IEEE P802.11  
Wireless LANs

|  |
| --- |
| **Resolutions to LB230 comments submitted to**  **subclauses 9.3.1.8 and 10.24.6** |
| **Date:** 2018-04-27 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tomoko Adachi | Toshiba | 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, Japan | +81 44 549 2283 | tomo.adachi@toshiba.co.jp |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Abstract

This submission proposes resolutions for the following CIDs to subclauses 9.3.1.8 and 10.24.6 (**4 CIDs**):

* 11498, 14323
* 13659
* 11056

Interpretation of a Motion to Adopt

A motion to approve this submission means that the editing instructions and any changed or added material are actioned in the TGax Draft. This introduction is not part of the adopted material.

***Editing instructions formatted like this are intended to be copied into the TGax Draft (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents).***

***TGax Editor: Editing instructions preceded by “TGax Editor” are instructions to the TGax editor to modify existing material in the TGax draft. As a result of adopting the changes, the TGax editor will execute the instructions rather than copy them to the TGax Draft.***

# 9.3.1.8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Commenter** | **PP.LL** | **Comment** | **Proposed Change** | **Resolution** |
| 11498 | Chunyu Hu | 75.54 | A mechanism is needed which allows the transmitter to request the receiver(s) to flush outstanding fragamented frames RX buffer. The idea/motivation/need is similar to the existing mechanism of using BAR to move the BA window. It can be selective in that the receipient only flushes incomplete MSDUs up to and including the indicated end sequence number. It can be also requesting flush all. No receiver window move occurs. No BlockAck is transmitted in response. MSDUs and fragments of MSDUs that are not covered by the SEQ number range or which are completely assembled in the buffer are unaltered. Adding this function to the BlockAckReq seems to be a good option. | as in the comment | Rejected.  To make the receipient flush the imcomplete MSDUs up to the intended sequence number, the originator can do so by the original rule, i.e., by setting the SSN in the BlockAckReq. To flush all the MSDUs at the receipient, the originator can do so by the original rule, i.e., by setting the SSN in the BlockAckReq higher than the largest sequence number of the MSDUs that it transmitted.  If the receiver window is not changed after the flush as the commenter suggested, it means that the originator can send fragments again and if as a result, the MSDU is completed and passed to the next MAC process, the MSDU reordering will get messed up. |
| 14323 | Zhou Lan | 75.54 | A transmitter-commanded RX buffer flushing mechanism is needed. It is selective in that the receipient only flushes incomplete MSDUs up to and including the indicated end sequence number. No receiver window move occurs. No BlockAck is transmitted in response. MSDUs and fragments of MSDUs that are not covered by the SEQ number range or which are completely assembled in the buffer are unaltered. Adding this function to the BlockAckReq. | as in the comment | Rejected.  To make the receipient flush the imcomplete MSDUs up to the intended sequence number, the originator can do so by the original rule, i.e., by setting the SSN in the BlockAckReq. To flush all the MSDUs at the receipient, the originator can do so by the original rule, i.e., by setting the SSN in the BlockAckReq higher than the largest sequence number of the MSDUs that it transmitted.  If the receiver window is not changed after the flush as the commenter suggested, it means that the originator can send fragments again and if as a result, the MSDU is completed and passed to the next MAC process, the MSDU reordering will get messed up. |
|  |  |  |  |  |  |
| 13659 | Tomoko Adachi |  | In the baseline, the BlockAck frame variant encoding and the BlockAckReq frame variant were aligned. As the BlockAck frame variant encoding has been changed from the baseline, the BlockAckReq frame variant encoding should be also changed.  The old subfield names should be also searched through the baseline and the draft and should be updated. | As in comment. | Revised.  See the instructions to the TGax editor in doc. 11-18/0733.  Note to the TGax editor: subclause 10.24.6 was modified by 11ak draft but it is not reflected in 11ax D2.3. The track changes shown in doc.11-18/0733 is based on the text in 11ax D2.3 but 11ak modification should be incorporated and changes should be made thereto. |

# 10.24.6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Commenter** | **PP.LL** | **Comment** | **Proposed Change** | **Resolution** |
| 11056 | Adrian Stephens | 202.44 | "The B2 bit of the BA Type subfield of the BA Control field shall be set to 1" -- this cries out for this bit to be named. Magic numbers in the text are inherently evil. | Name bit 2 at the point of definition once, and refer to it here.  Name bits 3-4 as a subfield at the point of definition and refer to the name here. | Revised.  In the past there were names for bit 2 and bits 3-4, but there was an agreement to change the encoding of bit fields to make better use of them. So, bringing back the names is not appropriate.  See the instructions to the TGax editor in doc. 11-18/0733. |

#### 9.3.1.8 BlockAckReq frame format

##### 9.3.1.8.1 Overview

Change the 4th paragraph as follows:

The TA field value is the address of the STA transmitting the BlockAckReq frame or a bandwidth signaling TA. In a BlockAckReq frame transmitted by a VHT STA or an HE STA in a non-HT or non-HT duplicate format and where the scrambling sequence carries the TXVECTOR parameter CH\_BAND-WIDTH\_IN\_NON\_HT, the TA field value is a bandwidth signaling TA.

TGax Editor: Insert Figure 9-27 from the baseline, P802.11ak D6.0, and change as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | B0 | ~~B1~~ | ~~B2~~ | ~~B3~~ | ~~B4~~ | B1 | B4 | B5 | B11 | B12 | B15 |
|  | BAR Ack Policy | ~~Multi-TID~~ | ~~Compressed~~  ~~Bitmap~~ | ~~GCR Mode~~ | | BAR Type | | Reserved | | TID\_INFO | |
| Bits: | 1 | ~~1~~ | ~~1~~ | ~~2~~ | | 4 | | 7 | | 4 | |

**Figure 9-27—** **BAR Control field(#13659)**

TGax Editor: Insert the 7th paragraph from the baseline, P802.11ak D6.0, and change as follows:

The BAR Type subfield(#13659) indicates which of the possible BlockAckReq frame variants is used, as indicated in Table 9-22 (BlockAckReq frame variant encoding).

TGax Editor: Insert Table 9-22 from the baseline, P802.11ak D6.0, and replace with the following:

**Table 9-22—** **BlockAckReq frame variant encoding(#13659)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
|  | |  |  | |  |
| **BAR Type** | **BlockAckReq frame variant** | | |
| 0 | Basic BlockAckReq | | |
| 1 | Extended Compressed BlockAckReq | | |
| 2 | Compressed BlockAckReq | | |
| 3 | Multi-TID BlockAckReq | | |
| 4 | Reserved | | |
| 5 | Reserved | | |
| 6 | GCR BlockAckReq | | |
| 7 | Reserved | | |
| 8 | Reserved | | |
| 9 | Reserved | | |
| 10 | GLK-GCR BlockAckReq | | |
| 11 | Reserved | | |
| 12 | Reserved | | |
| 13 | Reserved | | |
| 14 | Reserved | | |
| 15 | Reserved | | |

TGax Editor: Change texts under 10.24.6 in P802.11ax D2.3 as follows (Also see note in the resolution column for CID 13659):

### 10.24.6 Selection of BlockAck and BlockAckReq variants

Change as follows:

The Basic BlockAck and Basic BlockAckReq variants shall be used for all BlockAck and BlockAckReq frames sent where a block ack agreements exists but the agreement is not an HT-immediate agreement and shall not be used otherwise.(#13659, #11056)

The Multi-TID BlockAck variant shall be used for all BlockAck frames related to an HT-immediate agreement transmitted inside a PSMP sequence and shall not be used otherwise. The Multi-TID BlockAckReq variant shall be used for all BlockAckReq frames related to an HT-immediate agreement transmitted inside a PSMP sequence and shall not be used otherwise.(#13659, #11056)

In a DMG BSS, if the Compressed BlockAckReq variant is used related to an HT-immediate agreement, then all of the following BlockAck and BlockAckReq frames transmitted as part of the HT-immediate agreement shall use the Compressed BlockAck and Compressed BlockAckReq variants.(#13659, #11056)

In a DMG BSS, if the Extended Compressed BlockAckReq variant is used related to an HT-immediate agreement, then all of the following BlockAck and BlockAckReq frames transmitted as part of the HT-immediate agreement shall use the Extended Compressed BlockAck and Extended Compressed BlockAckReq variants.(#13659, #11056)

Where the terms BlockAck and BlockAckReq are used within 10.24.7 and 10.24.8, the appropriate variant according to this subclause (e.g., Compressed, Multi-TID) is referenced by the generic term. The BAR Type(#13659) subfield of the BAR Control field shall be set to GCR BlockAck(#13659, #11056) in all BlockAckReq frames where the block ack agreement is for a group address delivered using the GCR block ack retransmission policy and shall be set to 0 otherwise. The BA Type(#13659) ~~GCR~~ subfield of the BA Control field shall be set to GCR BlockAck(#13659, #11056) in all BlockAck frames where the block ack agreement is for a group address delivered using the GCR block ack retransmis-sion policy and shall be set to 0 otherwise.