IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC36 Resolution for CIDs related to MLO BA Procedure (Part 2) | | | | |
| Date: September 23, 2021 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Abhishek Patil | Qualcomm Inc |  |  | appatil@qti.qualcomm.com |
| Alfred Asterjadhi |  |  |  |
| George Cherian |  |  |  |
| Jouni Malinen |  |  |  |
| Duncan Ho |  |  |  |
| Gaurang Naik |  |  |  |
| Yanjun Sun |  |  |  |
| Abdel Karim |  |  |  |
| Rojan Chitrakar | Panasonic |  |  |  |
| Tomo Adachi | Toshiba |  |  |  |
| Liwen Chu | NXP |  |  |  |
| Michael Montemurro | Huawei |  |  |  |
| Insun Jang | LGE |  |  |  |

Abstract

This submission proposes resolutions for following 11 CID received for TGbe CC36:

7435, 5159, 4062, 7596, 6625, 6289, 7601, 7894, 6675, 6992, 6993

***TGbe editor: The baseline for this document is REVme D1.0 and 11be D1.3.***

**Revisions:**

* Rev 0: Initial version of the document.
* Rev 1: Revised based on feedback from several members (added as co-authors).

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** | **Commenter** | **Clause** | **Pg/Ln** | **Comment** | **Proposed Change** | **Resolution** |
| 7435 | Thomas Derham | 35.1 | 0.00 | Unprotected BAR allows DoS attack by advancing the sliding window of expected SNs. DoS attacks are becoming higher profile, and 11be should protect against them. PBAC mechanism is already defined in baseline | Mandate support and negotiation of PBAC between 11be devices | **Revised**  Agree with the comment. BlockAckReq frame is a Control frame and Control frames are not protected. There have been reports of a DoS attack that exploits the vulnerabilities of BAR frame. EHT STAs must be protected against such attacks.  The resolution proposes to mandate protected BA between two EHT STAs, extends the rules in 10.25.7 for protected BA setup between two MLDs and addresses an issue in which legacy STAs pretend to support PBAC.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 7435** |
| 5159 | GEORGE CHERIAN | 35.3.7.1 | 261.37 | Procedure for response to BAR and reshuffling of sequence number is not clear for MLD operation. Please define | As in the comment | **Revised**  The resolution for CID 7435 proposes that the BA setup between two MLDs will be a protected BA setup. This means that a BAR would only cause the recipient to send a BA containing the reception status. The BAR will not cause an update to the scoreboard context. An originator MLD would send a robust ADDBA Request frame to update the *WinStartR* and *WinStartB* at the recipient MLD.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 7435** |
| 4062 | Abhishek Patil | 35.3.7.1.1 | 261.41 | How does protect BA mechanism work in MLO? How does BAR (protected and unprotected) work in MLO - esp. since a successful scoreboard update needs to account for status from all links, which may not be immediate? Please clarify | As in comment | **Revised**  The resolution for CID 7435 proposes that the BA setup between two MLDs will be a protected BA setup. This means that a BAR would only cause the recipient to send a BA containing the reception status. The BAR will not cause an update to the scoreboard context. An originator MLD would send a robust ADDBA Request frame to update the *WinStartR* and *WinStartB* at the recipient MLD.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 7435** |
|  |  |  |  |  |  |  |

**Discussion:**

Control frames are not protected and therefore vulnerable to an attack by a rogue device. An attacker can inject a Control frame which can have a damaging effect at the receiver.

BlockAckReq frame is an unprotected (Control) frame, and an attacker can use it to mess up with the reorder buffer and scoreboard context at a recipient by setting an arbitrary value in the Block Ack Starting Sequence Control subfield. This can result in denial-of-service attack where the attacker can block delivery of Data frames for a specific TID.

The IEEE 802.11 (baseline) standard defines a mechanism to setup protected block ack session between an originator and a recipient. In a protected block ack setup, a BlockAckReq frame can only be used to solicit reception status (i.e., a BlockAck frame). It is ignored for the purpose of updating the *WinStartB*. In a protected block ack setup, the originator transmits a robust ADDBA Request frame to update the *WinStartB* and *WinStartR*. See 10.25.7 (Protected block ack agreement) for details of the operation

Unfortunately, there are no known implementations of protected BA and in recent months, there have been reports of an attack that exploited the vulnerabilities of BlockAckReq frame. Furthermore, based on offline feedback, it has been discovered that many legacy devices (deployed in the field) incorrectly set the PBAC bit in RSNE to 1 even when they do not support protected BA operation. In addition, based on offline feedback from many members, it was also discovered that the procedures described in baseline are lacking details.

To address the above issues, this contribution proposes the following (as a resolution to CID 7435):

1. Recommends that all EHT STAs in R1 timeframe support protected BA procedure.
   1. Mandatory for R2 and beyond
   2. Ideally, TGbe must mandate protected BA for all EHT STAs. However, the requirement is relaxed to a recommendation for R1 timeframe to accommodate limitations for some vendors.
2. Defined a new bit in RSN Extension element to signal support for PBAC while deprecating (set to Reserve) the existing PBAC bit in RSNE
3. Added missing details related to protected BA procedure in baseline (based on offline feedback from various members).
   1. In a protected BA agreement, the order of decryption and update to reorder buffer must be maintained (as shown in Figure 5-1). This is to prevent an attack where a fake Data frame causes the *WinStartB* to get updated while the decryption of the frame fails.
   2. BAR is ignored for the purpose of updating *WinStartR*
   3. *WinStartB* is not updated when an error/attack condition is determined
   4. Scoreboard context is purged if MPDU decryption or integrity check fails
   5. Make the normative text in 10.25.7 consistent with 9.6.4.1. The receiving STA (MLD) responds with a robust ADDBA Response as a confirmation (complete the handshake – same as setup). Originator updates *WinStartO* upon receiving robust ADDBA Response to its solicited ADDBA Request frame.

* **Protected block ack agreement**[7435]

***TGbe editor: Please update the following paragraph this subclause as shown below:***

A STA indicates support for protected block ack by setting the MFPC and MFPR subfields in RSN Capabilities field to 1 (see 9.4.2.24.4 (RSN Capabilities)) and the PBAC field in RSN Extension element to 1 (see 9.4.2.241 (RSN Extension element (RSNXE))). Such a STA is a PBAC STA; otherwise, the STA is a non-PBAC STA. An EHT STA with dot11EHTBaseLineFeaturesImplementedOnly equal to true should operate as a PBAC STA; Otherwise, an EHT STA shall operate as PBAC. A block ack agreement that is successfully negotiated between two PBAC STAs is a protected block ack agreement. A block ack agreement that is successfully negotiated between two STAs when either or both of the STAs is not a PBAC STA is a block ack agreement that is not a protected block ack agreement.

***TGbe editor: Please update the following paragraph this subclause as shown below:***

A STA that has successfully negotiated a protected block ack agreement shall obey the following rules as a block ack originator in addition to rules specified in 10.25.6.7 (Originator’s behavior) and 10.25.6.8 (Maintaining block ack state at the originator):

* To change the value of *WinStartB* at the receiver, the STA shall use a robust ADDBA Request frame
* Upon reception of a valid robust ADDBA Response frame, in response to its robust ADDBA Request frame, for an established protected block ack agreement whose TID and recipient address are the same as those of the block ack agreement, the STA shall validate the value carried in the Dialog Token field of the ADDBA Response frame before updating its *WinStartO* with the value of the Block Ack Starting Sequence Control field of the ADDBA Request frame it transmitted. The STA shall ignore the values carried in other fields of the ADDBA Response frame.

***TGbe editor: Please update the following paragraph this subclause as shown below:***

A STA that has successfully negotiated a protected block ack agreement shall obey the following rules as a block ack recipient in addition to rules specified in 10.25.6.3 (Scoreboard context control during full-state operation) to 10.25.6.6 (Receive reordering buffer control operation):

* The recipient STA shall respond to a BlockAckReq frame from a PBAC enabled originator with an immediate BlockAck frame. The Block Ack Starting Sequence Control subfield value shall be ignored for the purposes of updating the value of *WinStartB* and *WinStartR*. If the Block Ack Starting Sequence Control subfield value is greater than *WinEndB* or less than *WinStartB*, dot11PBACErrors shall be incremented by 1. If, for a block ack agreement with segmentation and reassembly, the MPDU Starting Sequence subfield value is greater than WinEndB or less than WinStartB, dot11PBACErrors shall be incremented by 1. The recipient STA shall not update the value of *WinStartB* when dot11PBACErrors is incremented.
* The recipient STA shall respond to an MPDU, with ACK policy other than No Ack, from a PBAC enabled originator with a BlockAck frame. If an MPDU does not pass decryption or integrity check, the recipient shall not update the value of *WinStartR*, shall purge the scoreboard context, and shall increment dot11PBACErrors by 1.
* Upon receipt of a valid robust ADDBA Request frame for an established protected block ack agreement whose TID and originator address are the same as those of the block ack agreement, the recipient STA shall update its *WinStartR* and *WinStartB* values based on the starting sequence number in the robust ADDBA Request frame according to the procedures outlined for reception of BlockAckReq frames in 10.25.6.3 (Scoreboard context control during full-state operation), 10.25.6.4 (Scoreboard context control during partial-state operation), 10.25.6.6.1 (General), and 10.25.6.6.3 (Operation for each received BlockAckReq), while treating the starting sequence number as though it were the *SSN* of a received BlockAckReq frame or, in case of a block ack agreement with segmentation and reassembly, treating the MPDU starting sequence number as though it were the MPDU SSN of a received BlockAckReq frame and shall transmit a robust ADDBA Response frame. The STA shall ignore the values carried in the other fields of the ADDBA Request frame.
* RSN capabilities[7435]

***TGbe editor: Please update Figure 9-350 to update bit 12 from “PBAC” to “Reserved”***

***TGbe editor: Please update the following bullet in this subclause shown below:***

* Bit 12: Reserved
* RSN Extension element (RSNXE)[7435]

***TGbe editor: Please add a new row to Table 9-363 as shown below:***

|  |  |  |
| --- | --- | --- |
| * Extended RSN Capabilities field | | |
| Bit | Information | Notes |
| <ANA> | PBAC | A STA sets the PBAC field to 1 to indicate it can establish a protected block ack agreement. Otherwise, this field is set to 0. |
| 8, 9, 10, <ANA>– (8´*n* – 1) | Reserved |  |

**35.3.7 Multi-link block ack**

**35.3.7.1 Multi-link BlockAck procedure**

**35.3.7.1.1 General**

***TGbe editor: Please update the 2nd paragraph this subclause as shown below:***

[7435]For each TID, there shall not be more than one block ack agreement established between two MLDs and the agreement shall apply to all the links to which the TID is mapped to (i.e., there are no independent block ack agreements for each TID on a per-link basis). A protected block ack agreement that is negotiated between two MLDs shall follow the procedure defined in 10.25.7 (Protected block ack agreement) along with additional rules described in this subclause.

***TGbe editor: Please add the following paragraph at the end of this subclause as shown below:***

[7435]For an existing protected block ack agreement between two MLDs:

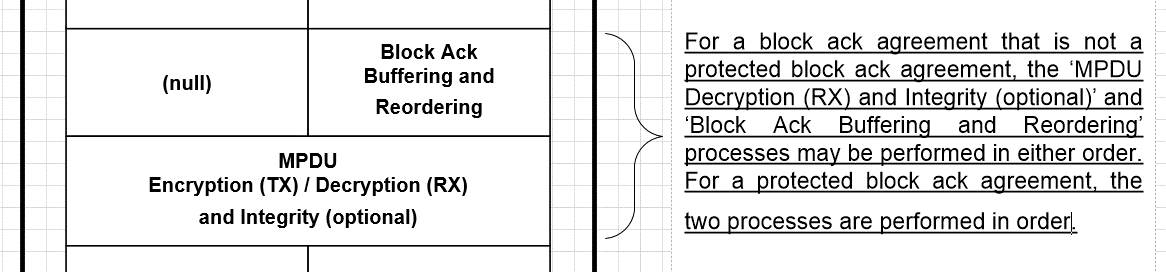
* an originator MLD shall transmit a robust ADDBA Request frame on a link on which the TID matching the block ack agreement is mapped to in order to update the *WinStartR* and *WinStartB* at the recipient MLD.
* a recipient MLD shall, upon receiving a robust ADDBA Request frame from the originator MLD, update the *WinStartR* and *WinStartB* maintained at the MLD-level based on the value carried in the Starting Sequence Number field in the received ADDBA Request frame and shall transmit a robust ADDBA Response frame on a link on which the TID matching the block ack agreement is mapped to.
* an originator MLD shall, upon receiving a valid robust ADDBA Response frame from the recipient MLD in response to its ADDBA Request frame, update the *WinStartO* with the value of the Block Ack Starting Sequence Control field of the ADDBA Request frame.

NOTE – A STA affiliated with a recipient MLD, upon receiving a BlockAckReq frame, responds with a BlockAck frame and ignores the value in the Block Ack Starting Sequence Control subfield of the BlockAckReq frame for the purpose of updating the *WinStartB* and *WinStartR* (see 10.25.7 (Protected block ack agreement)).

**5.1.5 MAC data service architecture**

**5.1.5.1 General** [7435]

***TGbe editor: Please replace the side note in Figure 5-1 as shown below:***



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CID** | **Commenter** | **Clause** | **Pg/Ln** | **Comment** | **Proposed Change** | **Resolution** |
| 7596 | Tomoko Adachi | 35.3.7.1.1 | 262.06 | The current spec (10.25.6.5) allows to set any value for the status between the SSN of the BA frame and adjusted WinStart\_R, if the adjusted WinStart\_R is greater than the SSN of the BA frame. The scoreboard context control is supposed to be at least in the link level MAC (lower MAC), but when responding like this, will the MLD level MAC (higher MAC) also hold the combined scoreboard state among the enabled links? Then, which WinStart\_R is applied when responding at a link? The combined one, I think. And the above rule in 10.25.6.5 should apply. Such clarification is needed. | As in comment. | **Revised**  Agree in principle.  **TGbe editor, no further changes are needed to address this comment** |
| 6625 | Po-Kai Huang | 35.3.7.1.1 | 261.38 | It should be clarified that independent scoreboard context control (partial state) can be used in any link. Dynamically coordinate the Block ack received status across links is difficult, and certainly can not be mandated. | add "an recipient MLD may have independent scoreboard context control during partial-state operation for each <peer MLD, TID> tuple under a block ack agreement in each setup link." | **Revised**  Agree in principle. A paragraph along with a figure is added to provide a description of the expected behavior when each STA affiliated with the recipient MLD maintains an independent scoreboard context.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 6625** |
| 7601 | Tomoko Adachi | 35.3.7.1.1 | 262.01 | The behavior descibed here means that a partial-state operation is at least required on per-link basis and full-state operation is not necessary. Such description by a note can be helpful. | As in comment. | **Revised**  Agree in principle. A paragraph along with a figure is added to provide a description of the expected behavior when each STA affiliated with the recipient MLD maintains an independent scoreboard context.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 6625** |
| 7894 | Yongho Seok | 35.3.7.1 | 262.24 | "It shall maintain its own state independent of the scoreboard context control to perform this reordering as specified in 10.25.6.6 (Receive reordering buffer control operation)." The partial-state operation should be clarified. | As in the comment. | **Revised**  Agree in principle. A paragraph along with a figure is added to provide a description of the expected behavior when each STA affiliated with the recipient MLD maintains an independent scoreboard context.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 6625** |
| 6675 | Rajat Pushkarna | 35.3.7.1.1 | 262.23 | "It shall maintain its own state independent of the scoreboard context control to perform this reordering..". Scoreboard context available at MLD or STA level is implementation specific. Scoreboard context control maintained at MLD cannot determine the link for which scoreboarding is done with <peer MLD, TID> tuple | Based on discussion in ARC SC scoreboard context control should be implementation specific. I personally prefer it to be maintained at STA level per link for tracking data frames delivered per link. If it has to maintained at MLD level the tuple should be modified to <peer MLD, TID, link ID> | **Revised**  Agree in principle. A paragraph along with a figure is added to provide a description of the expected behavior when each STA affiliated with the recipient MLD maintains an independent scoreboard context.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 6625** |
| 6992 | Sharan Naribole | 35.3.7.1.1 | 262.25 | Per-link Scoreboard context control operation is vague. The only description found is "Each received MPDU shall be analyzed by the scoreboard context control as well as by the receive reordering buffer control." It is not clear how the WinStartR is updated at a specific link's scoreboard context control | Specific operations details should be added how the WinStartR is updated at a specific link's scoreboard context control | **Revised**  Agree in principle. A paragraph along with a figure is added to provide a description of the expected behavior when each STA affiliated with the recipient MLD maintains an independent scoreboard context.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 6625** |
| 6993 | Sharan Naribole | 35.3.7.1.1 | 262.25 | Per-link Scoreboard context control operation is vague. The only description found is "Each received MPDU shall be analyzed by the scoreboard context control as well as by the receive reordering buffer control." Considering the different PHY capabilities at each link, it is not clear how the WinSizeR is assigned to each link's scoreboard context control. | Specific operations details should be added how the WinSizeR is assigned for a specific link's scoreboard context control | **Revised**  Agree in principle. A paragraph along with a figure is added to provide a description of the expected behavior when each STA affiliated with the recipient MLD maintains an independent scoreboard context.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 6625** |
| 6289 | Ming Gan | 35.3.7.1.1 | 261.51 | Based on the architecture documents 21/577r2, each affiliated STA has a scoreboard, the buffer size of each link should be negotiated by ADDBA request/response exchange. | as in the comment | **Revised**  The MLDs do not need to negotiate separate block ack agreements when each STA affiliated with the recipient MLD maintains an independent scoreboard context. A paragraph along with a figure is added to provide a description of the expected behavior.  **TGbe editor, please make changes as shown in doc 11-21/1582r1 tagged as 6625** |

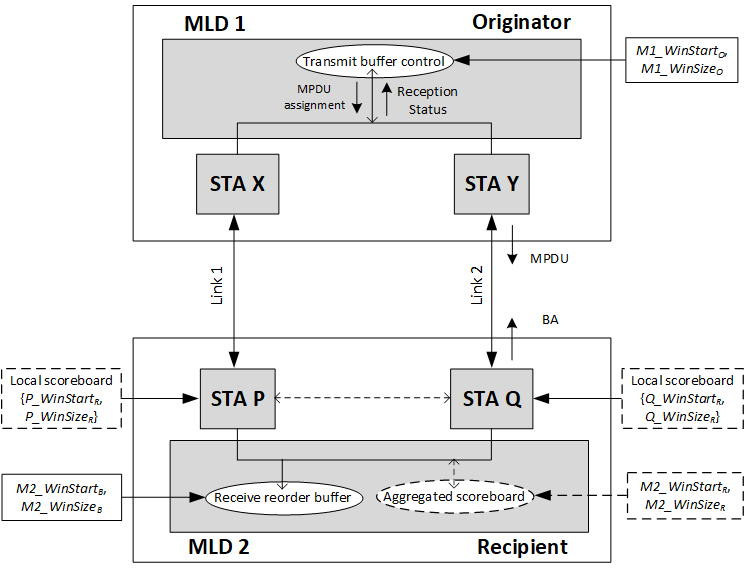
**35.3.7 Multi-link block ack**

**35.3.7.1 Multi-link BlockAck procedure**

**35.3.7.1.1 General**

***TGbe editor: Please add the following paragraph and figure at the end of this subclause as shown below:***

[6625]A recipient MLD may have independent scoreboard context control in each setup link during partial-state or full state operation for each <peer MLD, TID> tuple under a block ack agreement. Figure 35-xx (BA setup in MLO) illustrates a high-level diagram of such a configuration. In the figure, an originator MLD (M1) has negotiated a block ack agreement with a recipient MLD (M2) for a certain TID. M1 maintains a single common transmit buffer control (depicted as *M1\_WinStartO*, *M1\_WinSizeO*) for this block ack agreement. M2 can either have each affiliated STA (P and Q) can maintain a local scoreboard context (depicted as {*P\_WinStartR*, *P\_WinSizeR*} and {*Q\_WinStartR*, *Q\_WinSizeR*} respectively) or have a single common scoreboard context at the MLD level which reflects the combined status from each affiliated STA (depicted as *M2\_WinStartR*, *M2\_WinSizeR*) or a combination. Since either option is likely to be implemented, they are shown as dotted lines in the figure. M2 also maintains a single common receive reorder buffer (depicted as *M2\_WinStartB*, *M2\_WinSizeB*) for this block ack agreement*.*



**Figure 35-xx BA setup in MLO**[6625]