IEEE P802.11
Wireless LANs

|  |
| --- |
| 802.11 TGmb Subclause 10.3 Clean Up Proposal |
| Date: 2010-11-10 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Mark Hamilton | Polycom | 5755 Central AveBoulder, CO 80301 | 303-583-5239 | mark.hamilton@polycom.com |
|  |  |  |  |  |

Abstract

This document presents modifications to subclause 10.3 of TGmb Draft 6.0. These modifications are based on 10/0728r2 and 10/0826r5 (both of which were based on an earlier Draft), updated to show as changes to Draft 6.0.

These modifications are presented as a proposal to address CIDs on the first TGmb Letter Ballot, as listed below.

# The Comments resolved by this proposal:

| **CID** | **Page** | **Clause** | **Comment** | **Proposed Change** |
| --- | --- | --- | --- | --- |
| 10008 | 628.58 | 10.3 | 1. We have two separate variables in 10.3.1, but state elsewhere is treated as an enumeration.2. The state machine cannot be localized to either the SME or the MLME. Some of its transitions respond to events known only within the MLME and some to events known only within the SME. Having a state machine split over two architectural entities in this way implies the existence of communications not documented in the MLME-SAP. The distinction between states 3 and 4 has no effect on the operation of the MAC or MLME, and therefore serves no purpose. The experiment post D1.0 of REVmb to "clarify" this state has had the opposite effect and should be deemed a failed experiment.3. The location of the state variable and the responsibility for maintaining it are not defined.4. Given that the state variable is located in one or other of the MLME or SME, there is no interface to allow this variable to be read from the other architectural entity.5. Some of the text is impenetrable and needs to be restructured (e.g. 10.3.3.3 bullet c).I am lumping these comments together because I believe they need to be considered together. | 1. Replace two state variables with an enumeration2. Remove the distinction between states 3 and 4 within the MAC and MLME.3. Indicate that the state variable sits in the MLME and is maintained by the MLME and update language to reflect this.4. Add an MLME interface to export the MLME state variable (read-only)5. Restructure impenetrable text to improve readability, e.g. by adding sub-bullets inside longwinded list items. |
| 10084 | 629.04 | 10.3 | Restore the 3 state model, as there are layering issues that cannot be accurately represented currently. | As in comment. |
| 10094 | 628.58 | 10.3 | It needs to be clarified in this subclause which actions labeled as "STA shall" are really "MLME shall" and which are "SME shall" | See changes proposed in 10/0728r2 |
| 10095 | 628.58 | 10.3 | It is unclear what "appropriate reason code" implies in this subclause Use the reason code delivered via the various .request primitives. | See changes proposed in 10/0728r2 |
| 10096 | 628.58 | 10.3 | Several bullet list items in this subsection are impossible to understand. | See changes proposed in 10/0728r2 |
| 10097 | 628.58 | 10.3 | The architectural model is blurred in this sub-clause. It is not clear: 1) how memory of a previous failed SA Query procedure is retained, and whether this is in the MLME or SME; 2) whether the sending of the protected disassociation frame is an action of the MLME or SME. It is likely the latter, but it is expressed as though it were the former. | See changes proposed in 10/0728r2 |
| 10098 | 628.58 | 10.3 | Text in this sub-clause assumes the SME can see the contents of the Reassociation response frame, which is not true. | See changes proposed in 10/0728r2 |
| 10099 | 628.58 | 10.3 | 10.3.2.2 and 10.3.2.3 say the PTKSA gets deleted at 802.11 Authentication. 802.11 Authentication has nothing to do with the PTKSA | See changes proposed in 10/0826r5 |
| 10100 | 628.58 | 10.3 | Authentication failure does not change state (at either STA) even if not MFP (this is e.g. to avoid DoS between auth and asoc where an attacker keeps injecting fake broken auth frames while you're trying to get associated so you can engage MFP). Authentication success takes you from state 1 to state 2 but does not take you down from state 3 or 4 to state 2, even if not MFP. This is for protection against open system authentication attack (in MFP especially), by not going from state 3 or 4 down if authentication requested or even succeeds. | See changes proposed in 10/0826r5 |
| 10101 | 632.05 | 10.3.2.3 | The inserted text "The STA shall be set to state 1" is incorrect, and leads to an denial-of-service attack. | See changes proposed in 10/0826r5 |
| 10102 | 628.58 | 10.3 | Failure to (re)assoc on STA does not change state or delete keys in FT case. On tx of (re)assoc on STA if not FT then drop keys and go to state 3 if in state 4 (you have to drop keys if MFP so can be ready for 4WH). Failure to (re)assoc on AP does not change state or delete keys on FT case or MFP case (to avoid DOS). On rx of (re)assoc on AP if not FT or MFP then AP has to drop keys and go to state 3 if in state 4. On successful (re)assoc on AP if MFP then AP drops keys and goes to state 3 if in state 4 -- no action on failure. Successful (re)assoc if in state 4 without FT or MFP therefore takes you back to state 3 (or lower -- see next point) and you have to redo the 4WH. Failed (re)assoc without FT or MFP does not change state. | See changes proposed in 10/0826r5 |
| 10103 | 628.58 | 10.3 | The transitions from states 3 and 4 to state 2 for unsuccessful (re)association are not true at an AP. Doing this transition at an AP opens a denial-of-service attack -- a rogue STA sending failing (re)association requests on behalf of a target STA causes the AP to put the target STA back into state 2 and deny it service. | See changes proposed in 10/0826r5 |
| 10104 | 628.58 | 10.3 | The descriptions of the transitions from State 3 to State 4 do not make it clear which entity (SME or MLME) writes to the state variable (it needs to be MLME to not have more than one writer), nor how the MLME has enough information to know when to do this transtion. The architectural model for MAC state (which is said to be maintained by MLME) and the 802.1X state, is confusing in this regard. This makes FT transition very hard to understand as well. | See changes proposed in 10/0826r5 |
| 10115 | 628.00 |  | The text in 10.3 has some inconsistencies as documented in 11-10-0728r2. | use 11-10/826 to correct the issues identified. |
| 10119 | 634.05 | 10.3.3.2 | Line 5 uses the phrase "if RSNA establishment is required" and line 26 "if policy within the SME requires secure communication". | make them the same |
| 10126 | 629.33 | 10.3.1 | The transitions from states 3 and 4 to state 2 for unsuccessful (re)association are not true at an AP. Doing this transition at an AP opens a denial-of-service attack -- a rogue STA sending failing (re)association requests on behalf of a target STA causes the AP to put the target STA back into state 2 and deny it service. | It is not possible to show a simple transition that covers both the non-AP STA and AP case, as the behavior is different. Either qualify the label on both Unsuccessful (re)association transitions by including "at a non-AP STA", or delete the two transitions completely from the figure (as it is better to be silent than to say something incorrect). |
| 10144 | 629.10 | 10.3 | Introduction of the new "State 4" makes description of MLME unnecessarily complex. While a description of combinations of MLME State and IEEE 802.1X Controlled Port may indeed be useful as an introduction to RSN, it does not look necessary for describing MLME behavior in normative text. | Consider reverting the introduction of State 4 to simplify MLME state description in 10.3. |
| 10145 | 632.22 | 10.3.2.3 | The authentication procedure steps e and f seem to open a denial-of-service issue in FT and MFP cases (which provide protection for this). There is no real authentication in Open System authentication algorithm, so anyone could easily drop the association if these rules were used at an AP. | Limit steps e and f to apply only if FT or MFP are not used (like step b is already doing). |

# The Changes

Clause 10.3 in its entirety is shown below, with tracked changes resulting from the above comment resolution.

***Change 10.3 as shown below. (NOTE, numbering is not correct and does not match Draft 6.0; that should be ignored.)***

* STA authentication and association
* General

A STA keeps an enumerated state variable for each STA with which direct communication via the WM is needed. This variable expresses the relationship between the local STA and the remote STA. It takes on the following values:

* *State 1:* Initial start state, unauthenticated, unassociated.
* *State 2:* Authenticated, not associated.
* *State 3:* Authenticated and associated (Pending RSN Authentication)
* *State 4:* Authenticated and associated

The state variable is kept within the MLME (i.e., is written and read by the MLME). The SME can also read this variable.

Figure 10-6 (Relationship between state and services) shows the state transition diagram for these STA states. Note that only events causing state changes are shown



**Figure 10-6—Relationship between state and services**

The current state existing between the source and destination STAs determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs (see Clause 8 (Frame formats)). A unique state exists for each pair of source and destination STAs. The state of the sending STA given by Figure 10-6 (Relationship between state and services) is with respect to the intended receiving STA. The allowed frame types are grouped into classes and the classes correspond to the STA state. In State 1, only Class 1 frames are allowed. In State 2, either Class 1 or Class 2 frames are allowed. In State 3 and State 4, all frames are allowed (Classes 1, 2, and 3). The frame classes are defined as follows:

* Class 1 frames
* Control frames
* RTS
* CTS
* ACK
* CF-End+ACK
* CF-End
* Within an IBSS, Block Ack (BlockAck)
* Within an IBSS, Block Ack Request (BlockAckReq)
* Management frames
* Probe Request/Response
* Beacon
* Authentication
* Deauthentication
* ATIM
* Public Action
* Within an IBSS, all Action frames and all Action No Ack frames
* Data frames
* Data frames between STAs in an IBSS
* Data frames between peers using DLS
* Class 2 frames
* Management frames
* Association Request/Response
* Reassociation Request/Response
* Disassociation
* Class 3 frames
* Data frames
* Data frames between STAs in an infrastructure BSS
* Management frames
* Within an infrastructure BSS, all Action and Action No Ack frames except those that are declared to be Class 1 or Class 2 frames (above)
* Control frames
* PS-Poll
* Within an infrastructure BSS, Block Ack (BlockAck)
* Within an infrastructure BSS, Block Ack Request (BlockAckReq)

Class 2 and Class 3 frames are not allowed in an IBSS. If a STA in an IBSS receives a Class 2 or Class 3 frame, it shall ignore the frame.

The use of the word “receive” in 10.3 (STA authentication and association) refers to a frame that meets all of the filtering criteria specified in Clause 11 (Security) and Clause 9 (MAC sublayer functional description).

* Authentication and deauthentication
* General

This subclause describes the procedures used for IEEE 802.11 authentication and deauthentication. The states used in this description are defined in 10.3.1 (General).

Successful authentication sets the STA's state to State 2, if it was in State 1. Unsuccessful authentication leaves the STA's state unchanged. The STA shall not transmit Class 2 frames unless in State 2 or State 3 or State 4. The STA shall not transmit Class 3 frames unless in State 3 or State 4

Deauthentication notification sets the STA's state to State 1. The STA shall become authenticated again prior to sending Class 2 frames. Deauthentication notification when in State 3 or 4 implies disassociation as well.

If STA A in an infrastructure BSS receives a Class 2 or Class 3 frame from STA B that is not authenticated with STA A (i.e., the state for STA B is State 1), STA A shall discard the frame. If the frame has a unicast address in the Address 1 field, STA A shall send a Deauthentication frame to STA B.

Authentication is optional in an IBSS. In an infrastructure BSS, authentication is required. APs do not initiate authentication.

* Authentication—originating STA

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

* If the STA is in an IBSS the SME shall delete any PTKSA and temporal keys held for communication with the originating STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10 (RSNA security association termination))
* The STA shall execute one of the following:
* For the Open System or Shared Key authentication algorithm, the authentication mechanism described in 11.2.3.2 (Open System authentication) or 11.2.3.3 (Shared Key authentication), respectively.
* For the FT authentication algorithm in an ESS, the authentication mechanism described in 12.5 (FT Protocol)., or, if resource requests are included, 12.6 (FT Resource Request Protocol)
* If the authentication was successful within the AuthenticateFailureTimeout, the state for the indicated STA shall be set to State 2 if it was State 1, the state will remain unchanged if was other than State 1.
* The MLME shall issue an MLME-AUTHENTICATE.confirm primitive to inform the SME of the result of the authentication.
* Authentication—destination STA

Upon receipt of an Authentication frame with authentication transaction sequence number equal to 1, the destination STA shall authenticate with the originating STA using the following procedure:

* If Open System or Shared Key authentication algorithm is being used, the STA shall execute the procedure described in 11.2.3.2 (Open System authentication) or 11.2.3.3 (Shared Key authentication), respectively, including issuance of an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request.
* If FT authentication is being used, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the FT Authentication Elements, and the SME shall execute the procedure as described in 12.5 (FT Protocol) or 12.6 (FT Resource Request Protocol)
* If the STA is in an IBSS and Management Frame Protection was not negotiated when the PTKSA(s) were created,, the SME shall delete any PTKSA and temporal keys held for communication with the originating STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10 (RSNA security association termination))
* Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is not SUCCESS:, the MLME shall transmit an Authentication frame with a status code, as defined in 8.4.1.9 (Status Code field), other than Successful, and the state for the originating STA shall be left unchanged.. The Authentication frame shall be constructed per the appropriate procedure in 11.2.3.2, 11.2.3.3, 12.5 or 12.6
1. Upon receipt of an MLME-AUTHENTICATE.response primitive if the ResultCode was SUCCESS:the MLME shall transmit an Authentication frame constructed per the appropriate procedure in 11.2.3.2, 11.2.3.3, 12.5 or 12.6, with a status code of Successful, and the state for the originating STA shall be set to State 2 if it was in State 1

If the STA is in an IBSS, if the SME decides to initiate an RSNA, and if the SME does not know the security policy of the peer, it may issue an individually addressed Probe Request frame to the peer by invoking an MLME-SCAN.request primitive to discover the peer’s security policy.

* Deauthentication—originating STA

The originating STA shall deauthenticate with the indicated STA using the following procedure:

* The SME shall generate an MLME-DEAUTHENTICATE.request primitive containing the appropriate reason code for the STA deauthentication, as defined in Table 8-34 (Reason codes) of 8.4.1.7 (Reason Code field).
* On receipt of the MLME-DEAUTHENTICATE.request primitive, if the state for the indicated STA is State 2, State 3, or State 4, the MLME shall transmit a Deauthentication frame to the indicated STA.
* The SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) and by invoking MLME-SETPROTECTION.request(None),
* The state for the indicated STA shall be set to State 1.
* The MLME shall issue an MLME-DEAUTHENTICATE.confirm primitive to inform the SME of the completion of the deauthentication.
* If the STA is an AP, its SME shall inform the DS of the disassociation, if the state for the indicated STA was State 3 or State 4.
* Deauthentication—destination STA

Upon receipt of a Deauthentication frame from a STA for which the state is State 2, State 3, or State 4, the destination STA shall deauthenticate with the originating STA using the following procedure:

* if Management Frame Protection was not negotiated when the PTKSA(s) were created, or if MFP is in use and the frame is not discarded per MFP processing, the MLME shall issue an MLME-DEAUTHENTICATE.indication primitive to inform the SME of the deauthentication, and set the state for the originating STA to State 1.
* Upon receiving an MLME-DEAUTHENTICATE.indication primitive, the SME shall:
1. delete any PTKSA and temporal keys held for communication with the originating STA by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) and by invoking MLME-SETPROTECTION.request(None)
2. If the STA is an AP, its SME shall inform the DS of the disassociation, if the state for the originating STA was State 3 or State 4.
* Association, reassociation, and disassociation
* General

Subclause 10.3.3 (Association, reassociation, and disassociation) describes the procedures used for IEEE 802.11 association, reassociation and disassociation.

The states used in this description are defined in 10.3.1 (General).

Successful association enables a STA to exchange Class 3 frames. Successful association sets the STA's state to State 3 or State 4.

Successful reassociation enables a STA to exchange Class 3 frames. Unsuccessful reassociation when not in State 1 leaves the STA's state unchanged (with respect to the AP that was sent the Reassociation Request (which may be the current STA)). Successful reassociation sets the STA's state to State 3 or State 4 (with respect to the AP that was sent the Reassociation Request). Successful reassociation when not in State 1 sets the STA's state to State 2 (with respect to the current AP, if this is not the AP that was sent the Reassociation Request). Reassociation shall only be performed if the originating STA is already associated in the same ESS.

Disassociation notification when not in State 1 sets the STA's state to State 2. The STA shall become associated again prior to sending Class 3 frames.

If STA A in an infrastructure BSS receives a Class 3 frame from STA B that is authenticated but not associated with STA A (i.e., the state for STA B is State 2), STA A shall discard the frame. If the frame has a unicast address in the Address 1 field, the MLME of STA A shall send a Disassociation frame to STA B.

Association is not applicable in an IBSS. In an infrastructure BSS, association is required. APs do not initiate association.

* Non-AP STA association initiation procedures

The SME shall delete any PTKSA and temporal keys held for communication with the AP by using MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) before invoking MLME-ASSOCIATE.request primitive.

Upon receipt of an MLME-ASSOCIATE.request primitive, a non-AP STA shall associate with an AP using the following procedure:

* If the state for the AP is State 1, the MLME shall inform the SME of the failure of the association by issuing an MLME-ASSOCIATE.confirm primitive, and this procedure ends.
* The MLME shall transmit an Association Request frame to the AP. If the MLME-ASSOCIATE.request primitive contained an RSN element with only one pairwise cipher suite and only one authenticated key suite, this RSN element shall be included in the Association Request frame.
* If an Association Response frame is received with a status code of Successful, the state for the AP shall be set to State 4 or State 3 if RSNA Establishment is required. The state for any other AP which is State 3 or State 4 prior to the association request shall be set to State 2, and the MLME shall issue an MLME-ASSOCIATE.confirm primitive to inform the SME of the successful completion of the association.
* If an Association Response frame is received with a status code other than Successful or the AssociateFailureTimeout expires the state for the AP shall be set to State 2, and the MLME shall issue an MLME-ASSOCIATE.confirm primitive to inform the SME of the failure of the association. The status code returned in the Association Response frame indicates the cause of the failed association attempt. Any misconfiguration or parameter mismatch, e.g., data rates required as basic rates that the STA did not indicate as supported in the STA’s Supported Rates element, shall be corrected before the SME issues an MLME-ASSOCIATE.request primitive for the same AP. If the status code indicates the association failed because of a reason that is not related to configuration, e.g., the AP is unable to support additional associations, the SME shall not issue an MLME-ASSOCIATE.request primitive for the same AP until a period of at least 2 s has elapsed.
* If an MLME-ASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and RSNA is required, then the SME shall perform a 4-way handshake to establish an RSNA. As a part of a successful 4-way handshake, the SME shall enable protection by invoking MLME-SETPROTECTION.request(Rx\_Tx) Upon receipt of the MLME-SETPROTECTION.request, the MLME shall set the state of the STA to State 4.
* AP association receipt procedures

Upon receipt of an Association Request frame from a non-AP STA for which the state is State 1, the AP shall transmit an Association Response frame with an appropriate status code.

Upon receipt of an Association Request frame from a non-AP STA for which the state is State 2, State 3, or State 4, the AP’s MLME shall associate with the non-AP STA using the following procedure:

* The MLME shall issue an MLME-ASSOCIATE.indication primitive to inform the SME of the association request.
* Upon receiving an MLME-ASSOCIATE.indication primitive, when Management Frame Protection is not in use, the SME shall delete any PTKSA and temporal keys held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination))
* In an RSNA, the AP shall check the values received in the RSN element to see whether the values received match the AP’s security policy. If not, the association shall not be accepted.
* If the AP’s state for the non-AP STA is 4, and the AP has a valid security association for the non-AP STA, and has negotiated Management Frame Protection with the non-AP STA,
1. the SME shall reject the Association Request by generating an MLME-ASSOCIATE.response primitive with ResultCode “Association request rejected temporarily; try again later”.

***Editor: add “***Association request rejected temporarily; try again later” ***to 6.3.7.5.2***

1. The SME shall not modify any association state for the non-AP STA, and shall include in the MLME-ASSOCIATE.response primitive a Timeout Interval element with Timeout interval type set to 3 (Association Comeback time), specifying a comeback time when the AP would be ready to accept an association with this STA.

***Editor: add Timeout Interval Eleement parameter*** ***to 6.3.7.5.2***

1. Following this, if the SME is not already engaging in an SA Query with the STA, the SME shall issue one MLME-SAQuery.request primitive addressed to the STA every dot11AssociationSAQueryRetryTimeout TUs until a matching MLME-SAQuery.confirm primitive is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure have passed.
2. The SME shall specify a TransactionIdentifier parameter value in the MLME-SAQuery.request primitive, and increment the value by 1 for each subsequent MLME-SAQuery.request primitive, rolling over the value to 0 after the maximum allowed value is reached.
3. The MLME may interpret reception of a valid protected frame as an indication of a successfully completed SA Query, and may generate an MLME-SAQuery.confirm primitive
4. If an MLME-SAQuery.confirm primitive with an outstanding transaction identifier is not received within dot11AssociationSAQueryMaximumTimeout period, the SME shall allow the association process to be started without starting an additional SA Query procedure
* The SME shall refuse an association request from a STA that does not support all the rates in the BSSBasicRateSet parameter
* The SME shall refuse an association request from an HT STA that does not support all the MCSs in the BSSBasicMCSSet parameter.
* The SME shall generate an MLME-ASSOCIATE.response primitive addressed to the non-AP STA. When the association is not successful, the AP shall indicate a specific reason for the failure to associate in the ResultCode parameter as defined in 6.3.7.5.2. If the ResultCode is SUCCESS, the association identifier assigned to the STA shall be included in this primitive., and the SME shall delete any PTKSA and temporal keys held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) .
* Upon receipt of an MLME-ASSOCIATE.response primitive, the MLME shall transmit an Association Response frame) to the STA.
* When the ResultCode of the MLME-ASSOCIATE.response primitive is not SUCCESS, if Management Frame Protection is in use the state for the STA shall be left unchanged. and if Management Frame Protection is not in use set to State 3 if it was in State 4.
* When the Association Response frame with a status code of Successful is acknowledged by the STA, the state for the STA shall be set to State 4 or State 3 if RSNA establishment is required.
* If RSNA establishment is required, the SME shall attempt a 4-way handshake. Upon a successful completion of a 4-way handshake,, the SME shall enable protection by invoking MLME-SETPROTECTION.request(Rx\_Tx). Upon receipt of the MLME-SETPROTECTION.request, the MLME shall set the state for the STA to State 4.
* The SME shall inform the DS of any changes in the association state.

In case of a failed SA Query procedure, if the SME receives an MLME-ASSOCIATE.request primitive from a STA with which it has an existing SA, then the SME shall send a MLME-DISASSOCIATE.request primitive to this STA prior to terminating the old SA, with Reason Code “Previous Authentication no longer valid”

NOTE—this MLME-DISASSOCIATE.request generates a protected Disassociation frame addressed to this STA.

* Non-AP STA reassociation initiation procedures

Except when the association is part of a fast BSS transition, the SME shall delete any PTKSA and temporal keys held for communication with the AP by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) before invoking an MLME-REASSOCIATE.request primitive.

Upon receipt of an MLME-REASSOCIATE.request primitive, a non-AP STA shall reassociate with an AP using the following procedure:

* If the STA is not associated in the same ESS or the state for the new AP is State 1, the MLME shall inform the SME of the failure of the reassociation by issuing an MLME-REASSOCIATE.confirm primitive, and this procedure ends.
* The MLME shall transmit a Reassociation Request frame to the new AP. If the MLME-REASSOCIATE.request primitive contained an RSN element with only one pairwise cipher suite and only one authenticated key suite, this RSN element shall be included in the Reassociation Request frame.
* If a Reassociation Response frame is received with a status code of Successful, the state variable for the new AP shall be set to State 4, or to State 3 if RSNA establishment is required and the FT Protocol is not used with respect to the new AP and, unless the old AP and new AP are the same, to State 2 with respect to the old AP, and the MLME shall issue an MLME-REASSOCIATE.confirm primitive to inform the SME of the successful completion of the reassociation.
* If a Reassociation Response frame is received with a status code other than Successful or the ReassociateFailureTimeout expires:
	+ Except when the association is part of a fast BSS transition, the state for the AP shall be set to State 2 with respect to the new AP
	+ the MLME shall issue an MLME-REASSOCIATE.confirm primitive to inform the SME of the failure of the reassociation. The ResultCode returned in the MLME-REASSOCIATE.confirm primitive indicates the cause of the failed reassociation attempt. Any misconfiguration or parameter mismatch, e.g., data rates required as basic rates that the STA did not indicate as supported in the STA’s Supported Rates element, shall be corrected before the SME issues an MLME-REASSOCIATE.request primitive for the same AP. If the status code indicates the reassociation failed because of a reason that is not related to configuration, e.g., the AP is unable to support additional associations, the SME shall not issue an MLME-REASSOCIATE.request primitive for the same AP until a period of at least 2 s has elapsed.
* If an MLME-REASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and RSNA is required, then the SME shall perform a 4-way handshake to establish an RSNA. As a part of a successful 4-way handshake, the SME shall enable protection by invoking MLME-SETPROTECTION.request(Rx\_Tx). Upon receipt of the MLME-SETPROTECTION.request, the MLME shall setthe state of the STA to State 4.
* AP reassociation receipt procedures

Upon receipt of an Reassociation Request frame from a non-AP STA for which the state is State 1, the MLME shall transmit an Reassociation Response frame with an appropriate status code

Upon receipt of a Reassociation Request frame from a STA for which the state is State 2, State 3, or State 4, the AP’s MLME shall reassociate with the STA using the following procedure:

* The MLME shall issue an MLME-REASSOCIATE.indication primitive to inform the SME of the reassociation request.
* In an RSNA, the SME shall check the values received in the RSN element to see whether the values received match the AP’s security policy. If not, the association shall not be accepted.
* If the AP’s state for the non-AP STA is 4 and the non-AP STA has a valid security association, the non-AP STA has negotiated management frame protection, and the reassociation is not a part of a Fast BSS Transition:
1. The SME shall reject the Reassociation Request by generating an MLME-REASSOCIATE.response primitive with ResultCode“Association request rejected temporarily; Try again later”.

***Editor: add “***Association request rejected temporarily; try again later” ***to 6.3.7.5.2***

1. The SME shall not modify any association state for the non-AP STA, and shall include in the MLME-REASSOCIATE.response primitive a Timeout Interval element with type set to 3 (Association Comeback time), specifying a comeback time when the AP would be ready to accept an association with this STA.

***Editor: add Timeout Interval Eleement parameter*** ***to 6.3.7.5.2***

1. Following this, if the SME is not in an ongoing SA Query with the STA, the SME shall issue one MLME-SAQuery.request primitive addressed to the STA every dot11AssociationSAQueryRetryTimeout TUs until a matching MLME-SAQuery.confirm primitive is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure have passed.
2. The SME shall insert the TransactionIdentifier in MLME-SAQuery.request primitive, and increment this by 1 for each subsequent MLME-SAQuery.request primitive, and rolling over to 0 after the maximum allowed value in this field.
3. An MLME may interpret reception of a valid protected frame as an indication of a successfully completed SA Query, and may generate an MLME-SAQuery.confirm primitive.
4. If an MLME-SAQuery.confirm primitive with an outstanding transaction identifier is not received within dot11AssociationSAQueryMaximumTimeout period, the SME shall allow the association process to be started without starting additional SA Query procedure.
* The SME shall refuse a reassociation request from a STA that does not support all the rates in the BSSBasicRateSet parameter.
* The SME shall refuse a reassociation request from an HT STA that does not support all the MCSs in the BSSBasicMCSSet parameter.
* The SME shall generate an MLME-REASSOCIATE.response primitive addressed to the non-AP STA. When the reassociation is not successful, the AP shall indicate a specific reason for the failure to reassociate in the ResultCode parameter as defined in 6.3.7.5.2

 If the ResultCode is SUCCESS, the association identifier assigned to the STA shall be included in this primitive.. Except when the association is part of a fast BSS transition or Management Frame Protection is in use,, the SME shall delete any PTKSA and temporal keys held for communication with the STA by using MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination))

* Upon receiving a MLME-REASSOCIATE.indication primitive Upon receipt of an MLME-REASSOCIATE.response primitive, the MLME shall transmit a Reassociation Response frame to the STA.
* When the Reassociation Response frame with a status value of Successful is acknowledged by the STA, the state variable for the STA shall be set to State 4, or to State 3 State 3 if RSNA establishment is required on the new AP and the FT Protocol is not used on the new AP.
* When the ResultCode of the reassociation is not SUCCESS, if Management Frame Protection is in use the state for the STA shall be left unchanged on the AP the Reassociation Request frame was sent to, When the ResultCode is not SUCCESS and Management Frame Protection is not in use and the association is not part of a fast BSS transition, the state for the STA is set to State 3 if it was in State 4.
* If RSNA establishment is required and FT is not in use, the SME shall attempt a 4-way handshake. , Upon a successful completion of a 4-way handshake,the SME shall enable protection by invoking MLME-SETPROTECTION.request(Rx\_Tx). Upon receipt of the MLME-SETPROTECTION.request, the MLME shall set the state for the STA to State 4.
* The SME shall inform the DS of any changes in the association state.

n case of a failed SA Query procedure, if the SME receives an MLME-REASSOCIATE.request from a STA with which it has an existing SA, then the SME shall send a MLME-DISASSOCIATE.request primitive to this STA prior to terminating the old SA, with Reason Code “Previous Authentication no longer valid”

NOTE—this MLME-DISASSOCIATE.request generates a protected Disassociation frame addressed to this STA

* Non-AP STA disassociation initiation procedures

The SME shall issue an MLME-DISASSOCIATE.request primitivethat includes an appropriate Reason Code as defined in Table 8-34 (Reason codes) of 8.4.1.7 (Reason Code field).

Upon receipt of an MLME-DISASSOCIATE.request primitive, a non-AP STA’s MLME shall disassociate from an AP using the following procedure:

* If the state for the AP is State 3 or State 4, the MLME shall transmit a Disassociation frame to the AP.
* The state for the AP shall be set to State 2 if it was not State 1.
* The MLME shall issue an MLME-DISASSOCIATE.confirm primitive to inform the SME of the successful completion of the disassociation.
* Upon receiving a MLME-DISASSOCIATE.confirm primitive, the SME shall delete any PTKSA and temporal keys held for communication with the AP by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) and by invoking MLME-SETPROTECTION.request(None).
* Non-AP STA disassociation receipt procedure

Upon receipt of a Disassociation frame from an AP for which the state is State 3 or State 4, if Management Frame Protection was not negotiated when the PTKSA(s) were created, or if MFP is in use and the frame is not discarded per MFP processing, a non-AP STA shall disassociate from the AP using the following procedure

* The state for the AP shall be set to State 2.
* The MLME shall issue an MLME-DISASSOCIATE.indication primitive to inform the SME of the disassociation.
* Upon receiving the MLME-DISASSOCIATE.indication primitive, the SME shall delete any PTKSA and temporal keys held for communication with the AP by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) and by invoking MLME-SETPROTECTION.request(None).
* If the reason code indicates a configuration or parameter mismatch as the cause of the disassociation, the SME shall not attempt to associate or reassociate with the AP until the configuration or parameter mismatch has been corrected.
* If the reason code indicates the STA was disassociated for a reason other than configuration or parameter mismatch, the SME shall not attempt to associate or reassociate with the AP until a period of 2 s has elapsed.
* AP disassociation initiation procedure

The SME shall issue an MLME-DISASSOCIATE.request primitive that includes an appropriate Reason Code as defined Table 8-34 (Reason codes) of 8.4.1.7 (Reason Code field).

Upon receipt of an MLME-DISASSOCIATE.request primitive, an AP shall disassociate a STA using the following procedure:

* If the state for the STA is State 3 or State 4, the AP shall send a Disassociation frame to the STA.
* The state for the STA shall be set to State 2, if it was not State 1.
* The MLME shall issue an MLME-DISASSOCIATE.confirm primitive to inform the SME of the disassociation.
* Upon receiving a MLME-DISASSOCIATE.confirm primitive, the SME shall delete any PTKSA and temporal keys held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) and by invoking MLME-SETPROTECTION.request(None).
* The SME shall inform the DS of the disassociation.
* AP disassociation receipt procedure

Upon receipt of a Disassociation frame from a STA for which the state is State 3 or State 4, if Management Frame Protection was not negotiated when the PTKSA(s) were created, or if MFP is in use and the frame is not discarded per MFP processing, the AP shall disassociate the STA using the following procedure:

* The state for the STA shall be set to State 2.
* The MLME shall issue an MLME-DISASSOCIATE.indication primitive to inform the SME of the disassociation.
* Upon receiving a MLME-DISASSOCIATE.indication primitive the SME shall delete any PTKSA and temporal keys held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 11.4.12 (RSNA security association termination)) and by invoking MLME-SETPROTECTION.request(None).
* The SME shall inform the DS of the disassociation.