IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comment resolution | | | | |
| Date: 2018-01-24 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Solomon Trainin | Qualcomm |  |  | strainin@qti.qualcomm.com |
|  |  |  |  |  |

Resolution of CIDs 1101, 1645, 1647, 1648, 1735, 1765, 1767, 2135, 2207, 2208, 2209, 2210, 2211, 2245, 2266, 2273, 2274, 1725, 1734, 2206, 2265, and 2275 is presented

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Page** | **Clause** | **Comment** | **Proposed Change** |
| 1101 | 130.04 | 10.24.12.5 | The buffered MSDUs in the receive reordering buffer control can be flushed when the STA stops receiving from the <TA, TID> pair the statement "Stop receiving" need to be better defined | Flushing the buffer can be done when the Duration time indicated in the last received MPDU has elapsed |

Proposal: **Revise**

Discussion:

The proposed change is not adequate in relation to the intention of the feature as it is presented “…ability of the recipient to flush the buffers when it starts receiving from a different originator or TID.” that the transmission with no need to flush the buffer may be longer than single TXOP or SP. Proposed change resolves the commented issue and aligns the normative text with the text at start of the sub clause.

***TGay Editor: Modify the text in P130L4 as follows:***

The buffered MSDUs in the receive reordering buffer control can be flushed when the STA starts receiving from different <TA, TID> pair.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1645 | 115.35 | 10.24.2 | SAR is an optional feature as indicated by this paragraph, so it is necessary for the upper layer to do the segmentation if SAR is not supported. Should SAR be a mandatory one to truly reflect the design purpose? | Please clarify |

Proposal: **Reject**

Discussion:

The referred text is completely clear that the SAR is optional and condition of supporting the feature is defined. Relevant capability is indicated in very early stages of association by the EDMG Capabilities element so, no excessive work is required if the recipient is not capable of SAR. No more clarification is needed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1647 | 119.40 | 10.24.6 | It seems unnecessary to differentiate TID and TSID, simply use TID. Throughout the draft, some places do not mention TSID at all, only TID. Based on the definition in 5.1.1.3 of IEEE-2016, TSID just limits TID values to 8-15. | Modify as suggested in the comment |

Proposal: **Revise**

Discussion: Commenter is correct that TID is used as general term that covers TSID as well, so no need to use the TID/TSID combination.

***TGay Editor: replace TID/TSID by TID in all appearances of TID/TSID in D1.0***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1648 | 124.04 | 10.24.7.6.2 | it is better to clarify that "increasing order of MPDU\_SN" means the MPDU\_SN is increasing by 1 for each different MPDU. Meanwhile, it might be easier to say that as long as the MPDU\_SN is non-continuous, the MPDU\_SN should not be regarded as the part of the same MSDU. | Modify or clarify as suggested in the comment |

Proposal: **Revise**

Discussion:

The comment assumes that current definition means that the MPDUs SNs are increasing by 1 to enable the expected release of the MSDU from the buffer and suggest some clarification to the existent text. The proposed clarification is not correct because each MPDU contains MSDU SN it belongs to. From other side it is true that the existent text does not define that makes the MSDU complete and this definition should be added.

***TGay Editor: append sentence to end of paragraph in P124L8***

MSDU is complete if all MPDUs identified with the same MSDU\_SN from the MPDU\_SN with Start of MSDUn subfield equal to one until the MPDU with End of MSDUn subfield equal to one are arrived in the receive reordering buffer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1735 | 115.38 | 10.24.2 | Inconsistent rules on MSDU/MPDU Modulo subfield lengths. Here it says MPDU <= 12 and (MPDU+MSDU) = 14. In P85L7 it says only (MPDU+MSDU) <= 14. | Change P85L7 to match the rules at P115L38. |

Proposal: **Revise**

***TGay Editor: modify in P85L7 as follows***

The sum of the values of the MSDU Modulo subfield and the MPDU Modulo subfield shall be equal to fourteen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1765 | 115.06 | 10.24.1 | If the STA uses segmentation and reassembly, parameters for SAR need to be added to MLME-ADDBA primitives. | Add parameters relating to SAR to MLME-ADDBA primitives in 6.3.29. |

Proposal: **Revise**

Discussion:

Solution is provided in reference to [2] IEEE Std 802.11-2016. In addition, EDMG Flow control parameters are added

***TGay Editor modify as follows***

In P362, in MLME-ADDBA.request(,

In P363, in MLME-ADDBA.confirm(

In P364, in MLME-ADDBA.indication(

In P365, in MLME-ADDBA.response(

after ADDBA Extension,

insert:

EDMG Flow Control Extension Configuration

SAR Configuration

In P362, P364, P365, and P366 insert two lines in the tables after ADDBA Extension



|  |  |  |  |
| --- | --- | --- | --- |
| EDMG Flow Control Extension Configuration | EDMG Flow Control Extension Configuration  element | As defined in 9.4.2.263 | Specifies EDMG Flow Control parameters |
| SAR Configuration | SAR Configuration  element | As defined in 9.4.2.266 | Specifies segmentation parameters in relation to SAR |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1767 | Kazuyuki Sakoda | 119.23 | 10.24.6 | If the STA uses Multi-TID BA, parameters for Multi-TID need to be added to MLME-ADDBA primitives. | Add parameters relating to Multi-TID to MLME-ADDBA primitives in 6.3.29. |

Proposal: **Reject**

Discussion:

There are no Multi-TID specific parameters to be added

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2135 | 119.35 | 10.24.6 | change "....with a Starting Sequence Number subfield set to the Starting Sequence Number subfield of the BlockAck Request Starting Sequence Control subfield and the length of the BlockAck Bitmap subfield calculated as defined in 10.24.2." to ....with a Starting Sequence Number subfield set to the value of the BlockAck Request Starting Sequence Control subfield and the length of the BlockAck Bitmap subfield calculated as defined in 10.24.2." | as suggested |

Proposal: **Accept**

Discussion:

It is editorial comment however it is indicated as technical

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2207 | 118.32 | 10.24.4 | How is it know that a block ack agreement is using segmentation and reassembly? Doesn't this need to point to a field so it is clear? | Clarify the text so it is clear as to when a block ack agreement is using segmentation and reassembly. Is it: "An HT-immediate block ack agreement in which the SAR Configuration element was included in the both the ADDBA Request and ADDBA Response frames and that had the SAR Enabled subfield set to one in both frames is considered as a block ack agreement with segmentation and reassembly. That is, a block ack agreement with segmentation and reassembly is an HT-immediate block ack agreement." as taken from 10.24.7.1.  It may be best to separate the current BA behavior from the BA for segmentation and reassembly behavior, by creating a separate clause. |

Proposal: **Reject**

Discussion:

Concern of the comment “How is it know that a block ack agreement is using segmentation and reassembly? Doesn't this need to point to a field, so it is clear?” does not make sense, because the relevant definition already exists in the draft that is also mentioned in the proposed change column.

The part of proposed change “to separate the current BA behavior from the BA for segmentation and reassembly behavior, by creating a separate clause” is very general and fails to identify a specific issue to be addressed. It fails to identify changes in sufficient detail so that the specific wording of the changes that will satisfy the commenter can be determined.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2208 | 120.07 | 10.24.7.1 | What is the point of the sentence: "That is, a block ack agreement with segmentation and reassembly is an HT-immediate block ack agreement." It doesn't seem to provide any useful information as this whole section is related to HT-immediate block ack. | delete: "That is, a block ack agreement with segmentation and reassembly is an HT-immediate block ack agreement." |

Proposal: **Accept**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2209 | 121.19 | 10.24.7.3 | I think it would be cleaner and less confusing if the various cases were separated. Hence, I think 10.24.7.3 would become Scoreboard context control during full-state operation without segmentation and reassembly. And 10.24.7.4 would also state it is without segmentation and reassembly. A new sections 10.24.73a and 10.24.7.4a could be added for the case with segmentation and reassembly. | Remove the changes to 10.24.7.3 and 10.24.7.4 and move them to new clauses 10.24.7.3a and 10.24.7.4a which define the operation when segmentation and reassembly are present. |
| 2210 | 121.39 | 10.24.7.4 | I think it would be cleaner and less confusing if the various cases were separated. Hence I think 10.24.7.3 would become Scoreboard context control during full-state operation without segmentation and reassembly. And 10.24.7.4 would also state it is without segmentation and reassembly. A new sections 10.24.73a and 10.24.7.4a could be added for the case with segmentation and reassembly. | Remove the changes to 10.24.7.3 and 10.24.7.4 and move them to new clauses 10.24.7.3a and 10.24.7.4a which define the operation when segmentation and reassembly are present. |

Proposal: **Reject**

Discussion:

The disputed sub clauses 10.24.7.3 and 10.24.7.4 provide definition of Scoreboard context control. Distinct definition of few variables to distinguish between Block Ack with SAR and w/o SAR allows keeping the behavioural text common for both cases. The proposed change is very general and fails to identify a specific issue to be addressed. It fails to identify changes in sufficient detail so that the specific wording of the changes that will satisfy the commenter can be determined. If taking literally it leads to duplication of big parts of equal text that no need for that is justified.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2211 | 122.27 | 10.24.7.6.1 | A statement that there are now two types of reordering buffers should be made and then the two reordering buffers described. The way it is now is confusing. | State that there are two types of reordering buffers, one with out segmentation and reassembly and one with segmentation and assembly. Then start each of the sections: "For each HT-immediate block ack agreement that does not use segmentation and reassembly a receive reordering buffer shall be maintained."  and  "For each HT-immediate block ack agreement that uses segmentation and reassembly a receive reordering buffer shall be maintained." |

Proposal: **Revised**

Discussion:

Separation of SAR rules and no SAR rules is defined as suggested in the proposed changes.

***TGay Editor modify as follows***

In [1] IEEE P802.11ay/D1.0, November 2017

In P122L27

For each HT-immediate block ack agreement that does not use segmentation and reassembly a receive reordering buffer shall be maintained for each HT-immediate block ack agreement Each receive reordering buffer includes a record comprising the following:

In P122L31

For each HT-immediate block ack agreement that uses segmentation and reassembly a receive reordering buffer shall be maintained for each block ack agreement that uses segmentation and reassembly. Each receive reordering buffer includes a record comprising the following:

In P123L10

For each HT-immediate block ack agreement that uses segmentation and reassembly WinStart and WinEnd variable shall be initialized as follows:

In P123L18

Move to end of the sub clause sentence “Any MSDU that has been passed up to the next MAC process shall be deleted from the receive reordering buffer.”

In P123L20

For each HT-immediate block ack agreement that uses segmentation and reassembly the recipient shall always pass MSDUs or A-MSDUs up to the next MAC process in order of increasing 20 MSDU Sequence Number subfield value.

In reference to [2] EEE Std 802.11-2016

In P1426

For each HT-immediate block ack agreement that does not use segmentation and reassembly WinStart and WinEnd variable shall be initialized as follows:

*WinStartB* is initialized to the Starting Sequence Number subfield value of

Remove sentence “Any MSDU or A-MSDU that has been passed up to the next MAC process shall be deleted from the receive reordering buffer.”

In P1427

For each HT-immediate block ack agreement that does not use segmentation and reassembly the recipient shall always pass MSDUs or A-MSDUs up to the next MAC process in order of increasing Sequence Number subfield value.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2245 | 116.03 | 10.24.2 | It is not clear whether a recipient can accept a BA agreement by setting SAR enabled to 0 (or SAR config element not present) and accepts/modifies the Block Ack Parameter set in ADDBA response | should not allow such behavior  change to "A recipient may reject the BA agreement ..." |

Proposal: **Reject**

Discussion:

A recipient may reject the ADDBA request by setting the Status code in the ADDBA response frame to anything but SUCCESS. From the other side the originator may reject the Block Ack agreement by issuing DELBA frame to the recipient. There are multiple parameters that may be negotiated over Block Ack agreement so, no need is seeing to require rejection of Block Ack agreement establishment in case of recipient rejection support of SAR.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2266 | 115.40 | 10.24.2 | Not sure there is a use case for MSDU Modulo > MPDU Modulo | add a requirement that MPDU modulo >=7 |

Proposal: **Reject**

Discussion:

The comment is very general and fails to identify a specific issue to be addressed. No justification of proposed value.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2273 | 124.11 | 10.24.7.6.2 | There is no procedures related to WinStartJ other than setting the value | Specify the procedure which needs to check the value of this variable, or remove this variable |

Proposal: **Revised**

***TGay Editor remove***

P124L11

4) Set WinStartJ to the value of the MSDU\_SN subfield of the MSDU that was passed up to the next MAC process plus one.

P123L8

A WinStartJ parameter, indicating the value of the MSDU Sequence Number subfield of the first (in order of ascending sequence number) MSDU that has not yet been received.

P123L16

WinStartJ is initialized to the MSDU Starting Sequence Number subfield value of the ADDBA Request frame that elicited the ADDBA Response frame that established the block ack agreement.

***TGay Editor modify***

P124L13

Change number to 4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2274 | 125.06 | 10.24.7.6.3 | Should be + | change to + |

Proposal: **Revised**

***TGay Editor change*** in P125L6

2) Set WinEndB = WinStartB +WinSizeB – 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1725 | 115.06 | 10.24.1 | It appears that segmentation allows multiple MSDUs to be "in process" simultaneously (that is, segments from different MSDUs can be mixed together in time while being transmitted). If this is true, then what happens when the modulo counter(s) wrap? | Add text explaining what happens if either of the MSDU Sequence Number or MPDU Sequence Number wraps, due to the modulo operation. |
| 1734 | 115.07 | 10.24.1 | Missing verb "modulo" | Change to "operations ... are performed \_modulo\_ the MSDU Modulo and MPDU Modulo, respectively." Better yet, might be to rename MSDU/MPDU Modulo to something else (like "Max") so we don't have the phrase "modulo MSDU Modulo" which is really hard to parse. |

Proposal: **Revise**

Discussion:

Text that explains the modulo operation for MSDU Sequence Numbers and MPDU Sequence Numbers is provided.

***TGay Editor: modify in P115L6 as follows***

Under a block ack agreement using segmentation and reassembly, operations on MSDU Sequence Number and MPDU Sequence Number are performed modulo MSDU\_Modulo and modulo MPDU\_Modulo respectively (see 10.62), where MSDU\_Modulo and MPDU\_Modulo are as defined in the SAR Configuration element. Operations on MPDU sequence number and MSDU sequence number are performed modulo 2MPDU\_Modulo and 2MSDU\_Modulo respectively. Comparisons between MPDU sequence number and MSDU sequence number are circular modulo 2MPDU\_Modulo and 2MSDU\_Modulo respectively, i.e., the sequence number space is considered divided into two parts, one of which is “old” and one of which is “new,” by means of a boundary created by adding half the sequence number range to the current start of receive window (modulo 2MPDU\_Modulo and 2MSDU\_Modulo respectively).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2206 | 115.06 | 10.24.1 | It is not clear that the Otherwise, also includes the rest of the paragraph. Also since the current paragraph is refereeing to all block ack agreements as per the section, it seems very out of place to stick the modulo requirement in the middle of the discussion on how block ack agreements work for all cases except those using segmentation and reassembly. It would be cleaner to leave the current description in place with a statement that for cases where segmentation and reassembly are used see section xxx. Also it may be beneficial to provide a new Block acknowledgement clause for Block acknowledgement for segmentation and reassembly. | Remove all the changes from 10.24.1 and add a new clause 10.xx Block acknowledgement for segmentation and reassembly. |

Proposal: **Revised**

Discussion:

In the basic spec [2] definition of the modulo operation in relation to the sequence numbers is defined in the sub clause 10.24.1. Resolution to the CID1734 provides rules for modulo operation in relation to the SAR parameters MSDU\_modulo, and MPDU\_modulo. Remaining part of the sub clause 10.24.1 is common for BlockAck so, no need to duplicate the rules in new sub clause.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2265 | 116.13 | 10.24.2 | Not clear why it is 2^(MPDU\_Modulo-2) instead of 2^(MPDU\_Modulo-1) | change to 2^(MPDU\_Modulo-1) |

Proposal: **Reject**

Discussion:

The reason of limiting MPDU Buffer Size <= 2^(MPDU\_Modulo-2) is to keep space of the “new” and “old” sequence numbers as required by the modulo operation descripted in 10.24.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2275 | 125.32 | 10.24.7.7 | The sentence on L28 allows the MPDU transmission for SN>WinStart\_O+WinSize\_O-1, but the corresponding sentence for SAR on L32 prohibits this. There are no other places specifying how to move WinStart\_O(J) forward | Allow transmission of MPDU\_SN / MSDU\_SN >WinStart\_O(J)+WinSize\_O(J)-1 |

Proposal: **Revised**

Discussion: relevant changes are provided to resolve both commented cases

***TGay Editor change*** in P125L32

P116L17

The originator may set the MSDU Buffer Size subfield to a value greater than 2MSDU Modulo-2 only if …

P125L32

Under a block ack agreement with segmentation and reassembly, the originator may transmit QoS Data frames with a TID matching an established block ack agreement in any order provided that their MPDU\_SN and MSDU\_SN lie within the current transmission window. The originator may transmit an MPDU with a MSDU\_SN that is beyond the current transmission window (MSDU\_SN > WinStartOJ + WinSizeOJ – 1), in which case the originator’s transmission window (and the recipient’s window) is moved forward. The originator shall not transmit MSDUs that are lower than (i.e., SN < WinStartOJ) the current transmission window.

NOTE: This rule guarantees that delivering of MPDU with MPDU\_SN>WinEndO will not lead to misdetection of lost MSDUs at the recipient MAC SAP thus preventing to appear MSDUs with subsequent MSDU\_SN after previous MSDUs are lost. In case of missing MSDU it will be always gap in the MSDU\_SN.

***TGay Editor append at end of subclause 10.24.7.8 Maintaining block ack state at the originator***

Under a block ack agreement with segmentation and reassembly, the originator shall update WinStartOJ and WinEndOJ at arrival of BlockAck frame. At each subsequent MPDU sent with End of MSDUn subfield set to one in the Sequence Control field the WinStartOJ shall be set to MSDU\_SN+1 and WinEndOJ shall be set to WinStartOJ+WinSizeOJ – 1 if following condtions are met

* the MPDU is indicated as acknowledged in the BlocAck bitmap
* all preceding MPDUs starting from MPDU\_SN with Start of MSDUn subfield set to one in the Sequence Control field are indicated as successfully delivered

and shall not be changed otherwise.

**References:**

1. IEEE P802.11ay/D1.0, November 2017
2. IEEE Std 802.11-2016