IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LB202 MAH Assigned comments | | | | |
| Date: 2014-11-5 | | | | |
| Author(s): | | | | |
| Name | Company | Address | Phone | email |
| Mark Hamilton | Spectralink | 2560 55th St  Boulder, CO 80301 USA | +1 303 441 7553 | [mark.hamilton@spectralink.com](mailto:mark.hamilton@spectralink.com) |
|  |  |  |  |  |

Abstract

This submission contains proposed comment resolutions for CIDs:

R0 – 3150, 3507, 3502, 3144, 3521, 3145, 3146, 3520, 3061, 3062, 3500, 3063, 3523, 3524, 3068, 3128, 3129, 3131, 3391, 3285, 3211, 3519, 3132, 3506.

R1 – Minor updates per discussion, and green highlight of accepted resolutions, from San Antonio face-to-face

R2 – Minor change to CID 3524, based on face-to-face discussion, change to CID 3391 based on feedback.

R3 – Minor changes to CID 3128, et al, and additional change to CID 3391 based on feedback.

R4 – Updates per discussion at face-to-face session, green highlight of accepted resolutions

# CID 3150 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3150 | 98.23 | 4.5.2.1 | Figure B.6 (IEEE Std 802.11 infrastructure model) of IEEE P802 D2.0 is more adequate to describe distribution of MSDUs within a DS.. | Insert the modified Figure B.6 of IEEE P802 D2.0 in the subclause 4.5.2.1, and replace the reference to Figure 4-14 by the reference to the new figure. |

**Discussion:**

The context is:

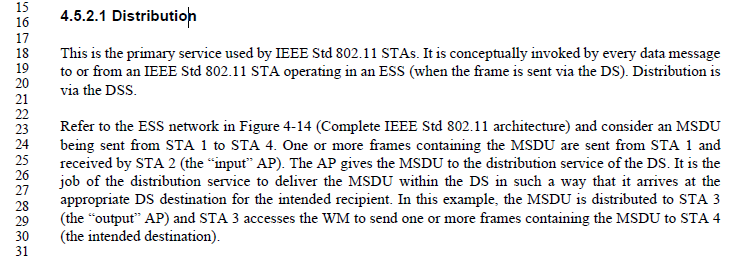


Figure 4-14 is:

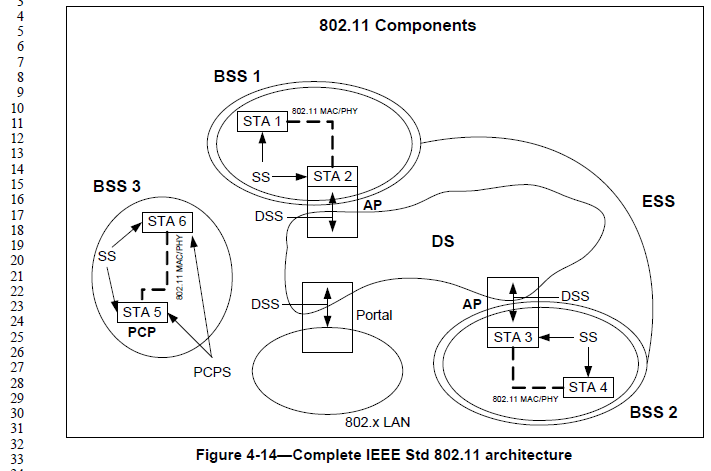
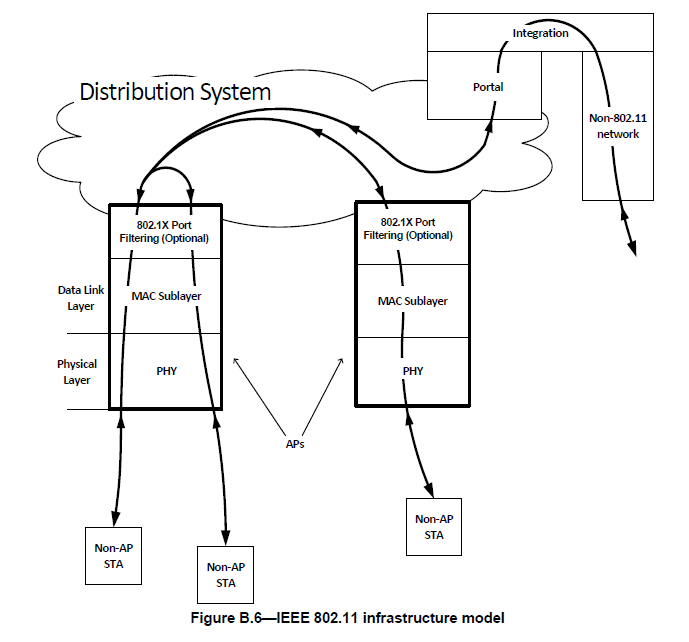


Figure B.6 (from 802-2014):



Presumably, this version of Figure B.6 (from the approved 2014 802 Std) is the one refrenced by the commenter as “the modified Figure B.6 of IEEE P802 D2.0”.

It does seem that Figure B.6 shows the data paths, which is the subject of this paragraph, more clearly than Figure 4-14, so generally the comment is agreed. However, the text also needs to reference specific STAs and APs by name, to discuss the data flow explicitly. So, Figure B.6 needs to be enhanced to add STA and AP identifiers. This is shown/suggested in the figure below.

**Proposed resolution: Revised**

Insert the figure shown in <this document> in the Proposed Resolution to CID 3150, as new Figure 4-14a, near the second paragraph of 4.6.2.1. Change the first sentence of 4.6.2.1 second paragraph to start, “Refer to the ESS network in Figure 4-14a (IEEE 802.11 Infrastructure model) …”

*New Figure 4-14a:*



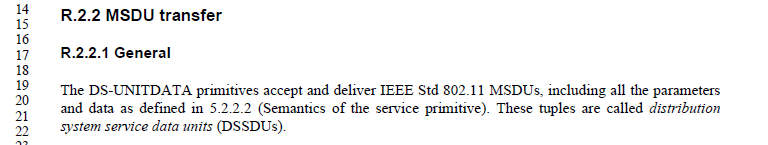
**Figure 4-14a – IEEE 802.11 Infrastructure model**

# CID 3507 and 3506 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3507 | 126.01 | 5 | The concept of a tuple of MSDU and all its associated parameters, currenlty only used in Annex R (see R.2.2.1, called a DSSDU), is probably useful to describe what information "goop" gets handled as bundle inside the MAC stack, queuing, etc. Make the term more generic (not DS-centric) and use it in clause 5. | Needs submission to generalize this DS-centric concept, and add it to appropriate places in clause 5. |
| 3506 | 3523.42 | R.2.2.2.2 | Actually, the type of the DSSDU distributed by the DS is "DSSDU" which is defined above as a "tuple of MSDU and all parameters" (as described in the UNITDATA primitives in 5.2.2.2). | Change "IEEE Std 802.11 MSDU" to "Tuple of IEEE 802.11 MSDU and all parameters". Same change in R.2.2.3.2. Text in 4.5.2 should refer to DSSDUs not MSDUs as the unit of information that is distributed. Change MSDU to DSSDU there. (Probably means moving the definition to somewhere more normative, too?) |

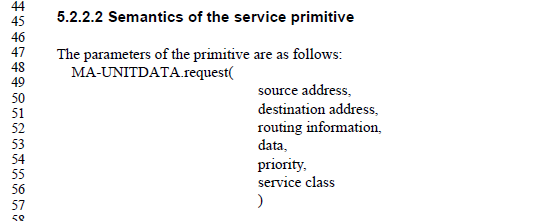
**Discussion:**

The reference from Annex R is:



Note that this discusses DSSDUs as “MSDUs, including all the parameters and data (sic) as defined in 5.2.2.2.” That is, a DSSDU is actually more than the MSDU, but is meant to be all the parameters that cross the MAC Service primitives.

From 5.2.2.2, that list of parameters is:



Thus, the intention is that the tuple of information that traverses the DS is the above collection of information, including the “data” (MSDU), and other information needed to deliver the MSDU to the correct “output”point of the distribution system, and to provide the other parameters needed by the MAC Service at that output point. Note additionally that 802.11 specifically designates the routine information to be null (and not used).

Thus, this is specifically the tuple <source address, destination address, msdu, priority, service class>.

Per discussion on a teleconference, it was agreed that using the acronym “SDU” within the name of this tuple in any form would be confusing and misleading, since this is more information than a “data unit”. “MAC Service tuple” was suggested, which seems like as good a suggestion as any.

**Proposed resolution: Revised**

*Add a definition to 3.1:*

**medium access control (MAC) service tuple:** The collection of an MSDU along with the source address, destination addresses, priority and service class associated with the MSDU, which are passed as parameters across the MAC SAP, and are delivered across the distribution system between access points (APs), mesh gates, and portals of an extended service set (ESS).

*Change the following definitions as shown:*

**distribution service:** The service that, by using association information, delivers medium access control (MAC) service ~~data units (MSDUs)~~ tuples within the distribution system (DS).

**distribution system service (DSS):** The set of services provided by the distribution system (DS) that enable the medium access control (MAC) to transport MAC service ~~data units (MSDUs)~~ tuples between stations (STAs) that are not in direct communication with each other over a single instance of the wireless medium (WM).

NOTE 4—These services include transport of ~~MSDUs~~MAC service tuples between the access points (APs) of basic service sets (BSSs) within an extended service set (ESS), transport of ~~MSDUs~~MAC service tuples between portals and BSSs within an ESS, transport of ~~MSDUs~~MAC service tuples between mesh gates in the same or different mesh basic service sets (MBSSs), transport of ~~MSDUs~~MAC service tuples between mesh gates and APs, transport of ~~MSDUs~~MAC service tuples between mesh gates and portals, and transport of ~~MSDUs~~MAC service tuples between STAs in the same BSS in cases where the ~~MSDUs~~MAC service tuples has a group destination address or where the destination is an individual address and the STA is associated with an AP.

*In the second and following paragraphs of 4.5.2.1, make changes as shown:*

Refer to the ESS network in Figure 4-14 (Complete IEEE Std 802.11 architecture) and consider an MSDU being sent from STA 1 to STA 4. One or more frames containing the MSDU are sent from STA 1 and received by STA 2 (the “input” AP). The AP gives a MAC service tuple containing the MSDU to the distribution service of the DS. It is the job of the distribution service to deliver the ~~MSDU~~ MAC service tuple within the DS in such a way that it arrives at the appropriate DS destination for the intended recipient. In this example, the ~~MSDU~~ MAC service tuple is distributed to STA 3 (the “output” AP) and STA 3 accesses the WM to send one or more frames containing the MSDU to STA 4 (the intended destination).

How the ~~MSDU~~ MAC service tuple is distributed within the DS is not specified by IEEE Std 802.11. All IEEE Std 802.11 is required to do is to provide the DS with enough information for the DS to be able to determine the “output” point that corresponds to the intended recipient. The necessary information is provided to the DS by the three association related services (association, reassociation, and disassociation).

The previous example was a case in which the AP that invoked the distribution service was different from the AP that received the distributed ~~MSDU~~ MAC service tuple. If the MSDU had been intended for a STA that was a member of the same BSS as the sending STA, then the “input” and “output” APs for the ~~MSDU~~ MAC service tuple would have been the same.

In either example, the distribution service was logically invoked. Whether the ~~MSDU~~ MAC service tuple actually had to traverse the physical DSM or not is a DS implementation matter and is not specified by this standard.

*Make the following changes in clause 4.5.2.2 (Integration):*

If the distribution service determines that the intended recipient of an MSDU is a member of an integrated LAN, the “output” point of the DS would be a portal instead of an AP.

~~MSDUs~~ MAC service tuples that are distributed to a portal cause the DS to invoke the Integration function (conceptually after the distribution service). The Integration function is responsible for accomplishing whatever is needed to deliver an ~~MSDU~~ MAC service tuple from the DSM to the integrated LAN media (including any required media or address space translations). Integration is one of the services in the DSS.

MSDUs received from an integrated LAN (via a portal) by the DS for an IEEE Std 802.11 STA invoke the Integration function before the ~~MSDU~~ MAC service tuple is distributed by the distribution service.

*Add a paragraph at the end of 5.1.3 (MSDU ordering):*

When MSDU or A-MSDU reordering is performed, the information in the MAC service tuple(s) for the MSDU(s) shall be maintained and reordered as a unit.

*Add a paragraph, to become the only text body of 10.2 (Power management):*

When MSDUs or A-MSDUs are buffered for power management purposes, the information in the MAC service tuple(s) for the MSDU(s) shall be maintained (and reordered) as a unit.

*In Annex R, make changes as shown:*

The DS SAP interface specification describes the primitives required to get ~~MSDUs~~ MAC service tuples in and out of the DS and update the DS’s mapping of STAs to APs or to mesh gates. Describing the DS itself or the functions thereof is out of scope of this annex.

The DS SAP actions are as follows:

a) Accept MSDUs (as part of MAC service tuples) from APs, mesh gates, and portals.

b) Deliver MSDUs (as part of MAC service tuples) to APs, mesh gates, or portals.

c) Accept STA-to-AP mapping updates from the APs.

d) Accept STA-to-mesh gate mapping updates from the mesh gates.

When the DS delivers the ~~MSDUs~~ MAC service tuples to an AP, the AP then determines when and how to deliver the ~~MSDUs~~ MAC service tuples to the AP’s MAC (via the MAC SAP). When the DS delivers the ~~MSDUs~~ MAC service tuples to a mesh gate, the mesh gate then determines when and how to deliver the ~~MSDUs~~ MAC service tuples to the mesh gate’s MAC (via the MAC SAP).

*In R.2.2.1, make the following changes:*

The DS-UNITDATA primitives accept and deliver IEEE Std 802.11 ~~MSDUs~~ MAC service tuples, including an MSDU and all the parameters as defined in 5.2.2.2 (Semantics of the service primitive). ~~These tuples are called~~ *~~distribution system service data units~~* ~~(DSSDUs).~~

*Replace all occurrences of* “DSSDU” *in Annex R with* “MAC service tuple”.

*Delete* “DSSDU” *from 3.4 (Abbreviations and acronyms).*

# CID 3502 (MAC)

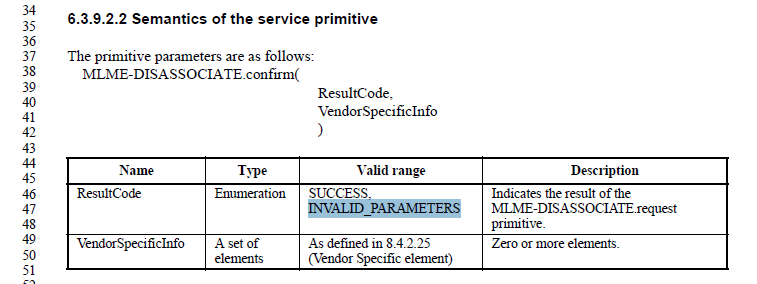
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3502 | 196.47 | 6.3.9.2.2 | INVALID\_PARAMETERS is implementation behavior, not interoperability. | Remove all INVALID\_PARAMETERS values for Result Codes or Status Codes |

**Discussion:**

Consider each usage on a case-by-case basis:

***Case #1***

On page 196:

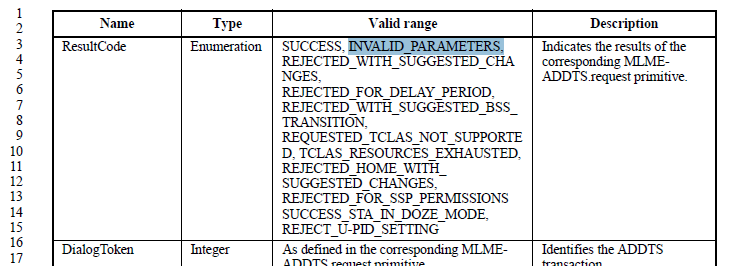
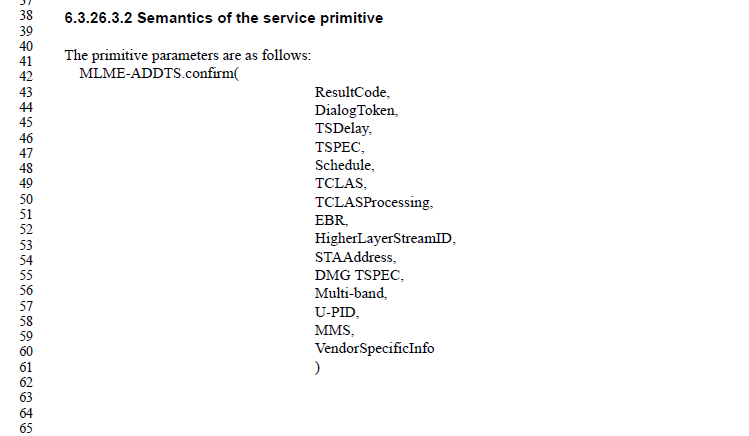


This usage in the Disassociate primitive is not supported by any discussion in text or frame formats. Thus, this is purely a local behavior with no affect on interoperability or externally visible behavior. As such, this usage should be removed from the Standard.

Note that by removing this ResultCode, the ResultCode parameter becomes moot (only one value is possible), so it should be removed. The VendorSpecificInfo is not a viable parameter as it is, anyway, since there is no protocol supporting the .confirm primitive – it is generated locally by the local MLME. So, no VendorSpecificInfo is available to be provided here. Thus, that parameter is vacuous. The MLME-DISASSOCIATE.confirm primitive is needed, per 10.3.5.6, presumably so the MLME can synchronize timing of any pending data transfers before the PTKSA is deleted. (A tenuous proposition, but arguably possible.) So, we leave the primitive in place, but with no parameters.

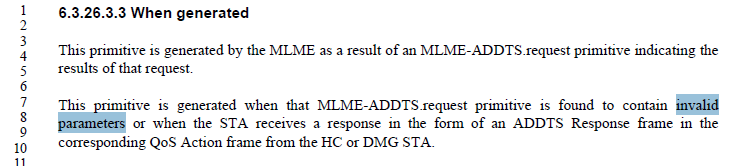
***Case #2***

On page 228-229:

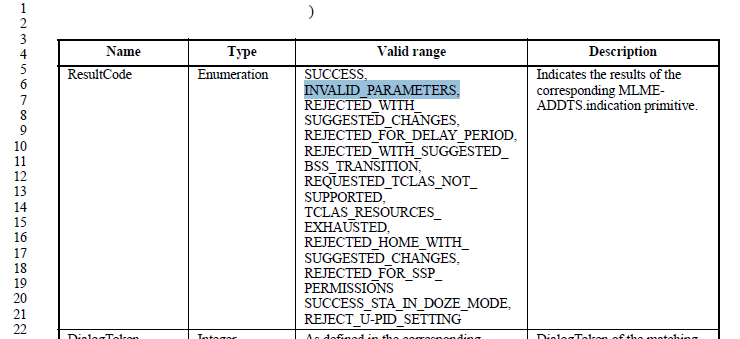
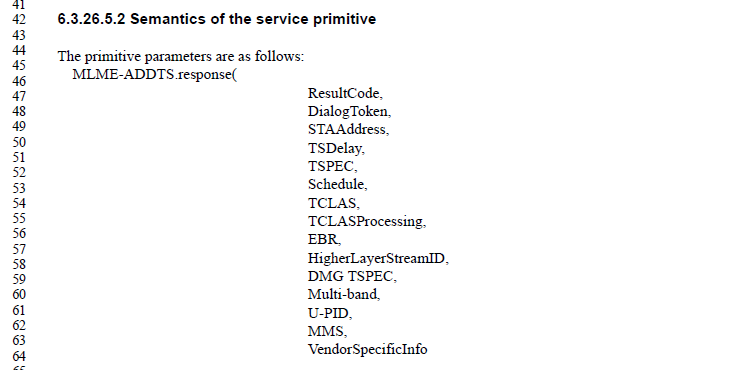


This usage in the ADDTS service is mentioned in the text, and is supported by the parameter values in the ADDTS response primitive, as shown below.

On page 231:



On pages 233-234:



So there is clearly some anticipation that the peer STA’s SME can detect some parameters of the request to be invalid, and return a response with “INVALID\_PARAMETERS” carried in the ResultCode. However, there is no text mentioning this behavior, or providing any guidance for when this would/should happen.

When we consider the parameters passed in to the MLME-ADDTS.request, there are many, and many of them are structures of information or references to external identifiers, so it can be imagined that they could be specified in invalid ways. The question is whether the Standard needs to cover the case of a requester generating an invalid request, or whether it should discuss the peer STA being part of the detection when this happens.

The strongest argument for this situation being explicit in the Standard is when the STA generating the request may not have the information to know what the peer STA will consider a valid request. However, these situations are handled in many places with specific result codes that anticipate that this can happen and provide a specific response from the peer for identified, possible invalid request parameters.

This seems to be true in this case, as well, as evidenced by the other, more specific Result Codes which are listed. If there are any remaining situations not covered, those can be added, rather than falling back on the vague (and therefore not very helpful to implementations) INVALID\_PARAMETERS result.

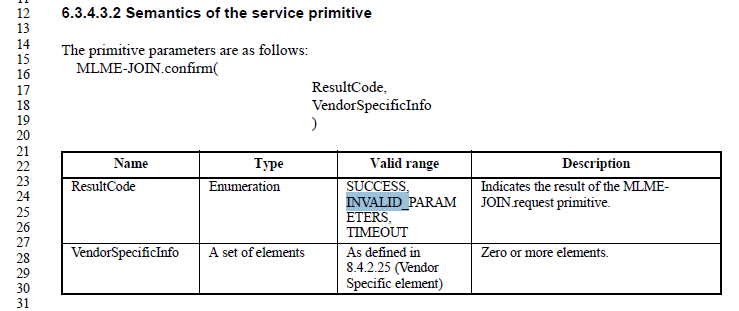
***Case #3***

On pages 251 and 254, the ADDBA confirm and ADDBA response primitives each list INVALID\_PARAMETERS as a possible value for the ResultCode. There is no other mention of this value being use in the text. In fact, the text specifically talks about both responding with altered values for some parameters to negotiate a particular setting, or responding with a rejection of the request if no suitable match of settings can be found. Given the description of the parameters that are passed in to the request, there does not appear to be any legal way to construct a request that has parameters that the peer would find to be invalid.

Thus, the recommendation is to delete the INVALID\_PARAMETERS option from the ADDBA confirm and ADDBA response primitives’ “Valid range”.

***Case #4***

On page159:



The MLME-JOIN service is local to the synchronizing STA, there is no protocol, nor remote peer which may have additional information to decide some parameters are invalid. Thus, this is a local matter, and this ResultCode is not needed within the Standard.

Thus the recommendation is to delete the INVALID\_PARAMETERS option from the ValidRange for this primitive’s ResultCode.

***Case #5***

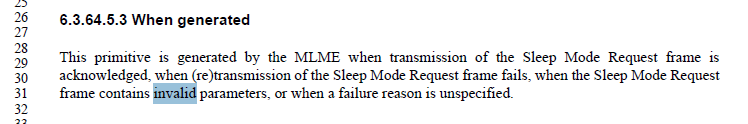
On pages 282 and 284, the MLME-ENABLEMENT confirm and MLME-ENABLEMENT response primitives each list INVALID\_PARAMETERS as a possible value for the ResultCode. This is also mentioned in the text in clause 10, stating that if any parameter in the .request is invalid, the MLME should (locally, and immediately) issue a .confirm with the INVALID\_PARAMETERS ResultCode. However, there is no parameter to the .request that could be invalid from a conforming implementation.

There is the possibility that the peer STA detects an invalid parameter, and responds with INVALID\_PARAMETERS in the .response, and the protocol indicates this value can be carried in the response frame. However, again, there are no parameters to the .request, or therefore supplied to the peer, that could be invalid, if the initiating SME is a conforming implementation. So, again, this is a local problem.

Thus, the recommendation is to delete the INVALID\_PARAMETERS option from the MLME-ENABLEMENT confirm and MLME-ENABLEMENT response primitives’ “Valid range”, and to delete the local invalid parameters check step from the procedure in 10.12.2.2.

***Case #6***

On page 366:



Yet, there is no parameter to the .confirm that would indicate an invalid parameter failure response, other than embedded with one of the parameters which is formatted as an element (per clause 8). However, none of these elements has a status field which can indicate a concept of “invalid parameters”. So, this is not only another example of local behavior.

Thus the recommendation is to delete this concept from the “When generated” clause.

Further, the rest of the text in the “When generated” clause is unusual, and doesn’t really make sense. There is a response frame for the Sleep Mode service, and the frame exchange shown in Figure 6-22 and discussed in all the other primitives’ discussion describes the frame exchange as the typical request/response behavior. So, the text in 6.3.64.5.3 only makes sense to follow that same general pattern.

Thus, the recommendation is to replace the text of 6.3.64.5.3 with the typical pattern for a request/response frame exchange service.

***Case #7***

On page 369, the TIM Broadcast setup exchange has exactly the same problem in its .confirm primitive as the Sleep Mode, above. In the case of TIM Broadcast, again there is a Status field embedded within an element format which is passed as a parameter, and again, that Status field has no value equivalent to “invalid parameters”.

Further, the TIM Broadcast .confirm primitive’s “When generated” clause (6.3.65.3.3) has nearly the exact text to that in the Sleep Mode .confirm. (Probably a cut-and-paste when this feature was added.)

Thus, the recommendation is the same as for the Sleep Mode primitive, to replace the entire text of the “When Generated” clause with typical request/response text.

***Case #8***

On page 401, the MESHPEERINGMANAGEMENT response primitive shows a ResultCode parameter, which has Valid Range of SUCCESS or INVALID\_PARAMETERS. There is no text mentioning how or when this is used. Further, there is no ResultCode in the .confirm primitive, nor any indication of carrying this result in the frame formats supporting this service.

Thus the recommendation is to delete the ResultCode parameter from the MESHPEERINGMANAGMENT.response primitive.

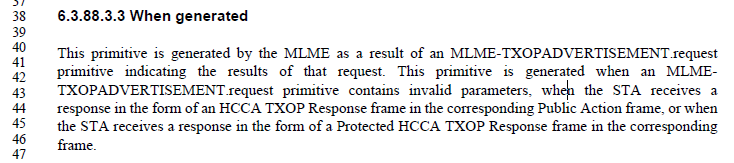
***Case #9***

On page2 412 and 413, the MCCA Setup service has the same pattern as the ADDBA service, discussed above (Case #3).

Thus, the recommendation is to delete the INVALID\_PARAMETERS option from the MCCASETUP confirm and MCCASETUP response primitives’ “Valid range”.

***Case #10***

On page448:

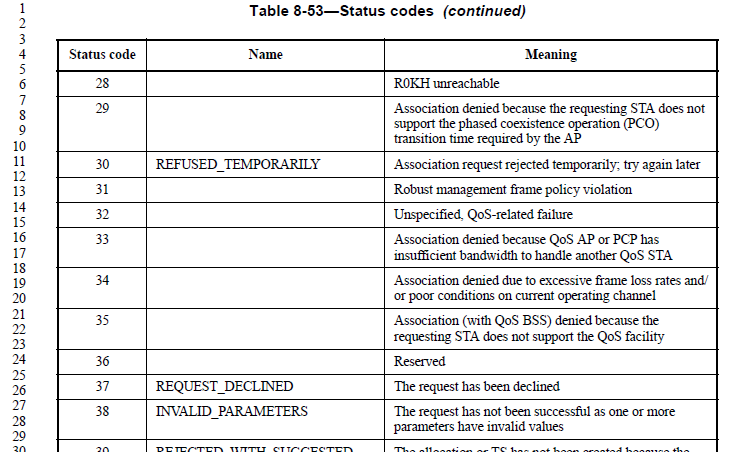


The “When generated” text including the mention of the .request having “invalid parameters” as a trigger for the .confirm implies the local detection of the invalid parameters, not anything from the peer STA and protocol exchange.

Thus the recommendation is to delete this concept from the “When generated” clause.

***Case #11***

On page 648:



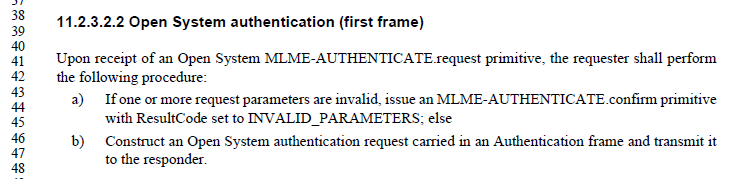
Given the changes above, it is not clear that there is any primitive invocation or other behavior which will generate this Status code in any protocol. However, it is easiest and safest to just leave this entry in the table (the number would have to be reserved, anyway).

Thus, the recommendation is to do nothing with this case.

Likewise, the same recommendation for the occurrence in Tables 8-296 and 8-298.

***Case #12***

On page 1851:



However, there is no support in the service primitives for the .confirm described in bullet (a), and this is another example of a local matter.

Thus, the recommendation is to delete bullet (a) from 11.2.3.2.2.

Exactly the same thing in 11.2.3.3.2.

The above cases are all summarized in the changes below, as the actual proposed resolution to this CID.

**Proposed resolution: Revised**

*In 6.3.9.2.2, delete both the ResultCode and VendorSpecificInfo parameters and delete the parameter table.*

*Delete* INVALID\_PARAMETERS *from the ResultCode list in 6.3.4.3.2, 6.3.26.3.2, 6.3.26.5.2, 6.3.29.3.2, 6.3.29.5.2, 6.3.39.3.2, 6.3.39.5.2.*

*Delete “*when that MLME-ADDTS.request primitive is found to contain invalid parameters or*” from 6.3.26.3.3.*

*Replace the text of 6.3.64.5.3 with, “*This primitive is generated by the MLME when the STA receives a Sleep Mode Response frame from the AP.”

*Replace the text of 6.3.65.3.3 with, “*This primitive is generated by the MLME when the STA receives a TIM Broadcast Response frame from the AP.”

*Delete “*when an MLMETXOPADVERTISEMENT.request primitive contains invalid parameters,*” from 6.3.88.3.3.*

*Delete bullet (a) from the list in 10.12.2.2.*

*Delete bullet (a) from the list in 11.2.3.2.2. Merge (b) into the paragraph text.*

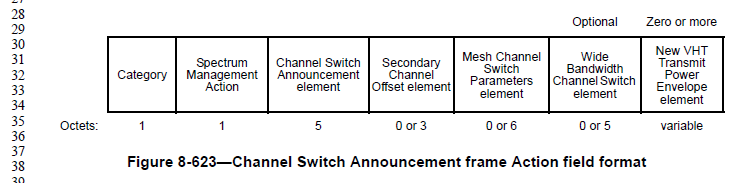
*Delete bullet (a) from the list in 11.2.3.3.2. Merge (b) into the paragraph text.*

# CID 3144 (MAC)

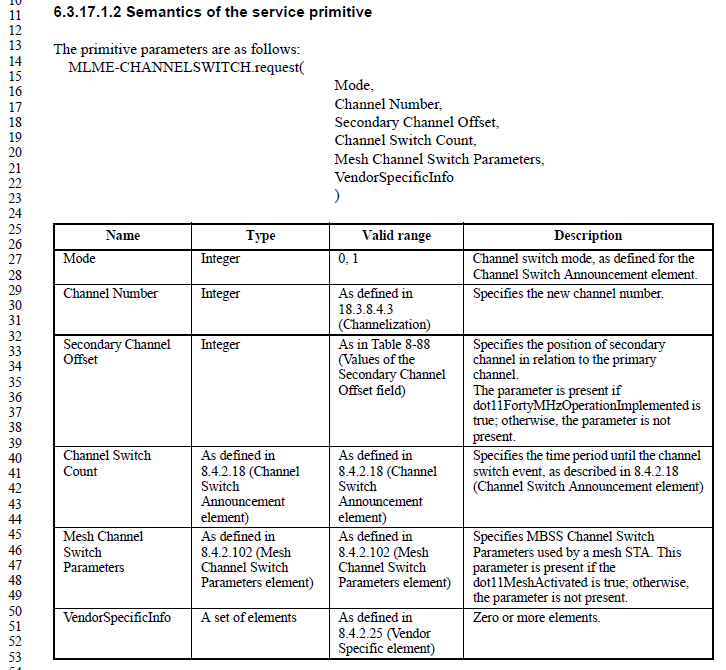
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3144 | 214.01 | 6.3.17 | A Channel Switch Announcement frame (8.6.2.6) had been extended to include optional Wide Bandwidth ChannelSwitch element and New VHT Transmit Power Envelope element. So, MLME-CHANNELSWITCH SAP interfaces need to be modified to include these primitive parameters. | Modify the primitive parameters of MLME-CHANNELSWITCH.request, MLME-CHANNELSWITCH.indication, and MLME-CHANNELSWITCH.response by adding Wide Bandwidth ChannelSwitch element and New VHT Transmit Power Envelope element. |

**Discussion:**

The context is:



With the .request primitive, for example:



Agreed, the parameters are missing.

**Proposed resolution: Revised**

In MLME-CHANNELSWITCH.request, MLME-CHANNELSWITCH.indication, and MLME-CHANNELSWITCH.response, add the following parameters just before VendorSpecificInfo:

Wide Bandwidth Channel Switch,

New VHT Transmit Power Envelope

And add the following rows to the parameter explanation tables:

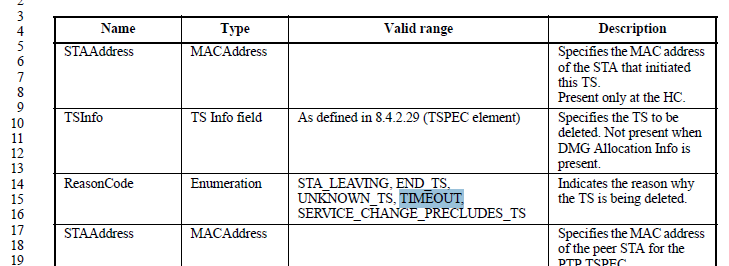
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| Wide Bandwidth Channel Switch | As defined in 8.4.2.160 (Wide Bandwidth Channel Switch element) | As defined in 8.4.2.160 (Wide Bandwidth Channel Switch element) | Specifies channel parameters used when switching to a channel width wider than 40 MHz. The parameter is present if dot11VHTOptionImplemented is true and switching to a channel width wider than 40 MHz; otherwise, the parameter is not present |
| New VHT Transmit Power Envelope | Set of New VHT Transmit Power Envelope elements | N/A | Specifies power parameters used for the BSS after the channel switch. Optionally present if dot11VHTOptionImplemented is true |

# CID 3521 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3521 | 237.15 | 6.3.26.6.2 | Get rid of all TIMEOUT ReasonCode values, unless there is explicit discussion of it in the protocol/behavior (and then give it a more descriptive name) | Get rid of all TIMEOUT ReasonCode values, unless there is explicit discussion of it in the protocol/behavior (and then give it a more descriptive name). I.e., change TIMEOUT in 6.3.26.6.2 (and 6.3.26.7.2, 6.3.27.6.2, 6.3.27.7.2, 6.3.29.6.2 and 6.3.29.7.2) ReasonCode to INACTIVITY\_TIMEOUT; delete TIMEOUT from 6.3.93.10.2 and 6.3.93.11.2 (unless Relay links have a timeout semantics specified somewhere else); etc. |

**Discussion:**

The context (in 6.3.26.6.2) is:

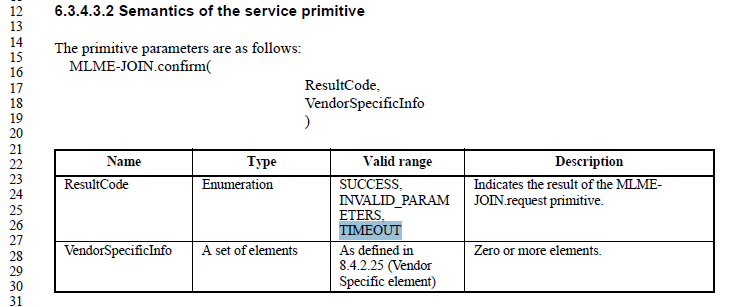


In this case, the TIMEOUT ReasonCode is used to indicate that the TS has been inactive too long. So, change the name to INACTIVITY\_TIMEOUT to make the meaning clear.

To the commeter’s point, do the same in 6.3.26.7.2, 6.3.27.6.2, 6.3.27.7.2, 6.3.29.6.2 and 6.3.29.7.2. This implies parallel changes in 10.4.10, 10.5.4, 10.7.2.2, 10.7.4.2

Note that there are two occurrences of Reason Code (or ReasonCode) in 10.5.4. One describes a reason code of “provide” which appears to be a nonsense typo. Change “provide” to INACTIVITY\_TIMEOUT, also. Same thing in 10.7.5.

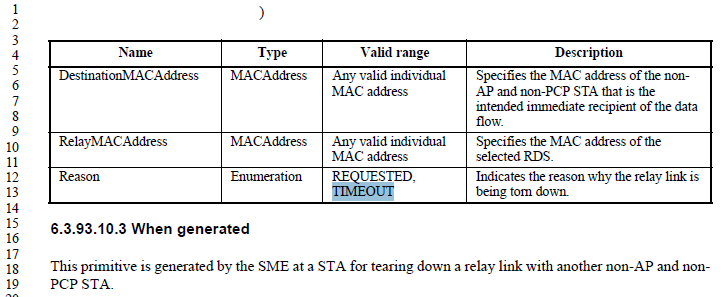
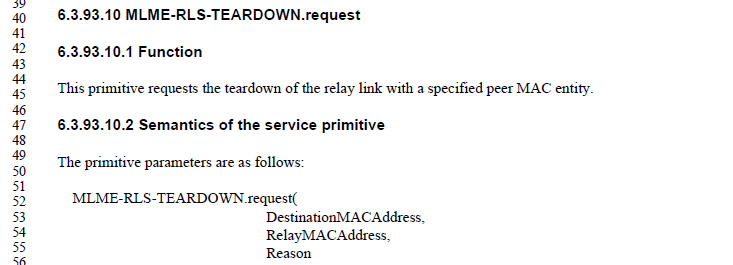
An example of another oneto rename:



In this case, the timeout is appropriate (there is a specific timeout parameter and semantics described in the text). However, the name could be JOIN\_FAILURE\_TIMEOUT (which matches the name of the controlling parameter), to be clear about the meaning.

And yet another is the TIMEOUT in the DMG BF-TRAINING service. In this case, there are explicit timeouts that can happen per the behavior described in the text. (These appear to be only for SLS, is that correct?) So, keeping the ResultCode is appropriate, but a more specific name, such as BF-TIMEOUT (or SLS-TIMEOUT?) would be preferred.

What about this one? In 6.3.93.10.2:



What does the Reason “TIMEOUT” mean in a request to teardown a relay link? (It is communicated to the peer in the .indication.)

Two other things to correct:

* DLS Setup does not have any support for a TIMEOUT at the service level (nor should it, it is just like any other request/response service, if the response doesn’t come back and the .confirm isn’t issued, it is up to the service user to decide what (and when) to take as appropriate action). But the text in 10.7.2.2 says the .confirm will be issued with ResultCode TIMEOUT. Recommend just deleting that sentence in 10.7.2.2.
* 10.37.2 has discussion of a TIMEOUT result code from a MLME-QAB.confirm. Again, this appears to be the same sort of situation as above for DLS Setup, and the text in 10.37.2 should be deleted.

**Proposed resolution: Revised**

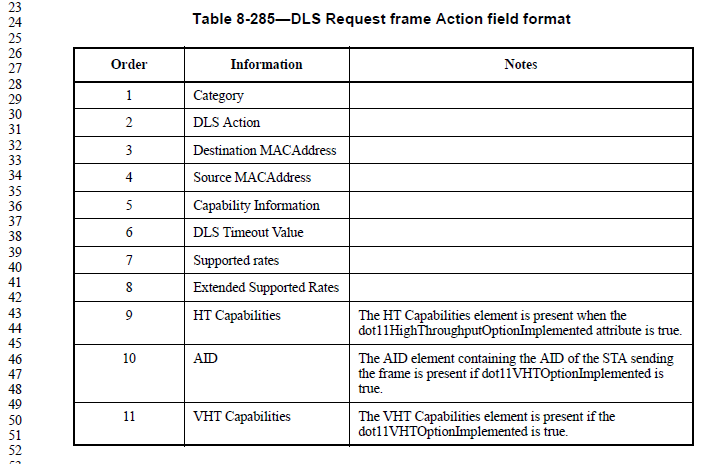
<TBD, once above red items are agreed>

# CID 3145 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3145 | 242.35 | 6.3.27 | A DLS Request frame format (8.5.4.2) and a DLS Response frame format (8.5.4.3) had been extended to include AID element and VHT Capabilities element. So, MLME-DLS SAP interfaces need to be modified to include these primitive parameters. | Modify the primitive parameters of MLME-DLS.confirm, MLME-DLS.indication, and MLME-DLS.response by adding AID element and VHT Capabilities element. |

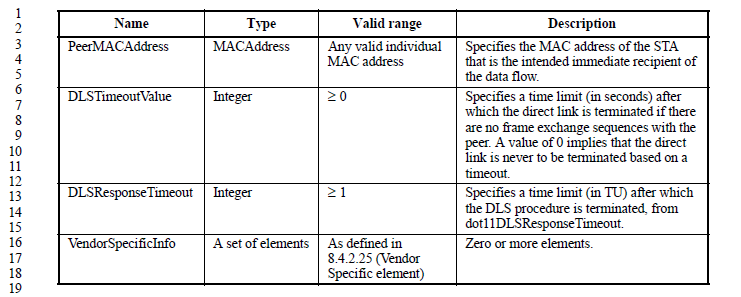
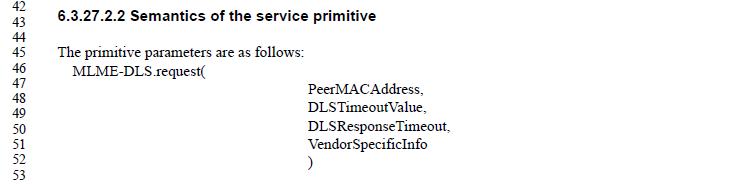
**Discussion:**

The context is:



(DLS Response frame is similar.)

With the .request primitive, for example:



We can see that many of the fields in the DLS Request frame are not provided in the service primitive, generally the values that the MAC would already know internally such as capabilities and supported rates.

The new fields seem to be in the same category; both AID and VHT Capabilities are already known by the MAC, and there is no reason for the service user to provide that information again across this primitive.

**Proposed resolution: Rejected**

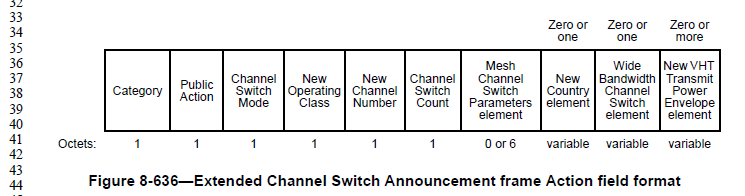
The MAC already has the information needed to fill in these new field values, with local information. The values do not need to be provided by the service user.

# CID 3146 (MAC)

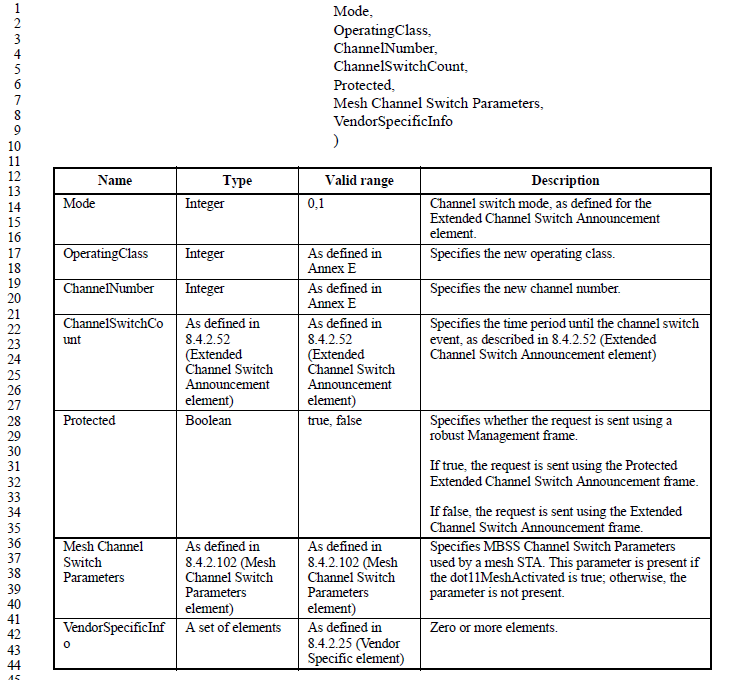
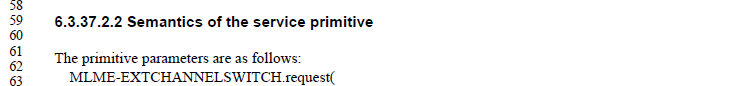
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3146 | 272.43 | 6.3.37 | A Extended Channel Switch Announcement frame format (8.6.8.7) had been extended to include New Country element, Wide Bandwidth ChannelSwitch element, and New VHT Transmit Power Envelope element. So, MLME-EXTCHANNELSWITCH SAP interfaces need to be modified to include these primitive parameters. | Modify the primitive parameters ofMLME-EXTCHANNELSWITCH.request, MLME-EXTCHANNELSWITCH.indication, and MLME-EXTCHANNELSWITCH.response by adding New Country element, Wide Bandwidth ChannelSwitch element and New VHT Transmit Power Envelope element. |

**Discussion:**

The context is:



With the .request primitive, for example:



Agreed, the parameters are missing.

**Proposed resolution: Revised**

In MLME-EXTCHANNELSWITCH.request, MLME-EXTCHANNELSWITCH.indication, and MLME-EXTCHANNELSWITCH.response, add the following parameters just before VendorSpecificInfo:

New Country,

Wide Bandwidth Channel Switch,

New VHT Transmit Power Envelope

And add the following rows to the parameter explanation tables:

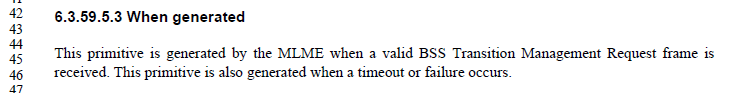
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid Range | Description |
| New Country | As defined in 8.4.2.9 (Country element) | As defined in 8.4.2.9 (Country element) | Specifies the country, operating class table and operating classes used for the BSS after the channel switch. Optionally present if dot11VHTOptionImplemented is true |
| Wide Bandwidth Channel Switch | As defined in 8.4.2.160 (Wide Bandwidth Channel Switch element) | As defined in 8.4.2.160 (Wide Bandwidth Channel Switch element) | Specifies channel parameters used when switching to a channel width wider than 40 MHz. The parameter is present if dot11VHTOptionImplemented is true and switching to a channel width wider than 40 MHz; otherwise, the parameter is not present |
| New VHT Transmit Power Envelope | Set of New VHT Transmit Power Envelope elements | N/A | Specifies power parameters used for the BSS after the channel switch. Optionally present if dot11VHTOptionImplemented is true |

# CID 3520 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3520 | 346.46 | 6.3.59.5.3 | In BSS Transition Management Request subclause of 6, "This primitive is also generated when a timeout or failure occurs" which isn't true (TIMEOUT has been removed) | Delete last sentence of 6.3.59.5.3. |

**Discussion:**

The context is:



Note that this subclause is the MLME-BTM.indication primitive. So, talking about timeout or failure here really makes no sense.

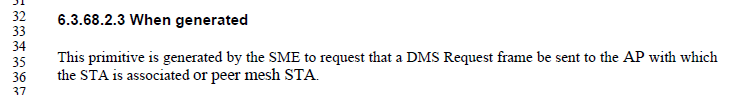
**Proposed resolution: Accepted.**

# CID 3061 (MAC)

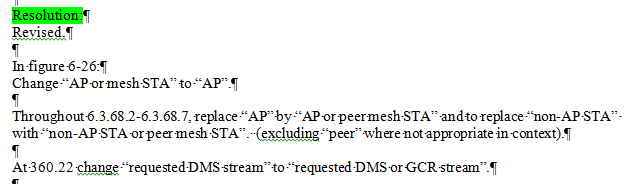
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3061 | 379.49 | 6.3.68.2 | The change from CID 2003 (see 11-14/207r1) is not sufficient. The GATS.request only uses DMS frames for a DMS stream. For GCR, it uses other frames.  The following locations need to be updated to be non-specific to DMS:  380.35, 380.41, 381.31, 381.36, 381.61, 382.03, 387.30, 382.49, 383.05, 383.10, 383.37, 383.41, 383.61, 384.22, 384.26, 384.45 | Either:  1. Make cited locations generic to both DMS and GCR  2. Make cited location specific to DMS and add parallel statement for GCR |

**Discussion:**

The context is:



For reference, the resolution of CID 2003 is:



**Proposed resolution: Rejected.**

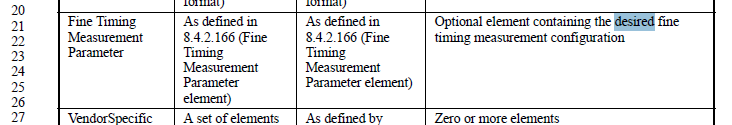
GCR streams are managed with DMS Request, DMS Response and some other frame types (ADDBA, Association, etc.) The DMS Request and DMS Response frames are requested with the cited primitives, including when they are used for GCR. The other frame types similarly have their own primitives No change is needed.

# CID 3062 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3062 | 387.21 | 6.3.70.3.2 | "desired" - desired by whom?  Ditto at 388.21. | Reword to avoid anthropomophism |

**Discussion:**

The context is:



**Proposed resolution: Revised.**

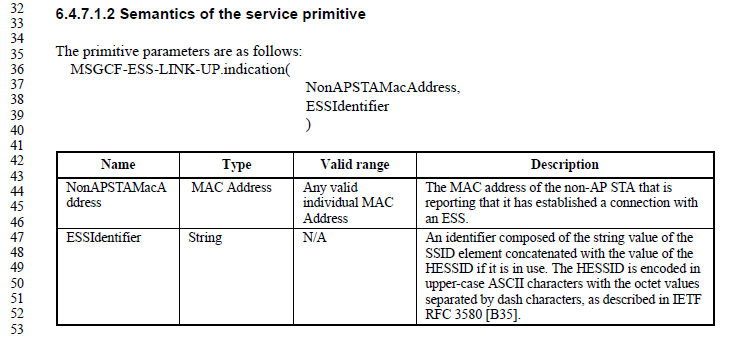
Replace “desired” with “requested” in both cited locations.

# CID 3500 (MAC)

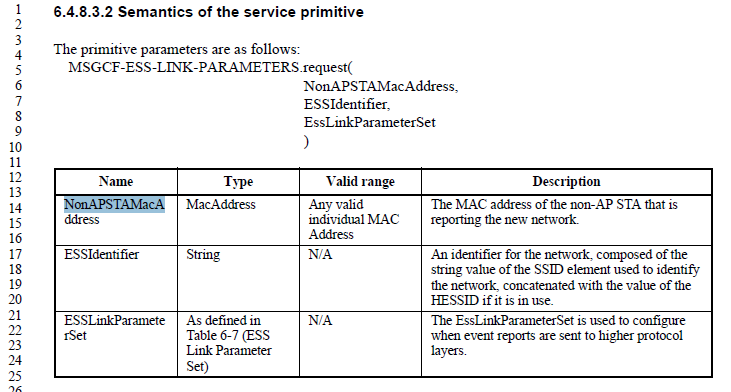
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3500 | 503.37 | 6.4.7.1.2 | Why do MSGCF-ESS-(Link)\* primitives need a NonAPMACAddress parameter? The higher layers knowing or determining the MAC Address of the STA that generates primitives is a local implementation function, not part of the MLME definition. | Remove the NonAPSTAMACAddress parameter from these primitives. |

**Discussion:**

The context is:



This parameter does appear to be self-referential to the STA issuing the primitive, in all the .indication and .confirm cases. In the .request cases, this parameter makes no sense at all, given the description, for example:



**Proposed resolution: Revised.**

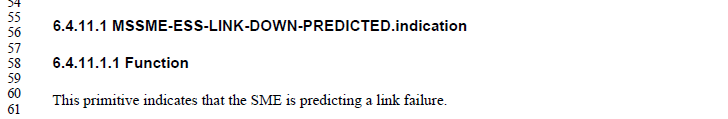
Remove the NonAPSTAMacAddress parameter from the parameter list and the parameter description table in the following primitives: MSGCF-ESS-LINK-UP.indication, MSGCF-ESS-LINK-DOWN.indication, MSGCF-ESS-LINK-GOING-DOWN.indication, MSGCF-ESS-LINK-EVENT-ROLLBACK.indication, MSGCF-ESS-LINK-DETECTED.indication, MSGCF-ESS-LINK-SCAN.confirm, MSGCF-ESS-LINK-CAPABILITY.request, MSGCF-ESS-LINK-CAPABILITY.confirm, MSGCF-ESS-LINK-PARAMETERS.request, MSGCF-ESS-LINK-PARAMETERS.confirm, MSGCF-ESS-LINK-THRESHOLD-REPORT.indication, MSGCF-ESS-LINK-COMMAND.request, and MSSME-ESS-LINK-GOING-DOWN.indication.

# CID 3063 (MAC)

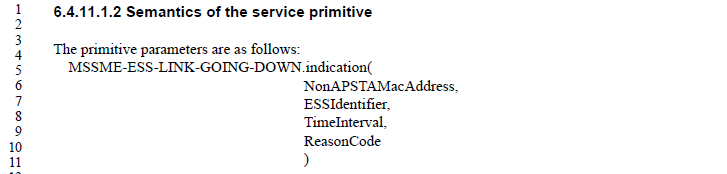
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3063 | 518.56 | 6.4.11.1 | "MSSME-ESS-LINK-DOWN-PREDICTED.indication" does not match "MSSME-ESS-LINK-GOING-DOWN.indication" at 519.05 | Make them the same |

**Discussion:**

The context is:



and:



Either name makes sense, given the description of the primitive. Neither name is used/referenced in any other location in the Standard. Thus, this is a coin toss. The LINK-GOING-DOWN would be more consistent with the very similar MSGCF-ESS-LINK-GOING-DOWN.indication primitive.

**Proposed resolution: Revised.**

Replace “MSSME-ESS-LINK-DOWN-PREDICTED” with “MSSME-ESS-LINK-GOING-DOWN” at 518.56

# CID 3524 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3524 | 647.01 | 8.4.1.9 | All StatusCodes and ResultCodes should have a name | It's just easier to talk about these, if they have a name. And, fix embedded magic numbers, such as in 10.33.2.2, use the name instead. |

**Discussion:**

It is assumed the commenter meant Reason codes (in Table 8-52), as that is the broadly used field and values. Result codes are actually generally constrained to an individual element/frame, and there are several, small tables of these, which could also be better named, but seems like less of a problem.

The context for Reason codes is shown below, with proposed names added like this:

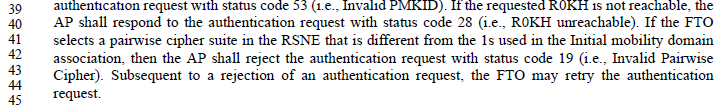
|  |  |  |
| --- | --- | --- |
| * Reason codes | | |
| Reason code | Name | Meaning |
| 0 |  | Reserved |
| 1 | UNSPECIFIED\_REASON | Unspecified reason |
| 2 | INVALID\_AUTHENTICATION | Previous authentication no longer valid |
| 3 | LEAVING\_NETWORK\_DEAUTH | Deauthenticated because sending STA is leaving (or has left) IBSS or ESS |
| 4 | REASON\_INACTIVITY | Disassociated due to inactivity |
| 5 | NO\_MORE\_STAS | Disassociated because AP is unable to handle all currently associated STAs |
| 6 | INVALID\_CLASS2\_FRAME | Class 2 frame received from nonauthenticated STA |
| 7 | INVALID\_CLASS3\_FRAME | Class 3 frame received from nonassociated STA |
| 8 | LEAVING\_NETWORK\_DISASSOC | Disassociated because sending STA is leaving (or has left) BSS |
| 9 | NOT\_AUTHENTICATED | STA requesting (re)association is not authenticated with responding STA |
| 10 | UNACCEPTABLE\_POWER\_CAPABILITY | Disassociated because the information in the Power Capability element is -unacceptable |
| 11 | UNACCEPTABLE\_SUPPORTED\_CHANNELS | Disassociated because the information in the Supported Channels element is -unacceptable |
| 12 | BSS\_TRANSITION\_DISASSOC | Disassociated due to BSS (#1202)transition (Ed)management |
| 13 | REASON\_INVALID\_ELEMENT | Invalid element, i.e., an element defined in this standard for which the content does not meet the specifications in Clause 8 (Frame formats) |
| 14 | MIC\_FAILURE | Message integrity code (MIC) failure |
| 15 | 4WAY\_HANDSHAKE\_TIMEOUT | 4-Way Handshake timeout |
| 16 | GK\_HANDSHAKE\_TIMEOUT | Group Key Handshake timeout |
| 17 | HANDSHAKE\_ELEMENT\_MISMATCH | (#1509)Element in 4-Way Handshake different from (Re)Association Request/Probe Response/Beacon frame |
| 18 | REASON\_INVALID\_GROUP\_CIPHER | Invalid group cipher |
| 19 | REASON\_INVALID\_PAIRWISE\_CIPHER | Invalid pairwise cipher |
| 20 | REASON\_INVALID\_AKMP | Invalid AKMP |
| 21 | UNSUPPORTED\_RSNE\_VERSION | Unsupported RSNE version |
| 22 | INVALID\_RSNE\_CAPABILITIES | Invalid RSNE capabilities |
| 23 | 802\_1\_X\_AUTH\_FAILED | IEEE Std(#130) 802.1X authentication failed |
| 24 | REASON\_CIPHER\_OUT\_OF\_POLICY | Cipher suite rejected because of the security policy |
| 25 | TDLS\_PEER\_UNREACHABLE | TDLS direct-link teardown due to TDLS peer STA unreachable via the TDLS direct link |
| 26 | TDLS\_UNSPECIFIED\_REASON | TDLS direct-link teardown for unspecified reason |
| 27 | SSP\_REQUESTED\_DISASSOC | Disassociated because session terminated by SSP request |
| 28 | NO\_SSP\_ROAMING\_AGREEMENT | Disassociated because of lack of SSP roaming agreement |
| 29 | BAD\_CIPHER\_OR\_AKM | Requested service rejected because of SSP cipher suite or AKM requirement |
| 30 | NOT\_AUTHORIZED\_THIS\_LOCATION | Requested service not authorized in this location |
| 31 | SERVICE\_CHANGE\_ PRECLUDES\_TS | TS deleted because QoS AP lacks sufficient bandwidth for this QoS STA due to a change in BSS service characteristics or operational mode (e.g., an HT BSS change from 40 MHz channel to 20 MHz channel) |
| 32 | UNSPECIFIED\_QOS\_REASON | Disassociated for unspecified, QoS-related reason |
| 33 | NOT\_ENOUGH\_BANDWIDTH | Disassociated because QoS AP lacks sufficient bandwidth for this QoS STA |
| 34 | MISSING\_ACKS | Disassociated because excessive number of frames need to be acknowledged, but are not acknowledged due to AP transmissions and/or poor channel conditions |
| 35 | EXCEEDED\_TXOP | Disassociated because STA is transmitting outside the limits of its TXOPs |
| 36 | STA\_LEAVING | Requesting STA(#1258) is leaving the BSS (or resetting) |
| 37 | END\_TS  END\_BA  END\_DLS | Requesting STA is no longer using the stream or session(#1259) |
| 38 | UNKNOWN\_TS  UNKNOWN\_BA | Requesting STA received frames using a mechanism for which a setup has not been completed(#1260) |
| 39 | TIMEOUT | Requested from peer STA due to timeout |
| 45 | PEERKEY\_MISMATCH | Peer STA does not support the requested cipher suite |
| 46 | PEER\_INITIATED | In a DLS Teardown frame: The teardown was initiated by the DLS peer  In a Disassociation frame: Disassociated because authorized access limit reached |
| 47 | AP\_INITIATED | In a DLS Teardown frame: The teardown was initiated by the AP  In a Disassociation frame: Disassociated due to external service requirements |
| 48 | REASON\_INVALID\_FT\_ACTION\_FRAME\_COUNT | Invalid FT Action frame count |
| 49 | REASON\_INVALID\_PMKID | Invalid pairwise master key identifier (PMKID) |
| 50 | REASON\_INVALID\_MDE | Invalid MDE |
| 51 | REASON\_INVALID\_FTE | Invalid FTE |
| 52 | MESH-PEERING-CANCELLED | Mesh peering cancelled for unknown reasons(#1261) |
| 53 | MESH-MAX-PEERS | The mesh STA has reached the supported maximum number of peer mesh STAs |
| 54 | MESH-CONFIGURATION-POLICY-VIOLATION | The received information violates the Mesh Configuration policy configured in the mesh STA profile |
| 55 | MESH-CLOSE-RCVD | The mesh STA has received a Mesh Peering Close frame(#2321) requesting to close the mesh peering. |
| 56 | MESH-MAX-RETRIES | The mesh STA has resent dot11MeshMaxRetries Mesh Peering Open frames(#2321), without receiving a Mesh Peering Confirm frame(#2321). |
| 57 | MESH-CONFIRM-TIMEOUT | The confirmTimer for the mesh peering instance times out. |
| 58 | MESH-INVALID-GTK | The mesh STA fails to unwrap the GTK or the values in the wrapped contents do not match |
| 59 | MESH-INCONSISTENT-PARAMETERS | The mesh STA receives inconsistent information about the mesh parameters between (#2314)mesh peering Management frames |
| 60 | MESH-INVALID-SECURITY-CAPABILITY | The mesh STA fails the authenticated mesh peering exchange because due to failure in selecting either the pairwise ciphersuite or group ciphersuite |
| 61 | MESH-PATH-ERROR-NO-PROXY-INFORMATION | The mesh STA does not have proxy information for this external destination. |
| 62 | MESH-PATH-ERROR-NO-FORWARDING-INFORMATION | The mesh STA does not have forwarding information for this destination. |
| 63 | MESH-PATH-ERROR-DESTINATION-UNREACHABLE | The mesh STA determines that the link to the next hop of an active path in its forwarding information is no longer usable. |
| 64 | MAC-ADDRESS-ALREADY-EXISTS-IN-MBSS | The Deauthentication frame was sent because the MAC address of the STA already exists in the mesh BSS. See 10.3.6 (Additional mechanisms for an AP collocated with a mesh STA). |
| 65 | MESH-CHANNEL-SWITCH-REGULATORY-REQUIREMENTS | The mesh STA performs channel switch to meet regulatory requirements. |
| 66 | MESH-CHANNEL-SWITCH-UNSPECIFIED | The mesh STA performs channel switch with unspecified reason. |
| 67–65 535 |  | Reserved |

The context for Status codes is shown below, with proposed names added like this:

|  |  |  |
| --- | --- | --- |
| * Status codes | | |
| Status code | Name | Meaning |
| 0 | SUCCESS | Successful |
| 1 | REFUSED, REFUSED\_REASON\_UNSPECIFIED | Unspecified failure |
| 2 | TDLS\_REJECTED\_ALTERNATIVE\_PROVIDED | TDLS wakeup schedule rejected but alternative schedule provided |
| 3 | TDLS\_REJECTED | TDLS wakeup schedule rejected |
| 4 |  | Reserved |
| 5 | SECURITY\_DISABLED | Security disabled |
| 6 | UNACCEPTABLE\_LIFETIME | Unacceptable lifetime |
| 7 | NOT\_IN\_SAME\_BSS | Not in same BSS |
| 8–9 |  | Reserved |
| 10 | REFUSED\_CAPABILITIES\_ MISMATCH | Cannot support all requested capabilities in the Capability Information field |
| 11 | DENIED\_NO\_ASSOCIATION\_EXISTS | Reassociation denied due to inability to confirm that association exists |
| 12 | DENIED\_OTHER\_REASON | Association denied due to reason outside the scope of this standard |
| 13 | UNSUPPORTED\_AUTH\_ALGORITHM | Responding STA does not support the specified authentication algorithm |
| 14 | TRANSACTION\_SEQUENCE\_ERROR | Received an Authentication frame with authentication transaction sequence number out of expected sequence |
| 15 | CHALLENGE\_FAILURE | Authentication rejected because of challenge failure |
| 16 | REJECTED\_SEQUENCE\_TIMEOUT | Authentication rejected due to timeout waiting for next frame in sequence |
| 17 | DENIED\_NO\_MORE\_STAS | Association denied because AP is unable to handle additional associated STAs |
| 18 | REFUSED\_BASIC\_RATES\_ MISMATCH | Association denied due to requesting STA not supporting all of the data rates in the BSSBasicRateSet parameter |
| 19 | DENIED\_NO\_SHORT\_PREAMBLE\_SUPPORT | Association denied due to requesting STA not supporting the short preamble option |
| 20 |  | Reserved(#302) |
| 21 |  | Reserved(#63) |
| 22 | REJECTED\_SPECTRUM\_MANAGEMENT\_REQUIRED | Association request rejected because Spectrum Management capability is required |
| 23 | REJECTED\_BAD\_POWER\_CAPABILITY | Association request rejected because the information in the Power Capability -element is unacceptable |
| 24 | REJECTED\_BAD\_SUPPORTED\_CHANNELS | Association request rejected because the information in the Supported Channels -element is unacceptable |
| 25 | DENIED\_NO\_SHORT\_SLOT\_TIME\_SUPPORT | Association denied due to requesting STA not supporting the Short Slot Time option |
| 26 |  | Reserved |
| 27 | DENIED\_NO\_HT\_SUPPORT | Association denied because the requesting STA does not support HT features |
| 28 | R0KH\_UNREACHABLE | R0KH unreachable |
| 29 | DENIED\_PCO\_TIME\_NOT\_SUPPORTED | Association denied because the requesting STA does not support the phased coexistence operation (PCO) transition time required by the AP |
| 30 | REFUSED\_TEMPORARILY | Association request rejected temporarily; try again later |
| 31 | ROBUST\_MANAGEMENT\_POLICY\_VIOLATION | Robust management frame policy violation |
| 32 | UNSPECIFIED\_QOS\_FAILURE | Unspecified, QoS-related failure |
| 33 | DENIED\_INSUFFICIENT\_BANDWIDTH | Association denied because QoS AP or PCP(11ad) has insufficient bandwidth to handle another QoS STA |
| 34 | DENIED\_POOR\_CHANNEL\_CONDITIONS | Association denied due to excessive frame loss rates and/or poor conditions on current operating channel |
| 35 | DENIED\_QOS\_NOT\_SUPPORTED | Association (with QoS BSS) denied because the requesting STA does not support the QoS -facility |
| 36 |  | Reserved |
| 37 | REQUEST\_DECLINED (11aa) | The request has been declined |
| 38 | INVALID\_PARAMETERS | The request has not been successful as one or more parameters have invalid values |
| 39 | REJECTED\_WITH\_SUGGESTED\_ CHANGES | The allocation or(11ad) TS has not been created because the request cannot be honored; however, a suggested TSPEC/DMG TSPEC(11ad) is provided so that the initiating STA can(#2324) attempt to set another allocation or(11ad) TS with the suggested changes to the TSPEC/DMG TSPEC(11ad) |
| 40 | STATUS\_INVALID\_ELEMENT | Invalid element, i.e., an element defined in this standard for which the content does not meet the specifications in Clause 8 (Frame formats) |
| 41 | STATUS\_INVALID\_GROUP\_CIPHER | Invalid group cipher |
| 42 | STATUS\_INVALID\_PAIRWISE\_CIPHER | Invalid pairwise cipher |
| 43 | STATUS\_INVALID\_AKMP | Invalid AKMP |
| 44 | UNSUPPORTED\_RSNE\_VERSION | Unsupported RSNE version |
| 45 | INVALID\_RSNE\_CAPABILITIES | Invalid RSNE capabilities |
| 46 | STATUS\_CIPHER\_OUT\_OF\_POLICY | Cipher suite rejected because of security policy |
| 47 | REJECTED\_FOR\_DELAY\_PERIOD | The TS or allocation(11ad) has not been created; however, the HC or PCP(11ad) might(#1262) be capable of creating a TS or allocation(11ad), in response to a request, after the time indicated in the TS Delay element |
| 48 | DLS\_NOT\_ALLOWED | Direct link is not allowed in the BSS by policy |
| 49 | NOT\_PRESENT | The Destination STA is not present within this BSS |
| 50 | NOT\_QOS\_STA | The Destination STA is not a QoS STA |
| 51 | DENIED\_LISTEN\_INTERVAL\_TOO\_LARGE | Association denied because the Listen Interval is too large |
| 52 | STATUS\_INVALID\_FT\_ACTION\_FRAME\_COUNT | Invalid FT Action frame count |
| 53 | STATUS\_INVALID\_PMKID | Invalid pairwise master key identifier (PMKID) |
| 54 | STATUS\_INVALID\_MDE | Invalid MDE |
| 55 | STATUS\_INVALID\_FTE | Invalid FTE |
| 56 | REQUESTED\_TCLAS\_NOT\_ SUPPORTED(#2061)(11aa) | Requested TCLAS processing is not supported by the AP or PCP(11ad). |
| 57 | INSUFFICIENT\_TCLAS\_ PROCESSING\_RESOURCES(11aa) | The AP or PCP(11ad) has insufficient TCLAS processing resources to satisfy the request. |
| 58 | TRY\_ANOTHER\_BSS | The TS has not been created because the request cannot be honored; however, the HC or PCP(11ad) suggests that the STA transition to a different BSS(#1263) to set (#1264)up the TS. |
| 59 | GAS\_ADVERTISEMENT\_ PROTOCOL\_NOT\_SUPPORTED | GAS Advertisement Protocol not supported |
| 60 | NO\_OUTSTANDING\_GAS\_ REQUEST | No outstanding GAS request |
| 61 | GAS\_RESPONSE\_NOT\_ RECEIVED\_FROM \_SERVER | GAS Response not received from the Advertisement Server |
| 62 | GAS\_QUERY\_TIMEOUT | STA timed out waiting for GAS Query Response |
| 63 | GAS\_QUERY\_RESPONSE\_ TOO\_ LARGE | GAS Response is larger than query response length limit |
| 64 | REJECTED\_HOME\_WITH\_ SUGGESTED\_CHANGES | Request refused because home network does not support request |
| 65 | SERVER\_UNREACHABLE | Advertisement Server in the network is not currently reachable |
| 66 |  | Reserved |
| 67 | REJECTED\_FOR\_SSP\_ PERMISSIONS | Request refused due to permissions received via SSPN interface |
| 68 | REFUSED\_UNAUTHENTICATED\_ACCESS\_NOT\_SUPPORTED | Request refused because the(#2323) AP or PCP(11ad) does not support unauthenticated access |
| 69-71 |  | Reserved |
| 72 | INVALID\_RSNE | Invalid contents of RSNE |
| 73 | U\_APSD\_COEXISTANCE\_NOT\_SUPPORTED | U-APSD Coexistence is not supported. |
| 74 | U\_APSD\_COEX\_MODE\_NOT\_SUPPORTED | Requested U-APSD Coexistence mode is not supported. |
| 75 | BAD\_INTERVAL\_WITH\_U\_APSD\_COEX | Requested Interval/Duration value cannot be  supported with U-APSD Coexistence. |
| 76 | ANTI\_CLOGGING\_TOKEN\_REQUIRED | Authentication is rejected because an Anti-Clogging Token is required. |
| 77 | UNSUPPORTED\_FINITE\_CYCLIC\_GROUP | Authentication is rejected because the offered finite cyclic group is not supported. |
| 78 | CANNOT\_FIND\_ALTERNATIVE\_ TBTT | The TBTT adjustment request has not been successful because the STA could not find an alternative TBTT. |
| 79 | TRANSMISSION\_FAILURE | Transmission failure |
| 80 | REQUESTED\_TCLAS\_NOT\_ SUPPORTED | Requested TCLAS Not Supported. |
| 81 | TCLAS\_RESOURCES\_EXHAUSTED | TCLAS Resources Exhausted. |
| 82 | REJECTED\_WITH\_SUGGESTED\_ BSS\_TRANSITION | Rejected with Suggested BSS (#1202)transition. |
| 83(11ad) | REJECT\_WITH\_SCHEDULE | Reject with recommended schedule |
| 84(11ad) | REJECT\_NO\_WAKEUP\_SPECIFIED | Reject because no wakeup schedule specified |
| 85(11ad) | SUCCESS\_POWER\_SAVE\_MODE | Success, the destination STA is in power save mode |
| 86(11ad) | PENDING\_ADMITTING\_FST\_SESSION | FST pending, in process of admitting FST session |
| 87(11ad) | PERFORMING\_FST\_NOW | Performing FST now |
| 88(11ad) | PENDING\_GAP\_IN\_BA\_WINDOW | FST pending, gap(s) in block ack(#2069) window |
| 89(11ad) | REJECT\_U-PID\_SETTING | Reject because of U-PID setting |
| 90–91(11ad) |  | Reserved |
| 92 | REFUSED\_EXTERNAL\_REASON | (Re)association refused for some external reason |
| 93 | REFUSED\_AP\_OUT\_OF\_MEMORY | (Re)association refused because of memory limits at the AP |
| 94 | REJECTED\_EMERGENCY\_ SERVICES\_NOT\_SUPPORTED | (Re)association refused because emergency services are not supported at the AP. |
| 95 | QUERY\_RESPONSE\_ OUTSTANDING | GAS query response not yet received. |
| 96(11ad) | REJECT\_DSE\_BAND | Reject since the request is for transition to a frequency band subject to DSE procedures and FST Initiator is a dependent STA |
| 97(11aa) | TCLAS\_PROCESSING\_ TERMINATED | Requested TCLAS processing has been terminated by the AP. |
| 98(11aa) | TS\_SCHEDULE\_CONFLICT | The TS schedule conflicts with an existing schedule; an alternative schedule is provided. |
| 99(11ad) | DENIED\_WITH\_SUGGESTED\_BAND\_AND\_CHANNEL | The association has been denied; however, one or more Multi-band elements are included that can be used by the receiving STA to join the BSS |
| 100 | MCCAOP\_RESERVATION\_ CONFLICT | The request failed due to a reservation conflict |
| 101 | MAF\_LIMIT\_EXCEEDED | The request failed due to exceeded MAF limit |
| 102 | MCCA\_TRACK\_LIMIT\_EXCEEDED | The request failed due to exceeded MCCA track limit |
| 103(11ad) | DENIED\_DUE\_TO\_SPECTRUM\_MANAGEMENT | Association denied because the information in the Spectrum Management field is unacceptable |
| 104(11ac) | DENIED\_VHT\_NOT\_SUPPORTED | Association denied because the requesting STA does not support VHT features. |
| 105(11af) | ENABLEMENT DENIED | Enablement denied |
| 106(11af) | RESTRICTION FROM AUTHORIZED GDB | Enablement denied due to restriction from an authorized GDB |
| 107(11af) | AUTHORIZATION DEENABLED | Authorization deenabled |
| 108–65 535 |  | Reserved |

These new names should be used in the text, where “magic numbers” or long phrase quotes are being used currently.

There are two problems in the sentence containing “19” in this excerpt from 12.5.2 (2019.42):

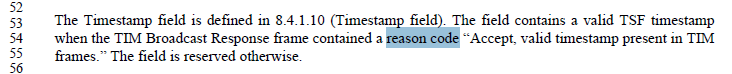


The first problem is the word “1s”. It appears this is supposed to be the English word “ones”. **Recommendation** is to replace “1s” with “ones”

The second problem is that status code 19 is listed in Table 8-53 as “Association denied due to requesting STA not supporting the short preamble option.” So, it doesn’t match the text above, which is discussing an invalid pairwise cipher suite selection. 19 does indicate an invalid pairwise cipher, when it is a Reason Code, the equivalent Status Code value is 42. It seems the two got mixed up in this text.

The challenge is that if this is fixed by changing the value to the correct one for a Status code (42), it changes over the air signalling behaviour. Of course, the current 19 can’t work properly, anyway, since that has a (different) existing meaning. **Recommendation** is to change the value to 42, while changing the text to reference the name of the Status code instead of the “magic number”. Same thing at 2021.52.

An aside, and actually not related directly to this comment (because the “reason code” mentioned here is from a different table (8-210), specific to TIM Broadcast Response element in 8.4.2.83), but this usage of “reason code” at 1171.54 is incorrect – the field is called Status field:



**Proposed resolution: Revised.**

Add names for unnamed Reason codes and Status codes, in their respective tables, as shown above.

Also, change:

* “reason code” to “Status field set to” at 1171 L54.
* ‘ “Previous Authentication no longer valid” ‘ to “INVALID\_AUTHENTICATION” at 1580.1 and 1583.46.
* ‘ “unspecified reason” ‘ to “UNSPECIFIED\_REASON” at 1586.56.
* “6 or 7” to “INVALID\_CLASS2\_FRAME or “INVALID\_CLASS3\_FRAME” at 1671.2
* ‘ “TDLS direct-link teardown for unspecified reason,” ‘ to “TDLS\_UNSPECIFIED\_REASON” at 1695.10
* ‘“TDLS direct-link teardown due to TDLS peer STA unreachable via the TDLS direct link.” ‘ to “TDLS\_PEER\_UNREACHABLE” at 1695.13.
* ‘3 (“Deauthenticated because sending STA is leaving (or has left) IBSS or ESS”)’ to “LEAVING\_NETWORK\_DEAUTH” at 1695.45 and 1695.49
* ‘“Disassociated because lack of SSP roaming agreement to SSPN.” ‘ to “NO\_SSP\_ROAMING\_AGREEMENT” at 1764.17
* ‘“Requested service rejected because of SSPN cipher suite requirement.” ‘ to “BAD\_CIPHER\_OR\_AKM” at 1764.32
* ‘ “Requested service not authorized in this location.” ‘ to “NOT\_AUTHORIZED\_THIS\_LOCATION” at 1764.41.
* ‘ “Disassociated because authorized access limit reached” ‘ to “PEER\_INITIATED” at 1765.13, 1765.21, 1765.28, 1765.36, and 1765.48
* “MIC failure” to “MIC\_FAILURE” at 1882.26
* ‘ “MIC failure” ‘ to “MIC\_FAILURE” at 1883.60.
* “2 (i.e., Previous authentication no longer valid)” to “INVALID\_AUTHENTICATION” at 2017.8

and

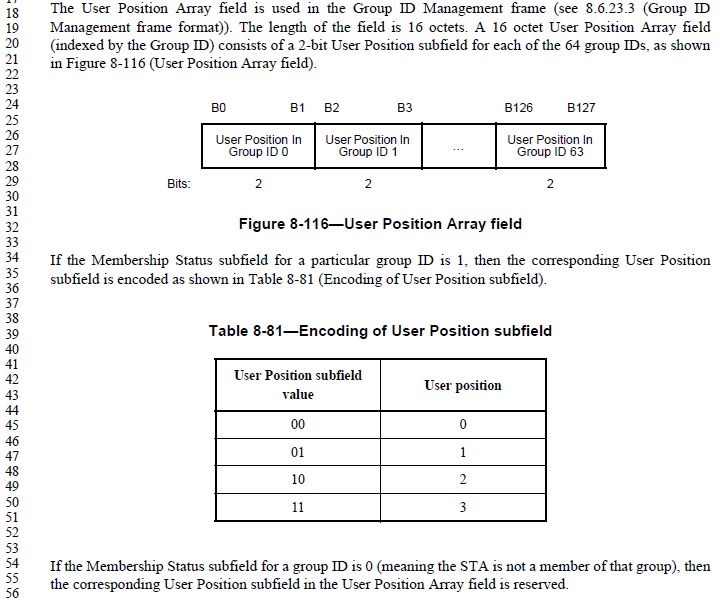
* “nonzero” to “not SUCCESS” at 448.31, 448.33, 450.35 and 450.39.
* “30” to “REFUSED\_TEMPORARILY” at 620.9 and 623.53
* “82” to “REJECTED\_WITH\_SUGGESTED\_BSS\_TRANSITION” at 632.22, 632.41 and 633.16
* “0” to “SUCCESS (0)” at 646.61.
* “85” to “SUCCESS\_POWER\_SAVE\_MODE” at 646.61
* “0” to “SUCCESS” at 654.27, 1080.14 and 1080.40
* “47” to “REJECTED\_FOR\_DELAY\_PERIOD” at 1080.36
* ‘0 (“Successful”)’ to “SUCCESS” at 1113.22
* “37 (The request has been declined)” to “REQUEST\_DECLINED” at 1113.23
* “nonzero” to “not SUCCESS” at 1116.17, 1116.27 and 1116.34
* “0” to “SUCCESS” at 1127.23 and 1128.48
* “0 (successful)” to “SUCCESS” at 1139.62, 1140.7, 1140.25, 1140.28, 1140.32, 1140.37, 1140.41, 1140.46, 1140.52, 1140.56, 1141.56, 1141.61, 1142.7, 1142.12, 1142.17 and 1142.27
* “0” to “SUCCESS” at 1140.10, 1140.17, 1140.19, 1141.7, 1141.10, 1142.23, 1145.14 and 1146.17
* ‘2 (“TDLS Wakeup Schedule rejected but alternative schedule provided”)’ to “TDLS\_REJECTED\_ALTERNATIVE\_PROVIDED” at 1146.20 and 1545.37
* “78” to “CANNOT\_FIND\_ALTERNATIVE\_TBTT” at 1185.46
* ‘0 (“Successful”)’ to “SUCCESS” at 1545.33, 1545.41 and 1546.50.
* Delete “nonzero” at 1565.53.
* “nonzero status code” to “status code not equal to SUCCESS” at 1592.52 and 1694.10
* “Successful” to “SUCCESS” at 1575.38, 1578.14, 1578.20, 1578.25, 1580.29, 1581.21, 1582.3, 1582.8
* “success” to “SUCCESS” at 1595.65.
* “0” to “SUCCESS” at 1607.28 and 1607.33.
* “SUCCESSFUL” to “SUCCESS” at 1613.5.
* “0 (“Successful”)” to “SUCCESS” at 1693.48, 1694.42, 1694.52, 1694.56, 1694.60, 1694.65, 1696.37, 1698.12,
* ‘37 (“The request has been declined”)”’to “REQUEST\_DECLINED” at 1693.31 and 1696.36
* ‘ “GAS Advertisement Protocol not supported” ‘ to “GAS\_ADVERTISEMENT\_PROTOCOL\_NOT\_SUPPORTED” at 1753.36
* ‘ “Advertisement Server in the network is not currently reachable,” ‘ to “SERVER\_UNREACHABLE” at 1753.45
* “the Advertisement Server is unreachable” to “SERVER\_UNREACHABLE” at 1753.50
* “success” to “SUCCESS” at 1753.62, 1755.10, 1755.18 and 1755.40
* ‘ “GAS Query Response larger than query response length limit” ‘ to “GAS\_QUERY\_RESPONSE\_TOO\_ LARGE” at 1754.17
* ‘ “The request has been declined.” ‘ to “REQUEST\_DECLINED” at 1754.33
* ‘ “Query Response too large” ‘ to “GAS\_QUERY\_RESPONSE\_TOO\_ LARGE” at 1754.1 and 1755.35.
* ‘status equal to “Response not received from server”’ to ‘status code equal to “GAS\_RESPONSE\_NOT\_RECEIVED\_FROM \_SERVER” at 1755.59
* ‘ “No request outstanding” ‘ to “NO\_OUTSTANDING\_GAS\_REQUEST” at 1756.2
* “37 (The request has been declined)” to “REQUEST\_DECLINED” at 1772.31, 1773.7, 1774.48, 1799.51 and 1802.25
* “0 (Successful)” to “SUCCESS” at 1772.36, 1772.37, 1773.20 and 1774.46
* “= 86” to “PENDING\_ADMITTING\_FST\_SESSION” at 1802.40
* “=88” to “PENDING\_GAP\_IN\_BA\_WINDOW” at 1802.43
* “= 39” to “REJECTED\_WITH\_SUGGESTED\_CHANGES” at 1802.45
* “= 37” to “REQUEST\_DECLINED” at 1802.49
* “= 96” to “REJECT\_DSE\_BAND” at 1802.58
* “= 0” to “SUCCESS” at 1803.16
* ‘37 (“The request has been declined”)’ to “REQUEST\_DECLINED” at 1806.10, 1815.37, 1815.42, 1816.10, 1816.14
* ‘0 (“Successful”)’ to “SUCCESS” at 1831.51, 1832.34 and 1833.4
* ‘105 (“Enablement denied”)’ to “ENABLEMENT DENIED” at 1831.53
* ‘38 (“The request has not been successful as one or more parameters have invalid values”)’ to “INVALID\_PARAMETERS” at 1831.53
* ‘106 (“Enablement denied due to restriction from the authorized GDB”)’ to “RESTRICTION FROM AUTHORIZED GDB” at 1831.54
* ‘107 (“Authorization Deenabled”)’ to “AUTHORIZATION DEENABLED” at 1831.58, 1831.62, 1832.38 and 1833.32
* “105, 106, 38, or Failed enablement attempt within dot11GDDEnablementTimeLimit” to “ENABLEMENT DENIED, RESTRICTION FROM AUTHORIZED GDB, INVALID\_PARAMETERS, or otherwise failed enablement attempt within dot11GDDEnablementTimeLimit” at 1832.28
* ‘ “successful” ‘ to “SUCCESS” at 1852.56 and 1852.60
* “13” to “UNSUPPORTED\_AUTH\_ALGORITHM” at 1853.30
* “0” to “SUCCESS” at 1853.37, 1853.63, 1854.13,
* “15” to “CHALLENGE\_FAILURE” at 1853.48, 1853.57, 1854.7, 1854.22, 1854.31, 1854.39, 1854.43, 1854.50, 1854.55 and 1854.61
* “zero (0). Nonzero status codes” to “SUCCESS. Status codes not equal to SUCCESS” at 1865.35, 1865.61
* " nonzero status code” to “status code not equal to SUCCESS” at 1866.14 and 1866.21
* “77” to “UNSUPPORTED\_FINITE\_CYCLIC\_GROUP” at 1866.15
* “76” to “ANTI\_CLOGGING\_TOKEN\_REQUIRED” at 1866.18
* ‘fifteen (15), “Authentication rejected; the response to the challenge failed.”’ To “CHALLENGE\_FAILURE” at 1866.24
* “76” to “ANTI\_CLOGGING\_TOKEN\_REQUIRED” at 1870.39, 1871.45
* “nonzero” to “not SUCCESS” at 1871.18
* “77” to “UNSUPPORTED\_FINITE\_CYCLIC\_GROUP” at 1870.21, 1871.51, 1872.5
* ‘ “Robust management frame policy violation” ‘ to “ROBUST\_MANAGEMENT\_POLICY\_VIOLATION” at 1913.55, 1914.10, 1915.51, 1915.57
* ‘Reason Code “Robust management frame policy violation.” ‘ to “Status Code ROBUST\_MANAGEMENT\_POLICY\_VIOLATION” at 1915.43 and 1915.46.
* ‘5 (“Security disabled”).’ To “SECURITY\_DISABLED” at 1977.22
* ‘38 (“The request has not been successful as one or more parameters have invalid values”)’ to “INVALID\_PARAMETERS” at 1977.27
* ‘44 (“Unsupported RSNE version”)’ to “UNSUPPORTED\_RSNE\_VERSION” at 1977.30
* ‘43 (“Invalid AKMP”)’ to “STATUS\_INVALID\_AKMP” at 1977.34
* ‘42 (“Invalid pairwise cipher”)’ to “STATUS\_INVALID\_PAIRWISE\_CIPHER” at 1977.39
* ‘45 (“Invalid RSNE capabilities”)’ to “INVALID\_RSNE\_CAPABILITIES” at 1977.41
* ‘6 (“Unacceptable lifetime”)’ to “UNACCEPTABLE\_LIFETIME” at 1977.45
* ‘55 (“Invalid FTE”)’ to “STATUS\_INVALID\_FTE” at 1977.48
* ‘44 (“Unsupported RSNE version”)’ to “UNSUPPORTED\_RSNE\_VERSION” at 1978.50
* ‘72 (“Invalid contents of RSNE”)’ to “INVALID\_RSNE” at 1978.55
* ‘42 (“Invalid pairwise cipher”)’ to “STATUS\_INVALID\_PAIRWISE\_CIPHER” at 1978.59 and 1978.62
* ‘6 (“Unacceptable lifetime”)’ to “UNACCEPTABLE\_LIFETIME” at 1979.2
* ‘7 (“Not in same BSS”)’ to “NOT\_IN\_SAME\_BSS” at 1979.5
* “54 (i.e., Invalid MDE)”. to “STATUS\_INVALID\_MDE” at 2015.42, 2018.8 and 2019.31 and 2021.43
* “43 (i.e., Invalid AKMP)” to “STATUS\_INVALID\_AKMP” at 2015.45, 2019.34 and 2021.45
* “55 (i.e., Invalid FTE)” to “STATUS\_INVALID\_FTE” at 2015.37, 2021.47, 2025.61, 2028.52 and 2029.63
* “53 (i.e., Invalid PMKID)” to “STATUS\_INVALID\_PMKID” at 2015.39, 2021.50, 2025.62, 2028.53 and 2030.2
* “28 (i.e., R0KH unreachable)” to “R0KH\_UNREACHABLE” at 2015.40 and 2021.51
* “1s” to “ones” at 2019.41
* “19 (i.e., Invalid Pairwise Cipher)” to “STATUS\_INVALID\_PAIRWISE\_CIPHER” at 2019.42 and 2021.55.
* “0, indicating success” to “SUCCESS” at 2020.3, 2022.15, 2023.15 and 2024.17
* ‘37 (“This request has been declined”)’ to “REQUEST\_DECLINED” at 2021.52
* “54 (i.e., Invalid MDE)” to “STATUS\_INVALID\_MDE” at 2023.3, 2024.5, 2025.54, 2028.45, 2029.61 and 2031.13
* “38 (i.e., the request has not been successful as one or more parameters have invalid values)” to “INVALID\_PARAMETERS” at 2024.37
* “52 (i.e., Invalid FT Action Frame Count)” to “STATUS\_INVALID\_FT\_ACTION\_FRAME\_COUNT” at 2024.44
* “14 (i.e., Received an Authentication frame with authentication transaction sequence number out of expected sequence)” to “TRANSACTION\_SEQUENCE\_ERROR” at 2024.46
* “nonzero” to “not SUCCESS” at 2026.50, 2029.58, 2053.43, 2055.11, 2055.16 and 2055.17
* “33 (i.e., Association denied because QoS AP has insufficient bandwidth to handle another QoS STA)” to “DENIED\_INSUFFICIENT\_BANDWIDTH” at 2030.20
* “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)” to “REFUSED\_REASON\_UNSPECIFIED, TRANSACTION\_SEQUENCE\_ERROR, or REJECTED\_SEQUENCE\_TIMEOUT” at 2030.29 and 2031.34
* “0” to “SUCCESS” at 2033.38, 2034.50, 2053.6, 2053.27, 2053.29, 2053.41, 2055.8, 2138.63
* “0, 1, or 78” to “SUCCESS, REFUSED\_REASON\_UNSPECIFIED, or CANNOT\_FIND\_ALTERNATIVE\_TBTT” at 2139.2
* “ 'successful' “ to “SUCCESS” at 2864.28, 2864.42, 2864.55, 2865.2, 2865.14
* “no request outstanding” to “NO\_OUTSTANDING\_GAS\_REQUEST” at 3243.30

# CID 3068 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3068 | 698.45 | 8.4.1.54 | Numeric values should generally not be specified in binary to avoid confusion with bitstrings, which might lead to an alternative interpretation. | Replace binary values by numeric ones in Table 8-81. |

**Discussion:**

Context is:



As this is a straightforward 2-bit field, there is no reason to have a table demonstrating how to encode the 4 values in bits. Delete the table.

**Proposed resolution: Revised.**

*Change paragraph at 698.34 to:*

“If the Membership Status subfield for a particular group ID is 1, then the corresponding User Position subfield in the User Position Array field contains the group ID’s User Position.”

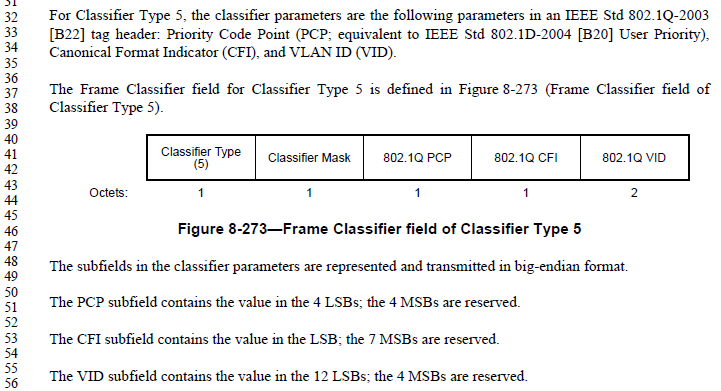
*Delete Table 8-81.*

# CIDs 3128, 3129, 3131 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3128 | 839.33 | 8.4.2.30 | (1) PCP means "PBSS Control Point" in 802.11, (2) 802.1 has abolishes CFI, replacing it in all cases with DEI, and (3) 802.1Q is being emphasized and 802.1D de-emphasized. | Replace paragraph with: "For Classifier Type 5, the clssifier parameters are the following parameters in an IEEE Std 802.1D/Q [B20]/[B22] tag header: Prioirty Code Point, Drop Eligibility Indicator (DEI), and VLAN ID (VID)." Also, add "VID" to the Acronyms. (DEI is already listed.) |
| 3129 | 839.41 | 8.4.2.30 | (1) PCP means "PBSS Control Point" in 802.11, and (2) 802.1 has abolishes CFI, replacing it in all cases with DEI. | In Figure 8-273, expand "PCP" to "Priority Code Point" and replace "CFI" with "DEI". |
| 3131 | 839.53 | 8.4.2.30 | 802.1 has abolishes CFI, replacing it in all cases with DEI. | Replace "CFI" with "DEI". |
| 3132 | 3512.58 | P.3 | 802.1 has abolishes CFI, replacing it in all cases with DEI. | Replace "CFI" with "DEI". |

**Discussion:**

Context is:



Generally agree with the commenter. 802.1D-2004 does not support CFI, and 802.1Q-2011 supports DEI. To maintain support for 802.1D networks, the classifier can be specified to use either the 802.1Q PCP or 802.1D UP, and let the other two fields be unique to 802.1Q mapping.

**Proposed resolution: Revised.**

*Change paragraph at 839.32 to:*

“For Classifier Type 5 when used to match an 802.1D-2004 [B20] frame, the classifier parameter is only the User Priority, and the DEI and VID parameters are ignored. For Classifier Type 5 when used to match an 802.1Q-2011 frame, the classifier parameters are: Priority Code Point, Drop Eligibility Indicator (DEI), and VLAN ID (VID)."

*Modify Figure 8-273 to label the third subfield as “802.1D UP/802.1Q Priority Code Point” and the fourth subfield as “802.1Q DEI”*

*Change the paragraphs starting at line 50 to:*

The 802.1D UP/802.1Q Priority Code Point subfield contains the value to be matched to the appropriate type frame header in the 4 LSBs; the 4 MSBs are reserved.

The DEI subfield contains the value to match against an 802.1Q-2011 frame header, in the LSB; the 7 MSBs are reserved. When matching an 802.1D-2004 frame header, this subfield is ignored.

The VID subfield contains the value to match against an 802.1Q-2011 frame header, in the 12 LSBs; the 4 MSBs are reserved. When matching an 802.1D-2004 frame header, this subfield is ignored.”

*Replace “CFI-0” with “DEI=0” in P.3 at 3512.58. (Note, change is CFI to DEI, and changing hyphen to equals).*

*Change “802.1d” to “802.1D” at two places (table column headings) in V.3.3.*

# CID 3391 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3391 | 1223.01 | 9 | 9.2.7 and 9.6 were written prior to 11e and do not make it clear how EDCA interacts with (de)fragmentation | Make it clear that fragmentation also operates on a per-TID basis in the case of EDCA. For example, 9.6 does not include the TID in the "information that is used by the destination STA to reassemble the MSDU or MMPDU" |

**Discussion:**

9.2.7 does mention transmission of fragments under EDCA. No further changes seem to be needed.

9.5 does have the following discussion about honoring TID for fragmentation:



9.6 does not list TID as one of the attributes of a fragment that must be used to do reassembly. Add that.

**Proposed resolution: Revised.**

*In the bullet list in 9.6, following the Sequence Control field bullet, add a new bullet:*

“Traffic identifier, for frames with a QoS Control field”

*Change the sentence at 1271.39 as follows:*

Except when using block ack, a DMG STA shall complete the transmission of a fragmented MSDU before starting transmission to the same receiver of another MSDU with the same TID ~~of~~ as the fragmented MSDU.

*Add a NOTE, at 1271.41:*

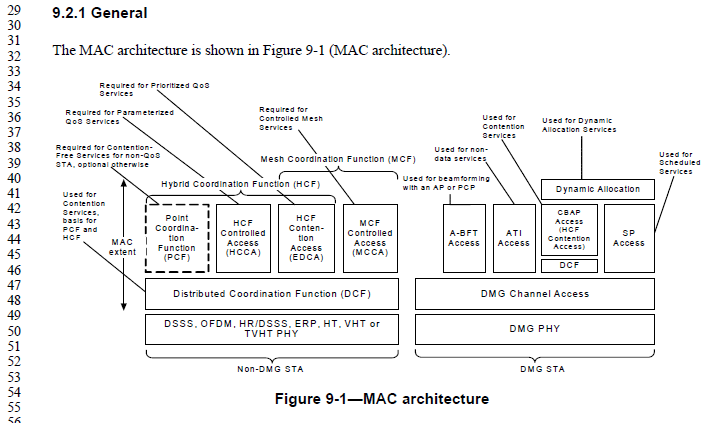
NOTE—A STA might interleave fragments of MSDUs with different TIDs sent to the same receiver, subject to any constraint caused by the number of replay counters.

# CID 3285 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3285 | 1223.43 | 9.2.1 | There is no need to save space or to reduce the number of pages. With the addition of the DMG MAC/PHY architecture this picture has become awful. Instead of squeezing everything into one picture, there should be two separate pictures. After all the DMG MAC/PHY is completely different from the other MAC/PHY. This is exactly the message that this picture conveys. Then however, there should be two separate figures.  There is only one reference to Figure 9-1 in Clause 9.4.1 on page 1263. This Clause addresses PCF and it is anyway marked as obsolete. So there is no reason to have both MAC/PHY architectures crammed into one picture. | Use Figure 9-1 from 802.11-2012 as Figure 9-1 in this revision. Add a PHY box to the 802.11-2012 figure as in the present figure. Change Figure 9-1 caption to "Non-DMG STA MAC architecture"  Add a new Figure 9-2 from the right hand part of the current (ugly) Figure 9-1. Add the following caption to this new Figure 9-2: "DMG STA MAC architecture"  Change the sentence on line 31 to read "The MAC architecture is shown in Figure 9-1 and Figure 9-2." |

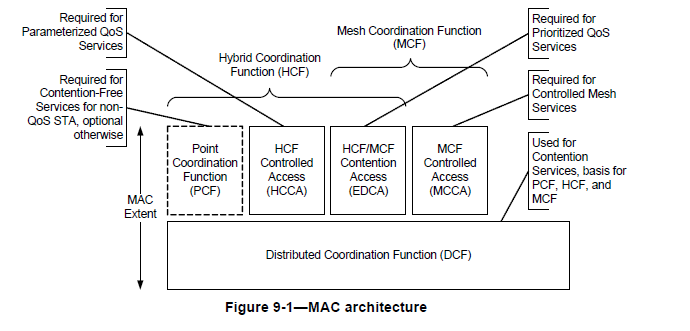
**Discussion:**

The context is:



The commenter seems be accurate, both in the references to Figure 9-1, and that these seem to be two separate architecture pictures.

The figure in 802.11-2012 looked like this:



Agree with the commenter about how to split and redraw the figures. Note that the new Figure 9-2 needs to match the (old) Figure 9-1 style (on the callouts).

**Proposed resolution: Revised.**

*Modify Figure 9-1 and add a new Figure 9-2 as described in the proposed resolution. Insert the new Figure 9-2 between the two bullet lists in 9.2.1, so just before the text “*In a DMG STA*”.*

*Note to Editor: Mark Hamilton volunteers to redraw these if desired, if the -2012 source material is provided.*

# CID 3211 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3211 | 2361.16 | 20.4.4 | aMPDUMaxLength is not defined in Table 20-25 (nor wasn't it defined in the 802.11n amendment). So what is the maximum length of an un-aggregated MPDU for the HT PHY? In Clause 16, 17, 18, and 19 this parameter is defined in the PHY characteristics table. For 802.11a and 802.11g the value is 4096 octets. However, the dot11FragmentationThreshold MIB variable limits the longest transmission to be 2346, originally for both PHYs, under the rules of Clause 9, but this was later changed to be 3000 octets in 802.11REVmb, and changed again in 802.11n to 8000 octets. With this last increase to dot11FragmentationThreshold, aMPDUMaxLength = 4096 octets became the effective limit for these two PHYs.  Now, in 802.11n, aMPUDMaxLength was not defined in the HT PHY parameters, although it is still required as one of the parameters in the initialization of the PHY primitive. aPSDUMaxLength = 65536, and the aPPDUMaxTime = 10ms were added, and the configurable dot11MaxAMSDUlength MIB variable = either 7935 or 3839 (default) was added. These 3 parameters, together, do limit the PHY transmission in both the aggregated and unaggregated cases, but then the default value for dot11FragmentationThreshold, which becomes the limit of an unaggregate packet, is ill-defined because aMPDUMaxLength is undefined for the Clause 20 HT PHY. All that is really known is that 8000 octets is the upper limit for dot11Fragmentation Threshold. (See the last sentence of the description of dot11FragmentationThreshold on page 3120, line 57, which gives the equation for calculating dot11FragmentationThreshold as a function of aMPDUMaxLength.) | Define aMPDUMaxLength for the HT PHY. In this way, the length of the maximum unaggregated MPDU (and the default maximum fragment as defined by the value of dot11FragmentationThreshold in the MIB) will be well-defined, as well as the parameter to the PHY initialization.  (Sorry for not finding this in REVmb or earlier ballots of REVmc.) |

**Discussion:**

Excerpts from e-mail exchanges with George Vlantis (the commenter):

Yes, I think we need to have a maximum MPDU length, even if the VHT PHY and 802.11ah PHY use A-MPDU formats only because of its use in the  fragmentation threshold equation (i.e. even if a non-aggregated A-MDU PHY format doesn’t exist.)

I submitted a similar comment against 802.11ah.

I can’t remember who it was, but I mentioned this to one of the MAC guys at the San Diego IEEE meeting.    He said it could be set to 4096 (which I believe is the value for 802.11a and 802.11g).

Based on this, and lack of getting a response from anyone else, recommend adding this with the value 4096.

**Proposed resolution: Revised.**

*Add a row to Table 20-25, with Characteristic “aMPDUMaxLength” and Value “4096 octets”, between the rows for aPHYConvolutionalTailLength and aPSDUMaxLength.*

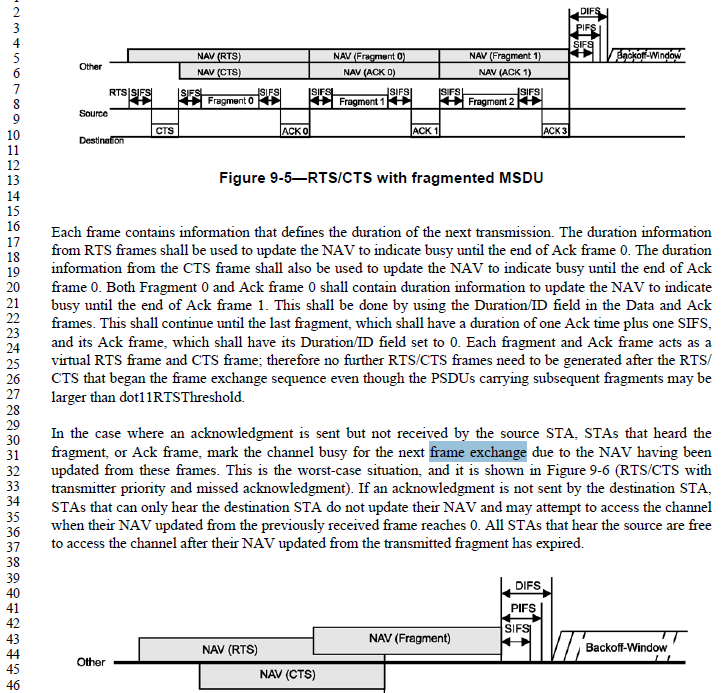
# CID 3519 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3519 | 3331.50 | G.2 | Annex G seems to say a frame exchange includes an RTS/CTS/Data/Ack/(Data/Ack)\* sequence (TXOP continuation, that is). This is not normally considered a single frame exchange. Change Annex G to not consider this a single frame exchange (it can be done as a TXOP "burst"/"continuation", but not a single frame exchange). (What does this break?) | Replace the BNF line "( [CTS | RTS CTS | PS-Poll] {frag-frame Ack} last-frame Ack ) |" with two lines: "( [CTS | RTS CTS | PS-Poll] frag-frame Ack ) |" and "( [CTS | RTS CTS | PS-Poll] last-frame Ack ) |" |

**Discussion:**

There are two points of view: 1) A frame exchange is limited as per the proposed resolution, limited to a single Data frame for example. 2) A frame exchange (sequence?) is any sequence that is (legitimately) done with only SIFS separations, but is otherwise unlimited, at least up to the TXOP limit.

Context, for the specific scenario in the comment (fragments in a stream of transmits within a TXOP), per 9.3.2.5 (RTS/CTS with fragmentation):



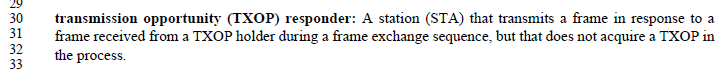
Note that this text describes the frame exchange sequence starting with the RTS/CTS (near end of first paragraph), and then describes a (singular) frame exchange as the time protected by each NAV during the continuation.

Perhaps the key is the difference between “frame exchange” and “frame exchange sequence”? It does seem clear from the text above that a frame exchange is a single Data frame (with Ack and preceded by optional RTS/CTS), while a frame exchange sequence is the string of multiple of these, especially in the scenario of transmitting a sequence of fragments.

Some other hints:



Note that this uses the plural “frame exchange sequences” within a TXOP. Pretty weak evidence, though.



This definition makes it clear that a TXOP holder transmission, followed by a responder transmission is still within a single frame exchange sequence (but it does not claim this is a single frame exchange).

From 9.2.4.2 (EDCA):



This uses clear plural language for the concept of multiple frame exchanges within a TXOP.

From 9.3.2.3.3 (SIFS):



So, it is clear that SIFS is necessary to continue to hold a TXOP across multiple frame exchange sequences. But, that is not in contention.

**Proposed resolution: Rejected.**

The particular BNF statement referenced in the comment is for frame exchange sequences per the comment (shortened to “frame sequence” in the actual BNF). Per the text in 9.3.2.5 describing transmission of fragments without releasing control of the medium, it is clear that all fragments (and their Acks) are transmitted as part of one frame exchange sequence. Thus, the BNF is correct as is.

# CID 3523 (MAC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3523 | 554.54 | 8.2.4.1.7 | The rules in 8.2.4.1.7 are not consistent with 10.2.2.2.  The concepts "MMDU is bufferable" and "PM bit is reserved" need to be separated. It makes no sense to say that an Action MMDU sent by a non-AP STA is bufferable, for example, just because you want to be able to say that the PM is valid in the MPDUs used to send it. | Consider documents 11-12/1199 and 11-13/0131 |

**Discussion:**

These documents are still a work in progress by their authors, at this time. To be considered later, if/when the documents are ready.