IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC36 comment resolution subclause 35.3.7.2 | | | | |
| Date: 2021-10-22 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Liwen Chu |  |  |  | Liwen.chu@nxp.com |

Abstract

This submission proposes resolutions for multiple comments related to TGbe D1.0 with the following CIDs:

4111, 5167, 7603, 7604, 7605, 4119, 5726, 4746, 5146, 5688,

6489, 7371, 7372, 7413, 7602, 5924, 5988, 6465, 6490, 6623,

6626, 6990

Revisions:

* Rev 0: Initial version of the document.

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 TGbe Draft. This introduction is not part of the adopted material.

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

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **PP** | **LL** | **Comment** | **Proposed Change** | Resolution |
| 4111 | 262 | 37 | Clause 35.3.7.2 should be moved out of MLO clause (i.e., 35.3) since the contents apply to all EHT STAs | Make clause 35.3.7.2 a subclause of clause 35. Move the contents of clause 35.3.7.1.1 (General) to clause 35.3.7 and remove the titles 35.3.7.1.1 General and 35.3.7.1 Multi-link BlockAck procedure. Rename clause 35.3.7 as Multi-link BlockAck procedure. | Revised.  Discussion: Some content (single transmitting buffer within a MLD) in 35.3.7.2 is related to MLD operation and some content (new BA bitmap size) in 35.3.7.2 is related to link level operation.  TGbe editor to make changes in 11-21/1601r4 under CID 4111. |
| 5167 | 262 | 58 | "An EHT AP shall not transmit a Multi-STA BlockAck frame that contains a BlockAck Bitmap field with length equal to 512 or 1024 bits as a response to an HE TB PPDU generated by at least one HE STA"  This text should not be in the MLD Block Ack procedure (35.3.7). Move this text to the geneneral EHT AP behavior | As in the comment | Revised  Generally agree with the commenter.  TGbe editor to make changes in 11-21/1601r4 under CID 5167. |
| 7603 | 262 | 58 | "An EHT AP shall not transmit a Multi-STA BlockAck frame that contains a BlockAck Bitmap field with length equal to 512 or 1024 bits as a response to an HE TB PPDU generated by at least one HE STA." It is better to determine not to use 512 or 1024 bits when the EHT AP requested an HE TB PPDU by a triggering frame that it transmits. It is simpler. And this statement is not limited under MLO. It can be applied to all EHT STAs. Suggests to generate a new subclause 36.X for EHT acknowledgment procedure and transplant this cited text after modifying as above. | As in comment. | Revised.  Discussion: The text related EHT STA/AP or STA/AP affiliated with MLD will be moved to a new ssubclause.  TGbe editor to make changes in 11-21/1601r4 under CID 7603. |
| 7604 | 262 | 42 | "The EHT acknowledgment procedure builds on the features defined for HT-immediate block ack (see 10.25.6 (HT-immediate block ack extensions)) and HE acknowledgement (see 26.4 (HE acknowledgment procedure)), with the following extensions: --Support for BlockAck Bitmap field lengths of 512 and 1024" This statement is not limited under MLO. It can be applied to all EHT STAs. | Generate a new subclause 36.X for EHT acknowledgement procedure and transplant this cited text there. | Revised  Generally agree with the commenter. The text related EHT STA/AP or STA/AP affiliated with MLD will be moved to a new ssubclause.  TGbe editor to make changes in 11-21/1601r4 under CID 7604. |
| 7605 |  |  | The description here is not limited under MLO. It can be applied to all EHT STAs. | Generate a new subclause 36.X for EHT acknowledgement procedure and transplant the content of this subclause there. By this change, 35.3.7.2 will have a single child subclause, 35.3.7.2.1. So, remove the subclause title 35.3.7.2.1 and transplant the content of 35.3.7.2.1 under 35.3.7.2. | Revised  Generally agree with the commenter. The text related EHT STA/AP or STA/AP affiliated with MLD will be moved to a new ssubclause.  TGbe editor to make changes in 11-21/1601r4 under CID 7605. |
|  |  |  |  |  |  |
| 4119 | 262 | 6 | D1.0 defines that recipient MLD STA can request information of reception of successful transmission of other STA(s). If each link (each STA in the MLD) manages such reception status individually, it should be required for some function to exchange information of reception of successful transmission between the STAs. This is too complicated. Therefore, it should be clarified that not each STA but the MLD should centrally manage the status of reception for each link as well as transmission status. | It should clarify the mechanism of recipient MLD in order to control Block Ack adequately. | Revised  Discussion: 11beD1.0 allows a recipient MLD to transmit BA in one link to optionally carry the acknowledgement information of frames received in another link. It is up to the recipient MLD to decide the method to implement the scoreboard context (e.g. per link scoreboard contect or MLD level scoreboard context).  In the current BA creation rules, the bits in BA bitmap whose related SNs are larger than WinEndR need to be set to 0 and the bits in BA bitmap whose related SNs are less than WinStartR are set to either 1 or 0. When other links’ acknowledgement information is carried in BA, the WinStartR, WinEndR should be redefined, e.g. when the difference between WinStartR in the link where the BA will be transmitted and WinEndR in another link whose acknowledgement is optionally transmitted is more than the negotiated BA bitmap size (64 when the negotiated buffer size is no more than 64, 256 when the negotiated buffer size is more than 64 and no more than 256…).  In 11baseline, when the SN of a received frame is more than WinStartR + 2^11 and less than WinStartR, the scoreboard context is not changed. With multiple links being used, the acknowledgement information may be wrong. The following is an example:  The AP MLD sends A-MPDU1 with SNs from 0 to 1023 in link 1 and receives BA where all the frames are acknowledged. The AP MLD sends A-MPDU2 with SNs from 1024 to 2047  in link 2 and receives BA where all the frames are acknowledged. The AP MLD sends A-MPDU3 with SNs from 2048 to 3071  in link 2 and receives BA where all the frames are acknowledged. The AP MLD sends A-MPDU4 with SNs from 3072 to 4095  in link 1. The STA of destined non-AP MLD in link 1 doesn’t updates its scoreboard context since the SNs are less than WinStartR and are >=WinStartR + 2048. The frames in A-MPDU4 will never be acknowledged. After discarding the frames in A-MPDU4, the AP MLD transmits A-MPDU5 with SNs from 0 to 1023 where only one frame is correctly received. However the non-AP MLD responds with BA with all 1s in its BA bitmap…  .  TGbe editor to make changes in 11-21/1601r4 under CID 4119 |
| 5726 | 262 | 6 | D1.0 defines that recipient MLD STA can request information of reception of successful transmission of other STA(s). If each link (each STA in the MLD) manages such reception status individually, it should be required for some function to exchange information of reception of successful transmission between the STAs. This is too complicated. Therefore, it should be clarified that not each STA but the MLD should centrally manage the status of reception for each link as well as transmission status. | It should clarify the mechanism of recipient MLD in order to control Block Ack adequately. | Revised  Discussion: 11beD1.0 allows a recipient MLD to transmit BA in one link to optionally carry the acknowledgement information of frames received in another link. It is up to the recipient MLD to decide the method to implement the scoreboard context (e.g. per link scoreboard contect or MLD level scoreboard context).  In the current BA creation rules, the bits in BA bitmap whose related SNs are larger than WinEndR need to be set to 0 and the bits in BA bitmap whose related SNs are less than WinStartR are set to either 1 or 0. When other links’ acknowledgement information is carried in BA, the WinStartR, WinEndR should be redefined, e.g. when the difference between WinStartR in the link where the BA will be transmitted and WinEndR in another link whose acknowledgement is optionally transmitted is more than the negotiated BA bitmap size (64 when the negotiated buffer size is no more than 64, 256 when the negotiated buffer size is more than 64 and no more than 256…).  In 11baseline, when the SN of a received frame is more than WinStartR + 2^11 and less than WinStartR, the scoreboard context is not changed. With multiple links being used, the acknowledgement information may be wrong. The following is an example:  The AP MLD sends A-MPDU1 with SNs from 0 to 1023 in link 1 and receives BA where all the frames are acknowledged. The AP MLD sends A-MPDU2 with SNs from 1024 to 2047  in link 2 and receives BA where all the frames are acknowledged. The AP MLD sends A-MPDU3 with SNs from 2048 to 3071  in link 2 and receives BA where all the frames are acknowledged. The AP MLD sends A-MPDU4 with SNs from 3072 to 4095  in link 1. The STA of destined non-AP MLD in link 1 doesn’t updates its scoreboard context since the SNs are less than WinStartR and are >=WinStartR + 2048. The frames in A-MPDU4 will never be acknowledged. After discarding the frames in A-MPDU4, the AP MLD transmits A-MPDU5 with SNs from 0 to 1023 where only one frame is correctly received. However the non-AP MLD responds with BA with all 1s in its BA bitmap…  .  TGbe editor to make changes in 11-21/1601r4 under CID 5726 |
| 4746 | 263 |  | This subclause introduces how the responder can signal/indicate the reduced bitmap size. It's a good effort. However, there should be also a signaling for the A-MPPDU transmitter to signal what's the execpted (reduced) bitmap size to reduce the BA size; doing so also help the NAV setting in the PPDU carrying the A-MPDU and the actual used time to be consistent. | As commented | Rejected  Discussion: introducing such signaing makes the implementation of the recipient of A-MPDU complicated. |
| 5146 | 262 | 59 | The current text is restricting the usage of 512 or 1024-bit bitmap even when it can be used. It may not be the intention. When an HE TB PPDU is generated by HE STAs and EHT STAs, 512 or 1024-bit bitmap can be used in the individually addressed RU for EHT STA. | Add an exception "except that an EHT AP may transmit a Multi-STA BlockAck frame that contains a BlockAck Bitmap field with length equal to 512 or 1024 in an individually addressed RU". | accepted |
|  |  |  |  |  |  |
| 5688 | 262 | 56 | Change 'the responding MLD' to 'the peer responding MLD'. | As in comment | Revised  TGbe editor to make changes in 11-21/1601r4 under CID 5688 |
| 6489 | 262 | 49 | Need a definition for Initiating MLD | as in comment | Revised  TGbe editor to make changes in 11-21/1601r4 under CID 6489 |
| 7371 | 262 | 48 | What is an "initiating MLD"? | Change the term "initiating MLD" to "transmitting MLD". | Revised  TGbe editor to make changes in 11-21/1601r4 under CID 7371 |
| 7372 | 262 | 49 | What is a "responding MLD"? | Change the term "responding MLD" to "receiving MLD". | Revised  TGbe editor to make changes in 11-21/1601r4 under CID 7372 |
| 7413 | 262 | 54 | Add "peer" in front of "responding MLD" in order to clarify the relationship between the initiaing MLD and the responding MLD. | As in comment. | Revised  TGbe editor to make changes in 11-21/1601r4 under CID 7413 |
| 7602 |  |  | The "initiating MLD" should be an "originator MLD" and the "responding MLD" should be a "recipient MLD" throughout this subclause. | As in comment. | Revised  TGbe editor to make changes in 11-21/1601r4 under CID 7602 |
|  |  |  |  |  |  |
| 5924 | 263 | 9 | In 11ax "The length of the Block Ack Bitmap subfield ... but shall be sufficient to include the recipient's scoreboard state for MPDUs beginning with the MPDU for which the Sequence Number subfield value is WinStartR". This limits the opportunities for recepient to reduce the bitmap length as described in Table 35-1 because most likely the received MPDU SN are closer to WinEndR | allow recipient to select a SSN>WinStartR in compressed BA and MBA | Rejected.  Discussion: The text chalendged by the commenter is that the BA bitmap of the responding BA needs to carry at least the BA bimap identified by WinStartR to WinEndR:  … may be less than the  maximum allowed Block Ack Bitmap but shall be sufficient to include the recipient’s scoreboard state for  MPDUs beginning with the MPDU for which the Sequence Number subfield value is *WinStartR* and ending  with a successfully received MPDU for which the Sequence Number subfield is less than or equal to  *WinEndR*.  The proposed change should be done in 11me. The commenter is encouraged to submit the comment to 11me. |
| 5988 | 262 | 48 | default TID to link mapping is missing from the paragraph. | Add it. | Revised  TGbe editor to make changes in 11-21/1601r4 under CID 5988 |
| 6365 | 262 | 54 | Please fix "subjected to" to "subject to" in below text. "An initiating MLD shall maintain a single transmission window for each block ack agreement negotiated with the responding MLD to submit MPDUs for transmission across links subjected to the TID to link" | as in comment | Accetped |
|  |  |  |  |  |  |
| 6490 | 262 | 62 | It is not clear who initiates the BA agreement. Does every affiliated STA negotiate a separate agreement? | clarify | Rejected  Discussion: it is the originator MLD sends the ADDBA Request through it affiliated STA/AP. The BA agreement is MLD level agreement. |
| 6623 | 262 | 58 | The reason why we need this sentence "An EHT AP shall not transmit a Multi-STA BlockAck frame that contains a BlockAck Bitmap field with length equal to 512 or 1024 bits as a response to an HE TB PPDU generated by at least one HE STA." to handle HE is due to the reason that we do not introduce enough mechanism for future extension in 11ax. To avoid the same problem happening in Wi-Fi 8 again, we need to have mechanism to avoid future generation from seeing the same problem. We propose to make sure that when EHT sees unreconginzed field in Multi-STA BA, they will stop processing the remainig part of Multi-STA BA. | suggest the following rule "Starting from EHT STA, when see an unrecognized field in Multi-STA BA, then shall ignore the rest BA information field. For EHT AP that sends multi-STA BA to a group of STAs, for any STA1 and STA2 in the group, if the Per AID TID Info of STA 1 can not be recognized by STA2, then put the Per AID TID Info of STA 2 in front of the Per AID TID Info of STA 1 STA ignores the rest of BA information in Multi-STA BA if sees its own AID in a Per AID TID Info and sees a different AID in a later Per AID TID Info " | Rejected  Discussion: In some sense, what the commenter proposed can give more chance to do MU transmission. However the current EHT AP can’t create M-BA (Multi-STA BA) with Per AID TID field whose BA bitmap is longer than 1024 bits. What the commenter proposed may create inter-op issue. |
| 6626 | 262 | 53 | Texts in Motion 112, #SP6 use transmit buffer control shown below. "For each block ack agreement between two MLDs, there exists one transmit buffer control to submit MPDUs for transmission across links." However, the spec texts uses transmission window. In the baseline, both terms are used for describing the same thing. Suggest to clarify this for MLD. Note that the baseline has the following. "The originator contains a transmit buffer control that uses WinStartO and WinSizeO to submit MPDUs for transmission and releases transmit buffers upon receiving BlockAck frames from the recipient." "The originator may transmit QoS Data frames with a TID matching a block ack agreement(#2608) in any order provided that their sequence numbers lie within the current transmission window." Clarification needs to be made so that both texts in the baseline can be reused. | Revise the texts as: An initiating MLD shall maintain a single transmission buffer control that uses WinStartO and WinSizeO for each block ack agreement negotiated with the responding MLD to submit MPDUs for transmission across links subjected to the TID to link and releases transmit buffers upon receiving BlockAck frames from the recipient MLD. Transmission buffer control and transmission window are equivalent in the description. | Revised  TGbe editor to make changes in 11-21/1601r4 under CID 6626 |
| 6990 | 262 | 50 | Typo'TID to link' | Change 'TID to link" to TID-to-link' | Accepted |

***TGbe editor: Please change subclause name of subclase 35.3.7 as shown below:***

**35.3.7 Block ack procedures in Multi-link operation**

***TGbe editor: Please delete subclause titles of 35.3.7.1 and 35.3.7.1.1 in 35.3.7: (#4111)***

***TGbe editor: Please add the following paragraphs in 35.3.7: (#4119, 5726)***

. .

A recipient MLD may do one of the following:

* Have a separate scoreboard context control with partial state operation in each link
* Have a separate scoreboard context control with full state operation in each link
* Have one scoreboard context control with partial state operation for all links
* Have one scoreboard context control with full state operation for all links

When a STA affiliated with a recipient MLD that has a separate scoreboard context control in each link transmits a BlockAck frame on a link with scoreboard context WinStartR and WinEndR carrying the reception status of one MPDU successfully received by another STA affiliated with the recipient MLD with sequence number equal to SN, where WinEndR < SN < WinStartR+2^11, then the STA shall follow 10.25.6.3 Scoreboard context control during full-state operation to update WinStartR and WinEndR.

If the following conditions are true

* a recipient MLD has a separate scoreboard context control in each link,
* The STA affiliated with the MLD is capable of using reordering buffer information to update its scoreboard context.
* a STA affiliated with the MLD receives a frame with SN that satisfies *WinStartR*+211  *SN*  *WinStartR*
* The SN of the frame doesn’t satisfy *WinStartB*+211  *SN*  *WinStartB*

the STA shall update the scoreboard context as if the frame with SN that satisfies *WinEndR*  *SN*  *WinStartR* +211 is received.

If the following conditions are true

* a recipient MLD has a separate scoreboard context control in each link,
* The STA affiliated with the MLD is capable of using reordering buffer information to update its scoreboard context.
* a STA affiliated with the MLD receives a frame with SN that is *WinStartR*  *SN*  *WinEndR*, and before the frame reception the other STA affiliated with the MLD receives a frame with SN that is *WinStartB*-WinSize  *SN*  *WinStartB*

the STA shall flush the scoreboard context and update the scoreboard context.

NOTE x----This can happen when the originator MLD uses more than one link to transmit (A)MPDUs for a TID and the recipient MLD uses either full-state operation or partial-state operation. See 10.25.6.3 (Scoreboard context control during full-state operation) and 10.25.6.4 (Scoreboard context control during partial-state operation).

If a STA affiliated with a recipient MLD in a link is not capable of using reordering buffer information to update its scoreboard context and the recipient MLD has a separate scoreboard context control in each link, the STA shall implement the partial-state operation and should discard the temporary record in the following defined time periods:

* After sending a BA **where the BA and the acknowledged A-MPDU(s) are in one TXOP** and before processing the scoreboard context of the next  received the QoS Data frame of the TID from the initiator MLD in the link if BA is transmitted
* After the end of the current TXOP and before processing the scoreboard context of the next received the QoS Data frame of the TID from the initiator MLD in the link in a new TXOP if BA is not transmitted at the end of the current TXOP

NOTE----a STA affiliated with a recipient MLD that discards the temporary record later than the time periods mentioned in the previous paragraph could fail to update the scoreboard context per the received frame within the transmit buffer control of the initiator MLD and can’t acknowledge the received frame.

***TGbe editor: Please add the following paragraph in 35.3.7:* (#4111, 5167, 7603, 7604, 7605)**

An originator MLD shall maintain a single transmit buffer control that uses *WinStartO* and *WinSizeO* for each block ack agreement negotiated with the recipient MLD to submit MPDUs for transmission across links subjected to TID-to-Link mapping restriction (see 35.3.6 (Link management)). An originator MLD shall release transmit buffer associated with an successful received MPDU upon receiving BlockAck frame containing the reception status for that MPDU (#7602, 7413, 7372, 7371, 6489, 5688, 6626, 5988)

***TGbe editor: Please delete 35.3.7.2* (#4111, 5167, 7603, 7604, 7605)**

***TGbe editor: Please change the first paragraph in 35.3.7.1 as follows*(#4111, 5167, 7603, 7604, 7605)**

(#1751)An MLD shall follow the mechanisms defined in 11.5 (Block ack operation) and 35.x (EHT acknowledgment procedure) with additional rules as defined in this subclause for performing block ack operation.

***TGbe editor: Please add the following as a new subclause in clause35 as shown below:***

**35.x EHT acknowledgment procedure (#4111, 5167, 7603, 7604, 7605)**

**35.x.1 Overview**

The EHT acknowledgment procedure builds on the features defined for HT-immediate block ack (see 10.25.6 (HT-immediate block ack extensions)) and HE acknowledgement (see 26.4 (HE acknowledgment procedure)), with the following extensions:

—Support for BlockAck Bitmap field lengths of 512 and 1024

An EHT AP shall not transmit a Multi-STA BlockAck frame that contains a BlockAck Bitmap field with length equal to 512 or 1024 bits as a response to an HE TB PPDU generated by at least one HE STA.

**35.x.2 Block ack bitmap lengths**

Both the Compressed BlockAck frame and Multi-STA BlockAck frame allow different Block Ack Bitmap subfield lengths. The length of the Block Ack Bitmap subfield is indicated in the Fragment Number subfield of the Block Ack Starting Sequence Control field as defined in 9.3.1.8 (BlockAck frame format). The allowed Block Ack Bitmap lengths for each of the negotiated buffer sizes are defined in Table 35-1 (Negotiated buffer size and Block Ack Bitmap subfield length).

## Negotiated buffer size and Block Ack Bitmap subfield length

|  |  |  |
| --- | --- | --- |
| **Negotiated buffer size** | **Block Ack Bitmap subfield length (bits) in a Compressed BlockAck frame** | **Block Ack Bitmap subfield length (bits) in a Multi-STA BlockAck frame** |
| 1–64 | 64 | 32 or 64 |
| 65–128 | 64 or 256 | 32, 64, or 128 |
| 129–256 | 64 or 256 | 32, 64, 128, or 256 |
| 257–512 | 64, 256, or 512 | 32, 64, 128, 256, or 512 |
| 513–1024 | 64, 256, 512, or 1024 | 32, 64, 128, 256, 512, or 1024 |
| NOTE—A 32-bit Block Ack Bitmap subfield length is not allowed unless the originator has set the 32-bit BA Bitmap Support field in the HE MAC Capabilities Information field in the HE Capabilities element to 1. | | |