IEEE P802.11
Wireless LANs

|  |
| --- |
| Resolution to CID 1023 |
| Date: 2024-04-29 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Michael Montemurro | Huawei | Toronto, Canada |  | montemurro.michael@gmail.com  |
| Stephen McCann | Huawei | Southampton, UK |  | stephen.mccann@ieee.org  |
| Arik Klein | Huawei TRC |  |  | arik.klein@huawei.com  |
| Oren Hencinski | Huawei Datacom PL |  |  | oren.hencinski@huawei.com  |

Background

The contribution proposes a resolution to CID 1023.

R0: Initial version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Page** | **Comment** | **Proposed Change** |
| 1023 | 37.8.2.5 | 38.36 | There are mobility scenarios where it would be advantageous to address some of the shortcomings with MLO as described in doc 11-24/778, and prevent a non-collocated MLO deployment. | To address highly distributed, high mobility environments, add a new optional encapsulation mechanism that allows MSDU's to be cryptographically encrypted prior to assigning an SN. |

### Discussion:

There are several features in P802.11bn which look to provide higher reliability and a better user experience for mobility scenarios. These include seamless roaming and relay operation.

<https://mentor.ieee.org/802.11/dcn/24/11-24-0778-00-00bn-nc-mlo-operation-issues.pptx> described some limitations with MLO as defined in IEEE 802.11be-2024 with respect to high reliability features. These include limitations with the requirements for Block ACK operation in the interaction between the upper MAC and lower MAC, limitations on achieving high throughput in mobility scenarios, and the ability to duplicate traffic across multiple links (during mobility scenarios) to facilitate better reliability.

<https://mentor.ieee.org/802.11/dcn/25/11-25-0201-00-00bn-frame-processing-enhancements-for-tgbn.pptx> provides the description of a proposal to address the above features. It proposes an optional feature to allow a UHR non-AP MLD and a UHR AP MLD to negotiate enhanced frame processing.

This document provides the changes to the UHR draft to incorporate that proposal.

### Proposed Resolution:

REVISED. Add an optional enhanced frame processing mechanism to the UHR draft as described in <this>.

# 3.4 Abbreviations and acronyms

## *Insert the following acronym definitions (maintaining alphabetical order):*

EFP enhanced frame processing

* RSNXE(#1776)

(#1776)The RSN extension element (RSNXE) contains additional information required to establish an RSNA. The format of the RSNXE is defined in Figure 9-888 (RSNXE format).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Element ID | Length | Extended RSN Capabilities |
| Octets: | 1 | 1 | *n* |
| * Figure 9-888 - RSNXE format
 |

The Element ID and Length fields are defined in 9.4.2.1 (General).

The Extended RSN Capabilities field, except its first 4 bits, is a set of 1-bit fields(#291) indicating the extended RSN capabilities being advertised by the STA transmitting the element. The length of the Extended RSN Capabilities field is a variable *n*, in octets, as indicated by the first 4 bits in the field. The Extended RSN Capabilities field is shown in Table 9-373 (Extended RSN Capabilities field).

***Update the bottom of the table as follows:***

|  |
| --- |
| Table 9-373 - Extended RSN Capabilities field |
| Bit | Information | Notes |
| 15(11az) | URNM-MFPR | A STA sets the URNM-MFPR field to 1 if dot11RSTARequiresPMFActivated is set to 2. Otherwise, it sets the field to 0. See 11.21.6.3.1 (General). |
| 21(M166) | SSID Protection | A STA sets this field to 1 to indicate support for protected exchange of the SSID during the 4-way handshake. Otherwise, it sets the field to 0. |
| 22(#7028) | QMF ACI Subfield Unmask Support | A STA sets the QMF ACI Subfield Unmask Support to 1 if the STA supports to unmask ACI subfield during AAD construction. Otherwise, this subfield is set to 0. |
| <ANA> | Enhanced Frame Processing  | A STA sets this field to 1 to indicate support for the cryptographic encapsulation of MSDUs. See 12.x.x.x. Otherwise, it sets the field to 0. |
| (M34)16–19, <ANA>–(8*n* – 1) | Reserved |  |

If a STA does not support (#470)any of the capabilities defined in the RSNXE, then the RSNXE is not present.

* RSNA confidentiality and integrity protocols

***Insert the following clause after 12.5.4:***

**12.5.5 Enhanced frame processing**

**12.5.5.1 General**

Subclause 12.5.5 specifies an Enhanced Frame Processing (EFP) feature that is based on CCMP and GCMP. EFP uses CCMP or GCMP to encapsulate MSDUs, A-MSDUs, or MMPDUs prior to the assignment of an SN.

A STA that supports EFP shall advertise support for EFP by setting the Enhanced Processing Field to 1 in the RSNXE in Beacons and Probe Response frames.

A STA shall negotiate EFP by setting the Enhanced Frame Processing field to 1 in the RSNXE in (Re-)Association Request frames. When CCMP or GCMP is negotiated as the RSN pairwise cipher and the Enhanced Frame Processing field is equal to 1, the STA shall use EFP with the negotiated CCMP or GCMP cipher suite.

The following subclauses describe the requirements for CCMP and GCMP when using EFP.

**12.5.5.2 Cryptographic encapsulation using EFP**

When EFP is negotiated, a STA uses the cryptographic encapsulation process described in 12.5.2 for CCMP if CCMP was the negotiated cipher suite, or 12.5.4 for GCMP if GCMP was the negotiated cipher suite. When EFP is negotiated, cryptographic encapsulation is performed on an MSDU rather than an MPDU.

The MPDU header format described for CCMP and GCMP is unchanged.

**12.5.5.3 Cryptographic Encapsulation**

The cryptographic encapsulation process for EFP shall follow 12.5.2.3 for CCMP or 12.5.4.3 for GCMP, operating on an MSDU, A-MSDU, or MMPDU instead of an MPDU.

**12.5.5.4 Cryptographic Decapsulation**

The cryptographic decapsulation process for EFP shall follow 12.5.2.4 for CCMP or 12.5.4.4 for GCMP, operating on an MSDU, A-MSDU, or MMPDU instead of an MPDU.

***Modify the following clauses as shown:***

1. **Extremely high throughput (EHT) MAC specification**

**35.3.8 Block ack procedures in MLO**

An MLD shall follow the mechanisms defined in 11.5 (Block ack operation) and 35.4 (EHT acknowledgment procedure) with additional rules defined in this subclause for block ack operation.

When EFP is negotiated, an MLD shall follow the mechnaisms defined in 11.5 (Block ack operation), although the block ack agreement between two MLDs is negotiated on a per-link basis. To setup a block ack agreement between two MLDs on a negotiated link, an originator MLD shall send, through any affiliated STA operating on the negotiated link, an ADDBA Request frame to the recipient MLD, subject to the power state of the affiliated non-AP STA operating on that link. The ADDBA Request frame shall indicate the TID for which the block ack agreement is being set up. Upon receiving an ADDBA Request frame, the recipient MLD shall initiate a transmission of an ADDBA Response frame through the affiliated STA operating on the negotiated link, with subject to the power state of the non-AP STA operating on that link. The recipient MLD has the option of accepting or rejecting the request. If the recipient MLD accepts the request, then a block ack agreement is established between the originator MLD and the recipient MLD for the specified link and for the TID specified in the ADDBA frames as defined in 10.25.2 (Setup and modification of the block ack parameters).

When EFP is not negotiated, 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 that the TID is mapped to (i.e., there are no independent block ack agreements for each TID on a per-link basis).

In this subclause, the MLD with data to send, using the block ack mechanism, is referred to as the originator MLD, and the MLD that is the intended recipient of that data is referred to as the recipient MLD.

To setup a block ack agreement between two MLDs when EFP is not negotiated, an originator MLD shall send, through any affiliated STA operating on an enabled link, an ADDBA Request frame to the recipient MLD, subject to the power state of the non-AP STA operating on that link. The ADDBA Request frame shall indicate the TID for which the block ack agreement is being set up. Upon receiving an ADDBA Request frame, the recipient MLD shall respond through any affiliated STA operating on an enabled link, with an ADDBA Response frame subject to the power state of the non-AP STA operating on that link. The recipient MLD has the option of accepting or rejecting the request. If the recipient MLD accepts the request, then a block ack agreement is established between the originator MLD and the recipient MLD for the TID specified in the ADDBA frames as defined in 10.25.2 (Setup and modification of the block ack parameters).

NOTE 1— When EFP is not negotiated, an originator MLD can attempt a retransmission of an ADDBA Request frame on any enabled link. A recipient MLD can attempt a retransmission of an ADDBA Response frame on any enabled link.

If an EFP is not negotiated, when two MLDs have established a block ack agreement between them for a certain TID, then QoS Data frames belonging to that TID may be exchanged between the two MLDs on any link that the TID is mapped to following the procedures described in 35.3.7.2 (TID-To-Link Mapping (TTLM)) and 35.3.12 (ML power management).

When an EFP is negotiated, an AP MLD and non-AP MLD that have established a block ack agreement between them for a certain TID and a certain setup link, QoS Data frames belonging to that TID may be exchanged between the two MLDs that link provided that the TID is mapped to that link following the procedures described in 35.3.7.2 (TID-To-Link Mapping (TTLM)) and 35.3.12 (ML power management)

An originator MLD shall maintain a single common transmit buffer control that uses WinStartO and WinSizeO (see 10.25.6.7 (Originator’s behavior)) for each block ack agreement negotiated with the recipient MLD to submit MPDUs for transmission across links subject to a TTLM restriction (see 35.3.7 (Link management)). An originator MLD shall release the transmit buffer associated with a successfully received MPDU upon reception of the BlockAck frame that indicates the reception of that MPDU.

In a block ack agreement between two MLDs, the buffer size is indicated based on the Buffer Size subfield (of the Block Ack Parameter Set field) together with the Extended Buffer Size field (when an ADDBA Extension element is included).

NOTE 2—The ADDBA Extension element is optionally present in an ADDBA Request or ADDBA Response frame (see 9.6.4 (Block Ack Action frame details)). When a block ack agreement is negotiated between two MLDs, if the ADDBA Extension element is present, then the total buffer size is computed as described in 9.4.2.138 (ADDBA Extension element).

During the block ack agreement establishment, the Block Ack Timeout field and buffer size indicated in the ADDBA Request frame are advisory.

When a block ack agreement is established between two MLDs, the originator may change the size of its transmission window (WinSizeO) provided the transmission window meets the following conditions:

— Is not greater than the buffer size indicated in the ADDBA Response frame.

— Is not greater than 1024.

If the buffer size indicated in the ADDBA Response frame is smaller than the buffer size indicated in the ADDBA Request frame, the originator shall change the size of its transmission window (WinSizeO) such that:

— The transmission window is not greater than the buffer size indicated in the ADDBA Response frame.

— The transmission window is not greater than 1024 if the sender and the receiver of the ADDBA Response frame are MLDs.

A STA affiliated with a recipient MLD shall provide, to the STA affiliated with the originator MLD that is operating on the same link, the reception status for any MPDU, with an ack policy other than No Ack, that is received on the link on which the STA affiliated with the recipient MLD is operating. When a TID is mapped to more than one link, a STA affiliated with a recipient MLD may provide a reception status (if available) to the STA affiliated with the originator MLD that is operating on the same link, indicating the successful reception of any MPDU that belongs to that TID and has an ack policy other than No Ack and that is received by a STA affiliated with the recipient MLD that is operating on a different link.

An originator MLD shall update the reception status of an MPDU in its transmit buffer corresponding to a block ack agreement, if the received status indicates successful reception.

An originator MLD shall not update the reception status of an MPDU in its transmit buffer corresponding to a block ack agreement that has already been acknowledged as successful.

A recipient MLD shall maintain a single common receive reordering buffer for each <peer MLD, TID> tuple under a block ack agreement, independent of the number of links that are setup. The receive reordering buffer shall be responsible for reordering MSDUs or A-MSDUs so that MSDUs or A-MSDUs are eventually passed up to the next MAC process in the order of received sequence number. It shall also be responsible for identifying and discarding duplicate frames (i.e., frames that have the same sequence number as a currently buffered frame) that are part of this block ack agreement. 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). Each received MPDU shall be analyzed by the scoreboard context control as well as by the receive reordering buffer control. The bitmap corresponding to each scoreboard context control shall have the same size, WinSizeR, as the smaller of the bitmap length and the buffer size indicated in the ADDBA Response frame.

When EFP is not negotiated, each <peer MLD, TID> tuple under a block ack agreement, a recipient MLD shall have one of the following modes of operation:

— maintain an independent scoreboard context control and partial state operation at each STA affiliated with the MLD operating on the link where the TID is mapped to, or

— have a common (single) scoreboard context control maintained by the MLD with partial state operation on each setup link where the TID is mapped to, or

— have a common (single) scoreboard context control maintained by the MLD with full state operation on each setup link where the TID is mapped to.

When EFP is negotiated, each <peer MLD, TID, LinkID> tuple under a block ack agreement, a recipient MLD shall maintain an independent scoreboard context control and partial state operation at each STA affiliated with the MLD operating on the link where the TID is mapped.

NOTE 3—A peer MLD is identified based on its MLD MAC address.

If a recipient MLD has independent scoreboard context control at an affiliated STA (i.e., the STA is not able to be in sync with the information at the common reordering buffer or the information at another STA affiliated with the same MLD to update its local scoreboard context), then the affiliated STA shall implement partial state operation and should discard the temporary record after the end of the current TXOP.

NOTE 4—If a recipient MLD has independent scoreboard context control at an affiliated STA (STA 1), then STA 1’s WinStartR might not be within 211 of the WinStartR at another affiliated STA (STA 2) of the same MLD. As a result, STA 1 can fail to accurately update the scoreboard context and hence, might provide an incorrect reception status for an

MPDU received in a subsequent TXOP. Therefore, it is recommended that STA 1 discards its temporary record in a timely manner. If the affiliated STA can be in sync with the latest information at another STA affiliated with the same MLD, then it does not have to discard the temporary record at the end of the current TXOP.

NOTE 5—A STA affiliated with the originator MLD is expected to solicit an immediate BA within the TXOP by following 10.25.6.9 (Originator’s support of recipient’s partial state) to reduce the probability that MPDUs are unnecessarily retransmitted.

NOTE 6—An originator MLD takes into account the SN of outstanding MPDU(s) on all enabled links when selecting the value to be carried in the SSN field of a BAR frame.

If EFP is not negotiated, when two MLDs have successfully negotiated a protected block ack agreement, they shall follow the procedure described in 10.25.7 (Protected block ack agreement). In a protected block ack agreement between two MLDs, the originator MLD shall transmit a PBAC WinStart Update frame, via its affiliated STA that is operating on an enabled link to which the TID belonging to the block ack agreement is mapped, to advance the WinStartR and WinStartB at the recipient MLD.

If EFP is negotiated, when two MLDs have successfully negotiated a protected block ack agreement per link, they shall follow the procedure described in 10.25.7 (Protected block ack agreement). In a protected block ack agreement between two MLDs for a given link, the originator MLD shall transmit a PBAC WinStart Update frame, via its affiliated STA that is operating on that enabled link to which the TID and link belonging to the block ack agreement is mapped, to advance the WinStartR and WinStartB at the recipient MLD

A block ack agreement between two MLDs may be torn down if there are no BlockAck, BlockAckReq or MPDUs received from the peer under the block ack agreement on any setup link(s) on which the TID for the block ack agreement is mapped for the direction matching the block ack agreement, within a duration of block ack timeout value (see 11.5.4 (Error recovery upon a peer failure)).

**35.3.9 Fragmentation in MLO**

A STA affiliated with an MLD shall not use the nondynamic fragmentation procedure described in

10.4 (MSDU, A-MSDU, and MMPDU fragmentation).

If EFP is negotiated, fragmentation procedures as described in 10.4 (MSDU, A-MSDU, and MMPDU fragmentation) are performed by an affiliated STA on a negotiated link.