."
::= { dot11Groups 36 }
```

In "Groups - units of conformance" of Annex D, insert the following objects after dot11SMTbase7 { dot11Groups 36 }:

```
dot11FTComplianceGroup OBJECT-GROUP
          OBJECTS { dot11FastBSSTransitionEnabled }
          STATUS current
       DESCRIPTION
             "This object class provides the objects from the IEEE
             802.11 MIB required to manage fast BSS transition
             functionality. Note that additional objects for managing
             this functionality are located in the dot11FastBSS
             TransitionConfigTable."
   ::= { dot11Groups 40}

dot11SMTbase8 OBJECT-GROUP
       OBJECTS {dot11MediumOccupancyLimit,
                dot11CFPollable,
                dot11CFPPeriod,
                dot11CFPMaxDuration,
                dot11AuthenticationResponseTimeOut,
                dot11PrivacyOptionImplemented,
                dot11PowerManagementMode,
                dot11DesiredSSID, dot11DesiredBSSType,
                dot11OperationalRateSet,
                dot11BeaconPeriod, dot11DTIMPeriod,
                dot11AssociationResponseTimeOut,
                dot11DisassociateReason,
                dot11DisassociateStation,
                dot11DeauthenticateReason,
                dot11DeauthenticateStation,
                dot11AuthenticateFailStatus,
                dot11AuthenticateFailStation,
                dot11MultiDomainCapabilityImplemented,
                dot11MultiDomainCapabilityEnabled,
                dot11CountryString,
                dot11SpectrumManagementImplemented,
                dot11SpectrumManagementRequired,
                dot11RSNAOptionImplemented,
                dot11RegulatoryClassesImplemented,
                dot11RegulatoryClassesRequired,
                dot11QosOptionImplemented,
                dot11ImmediateBlockAckOptionImplemented,
                dot11DelayedBlockAckOptionImplemented,
                dot11DirectOptionImplemented,
                dot11APSDOptionImplemented,
                dot11QAckOptionImplemented,
                dot11QBSSLoadOptionImplemented,
                dot11QueueRequestOptionImplemented,
                dot11TXOPRequestOptionImplemented,
                dot11MoreDataAckOptionImplemented,
                dot11AssociateinNQBSS,
                dot11DLSAllowedinQBSS,
                dot11DLSAllowed,
```



```
                dot11AssociateStation,
                dot11AssociateID,
                dot11AssociateFailStation,
                dot11AssociateFailStatus,
                dot11ReassociateStation,
                dot11ReassociateID,
                dot11ReassociateFailStation,
                dot11ReassociateFailStatus,
                dot11RadioMeasurementCapable,
                dot11RadioMeasurementEnabled,
                dot11RRMMeasurementProbeDelay,
                dot11RRMMeasurementPilotPeriod,
                dot11RRMLinkMeasurementEnabled,
                dot11RRMNeighborReportEnabled,
                dot11RRMParallelMeasurementsEnabled,
                dot11RRMRepeatedMeasurementsEnabled,
                dot11RRMBeaconPassiveMeasurementEnabled,
                dot11RRMBeaconActiveMeasurementEnabled,
                dot11RRMBeaconTableMeasurementEnabled,
                dot11RRMBeaconMeasurementReportingConditionsEnabled,
                dot11RRMFrameMeasurementEnabled,
                dot11RRMChannelLoadMeasurementEnabled,
                dot11RRMNoiseHistogramMeasurementEnabled,
                dot11RRMStatisticsMeasurementEnabled,
                dot11RRMLCIMeasurementEnabled,
                dot11RRMLCIAzimuthEnabled,
                dot11RRMTransmitStreamCategoryMeasurementEnabled,
                dot11RRMTriggeredTransmitStreamCategoryMeasurementEnabled,
                dot11RRMAPChannelReportEnabled,
                dot11RRMMIBEnabled,
                dot11RRMMaxMeasurementDuration,
                dot11RRMNonOperatingChannelMaxMeasurementDuration,
                dot11RRMMeasurementPilotTransmissionInformationEnabled,
                dot11RRMMeasurementPilotCapability,
                dot11RRMNeighborReportTSFOffsetEnabled,
                dot11RRMRCPIMeasurementEnabled,
                dot11RRMRSNIMeasurementEnabled,
                dot11RRMBSSAverageAccessDelayEnabled,
                dot11RRMBSSAvailableAdmissionCapacityEnabled,
                dot11RRMAntennaInformationEnabled,
                dot11FastBSSTransitionImplemented }
    STATUS current
    DESCRIPTION
          "The SMTbase8 object class provides the necessary support
          at the STA to manage the processes in the STA so that the
          STA may work cooperatively as a part of an IEEE 802.11
          network, when the STA is capable of multidomain operation.
          This object group should be implemented when the
          multidomain capability option is implemented."
::= { dot11Groups 41 }
```

Annex Q

(normative)

ASN.1 encoding of the RRM MIB

In dot11RRMNeighborReport TABLE in Annex Q, change Dot11RRMNeighborReportEntry as follows:

```
Dot11RRMNeighborReportEntry ::=
   SEQUENCE {
       dot11RRMNeighborReportIndex                   Unsigned32,
       dot11RRMNeighborReportIfIndex                 InterfaceIndex,
       dot11RRMNeighborReportBSSID                   MacAddress,
       dot11RRMNeighborReportAPReachability          INTEGER,
       dot11RRMNeighborReportSecurity                TruthValue,
       dot11RRMNeighborReportCapSpectrumMgmt         TruthValue,
       dot11RRMNeighborReportCapQoS                  TruthValue,
       dot11RRMNeighborReportCapAPSD                 TruthValue,
       dot11RRMNeighborReportCapRRM                  TruthValue,
       dot11RRMNeighborReportCapDelayBlockAck        TruthValue,
       dot11RRMNeighborReportCapImmediateBlockAck    TruthValue,
       dot11RRMNeighborReportKeyScope                TruthValue,
       dot11RRMNeighborReportRegulatoryClass         INTEGER,
       dot11RRMNeighborReportChannelNumber           INTEGER,
       dot11RRMNeighborReportPhyType                 INTEGER,
       dot11RRMNeighborReportNeighborTSInfo          OCTET STRING,
       dot11RRMNeighborReportPilotInterval           Unsigned32,
       dot11RRMNeighborReportPilotMultipleBSSID      OCTET STRING,
       dot11RRMNeighborReportRRMEnabledCapabilities  OCTET STRING,
       dot11RRMNeighborReportVendorSpecific          OCTET STRING,
       dot11RRMNeighborReportRowStatus               RowStatus,
       dot11RRMNeighborReportMobilityDomain          TruthValue }
```

In "dot11RRMNeighborReport TABLE" in Annex Q, insert the following object after dot11RRMNeighborReportRowStatus { dot11RRMNeighborReportEntry 21}:

```
dot11RRMNeighborReportMobilityDomain OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-create
          STATUS current
      DESCRIPTION
             "Indicates a common mobility domain identifier (MDID) and
             an identical value of the FT Capability and Policy value."
   ::= { dot11RRMNeighborReportEntry 22 }
```