IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comment resolution | | | | |
| Date: 2018-03-06 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Solomon Trainin | Qualcomm |  |  | strainin@qti.qualcomm.com |
|  |  |  |  |  |

Resolution of CIDs 2245, 2266, 2273, 2274, 1725, 1734, 2206, 2265, and 2275 is presented

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Page** | **Clause** | **Comment** | **Proposed Change** |
| 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: **Revised**

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. One clarification is needed to the case the agreement is established w/o SAR support however the SAR configuration element is present in the ADDBA request frame.

***TGay editor***

***In P116L5 append***

A recipient may reject the SAR configuration by setting the SAR Enabled field within the SAR Configuration element in the ADDBA Response to zero. The originator shall set to zero Sequence Number field in the MPDU it transmits first under the established Block Ack agreement in case that the Recipient rejects the SAR configuration and responds with Status code set to SUCCESS in the ADDBA response frame,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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: **Revised**

Discussion:

Due to that the defined solution keeps the general rule that no MPDUs with equal SN numbers are allowed in the Rx buffer the proposal is relevant.

***TGay Editor change*** in P115L38

…the value of the MPDU Modulo subfield shall be more than or equal to 7, and less than or equal to 12…

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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