IEEE P802.11
Wireless LANs

|  |
| --- |
| EDMG Segmentation and reassembly  |
| Date: 2017-09-9 |
| Author(s): |
| Name | Affiliation | Address | Phone | Email |
| Oren Kedem  | Intel  |  |  | oren.kedem@intel.com |
| Carlos Cordeiro  | Intel |  |  | carlos.cordeiro@intel.com |
| Nir Paz | Intel |  |  | nir.paz@intel.com |
| Michael Glik | Intel |  |  | michael.glik@intel.com |
| Solomon Trainin | Qualcomm |  |  | strainin@qti.qualcomm.com |
| Joe Andonieh  | Peraso |  |  | joe@perasotech.com |

Abstract

This document proposes new draft in the name of EDMG Segmentation and reassembly support for EDMG STAs.

**3. Definitions, acronyms, and abbreviations
3.1 Definitions**

*Add second as follows*

**Segmentation and Reassembly:** The process of partitioning a large medium access control (MAC) service data unit (MSDU) into a sequence of maximum size MAC protocol data units (MPDUs), each carry MSDU segment. The process of recombining a set of segmented MPDUs into an MSDU is known as reassembly.

**5.1.5 MAC data service architecture
5.1.5.1 General**

*Change second and third paragraphs as follows*

During transmission, an MSDU goes through some or all of the following processes: MSDU rate limiting,
aggregate MSDU (A-MSDU) aggregation, frame delivery deferral during power save mode, segmentation, sequence number assignment, integrity protection, fragmentation, encryption, frame formatting, and aggregate MAC

During reception, a received Data frame goes through processes of possible A-MPDU de-aggregation,
MPDU header and cyclic redundancy code (CRC) validation, duplicate removal, decryption, possible
reordering if the block ack mechanism is used, replay detection, defragmentation, integrity checking, reassembly
possible A-MSDU de-aggregation, and possible MSDU rate limiting. Then, one or more MSDUs are
delivered to the MAC SAP or to the DS via the DSAF.

*Replace* **Figure 5-1** and **Figure 5-2** *as follows*



**Figure 5-1—MAC data plane architecture**



**Figure 5-2—MAC data plane architecture (transparent FST)**

**9.2.4.4 Sequence Control field**

**9.2.4.4.1 Sequence Control field structure**

*Add the following after Figure 9-4 as follows*

Below figure depict the Sequence Control field used in case Segmentation and Reassembly is supported (section 10.42 EDMG Segmentation and Reassembly).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | Start of MSDUn | End of MSDUn | MSDU Sequence Number  | MPDU Sequence Number  |
| Bits: |  1 | 1 | MSDU Modulo  | MPDU Modulo |

**Figure 9-TBD1—Sequence Control field in case SAR is supported**

The Sequence Control field comprises the End of MSDUn, Start of MSDUn MSDU, MSDU Sequence Number and MPDU Sequence Number subfields.

Start of MSDUn is one bit subfields, when it set to zero it indicates that MPDU contain the first segment of MSDU. End of MSDUn is one bit subfields, when it set to zero it indicates that MPDU contain the last segment of MSDU.

MSDU Sequence Number subfield indicate the MSDU sequence number value associated with the respective MSDU segment. MPDU Sequence Number subfield indicates the MPDU sequence number that carries the MSDU segment. The length of MSDU Sequence Number and MPDU Sequence Number subfields is set by the MSDU Modulo and MPDU Modulo subfields within the SAR Configuration element as indicated in the recipient ADDBA Response when SAR is enabled according to the procedure as described in 10.24.2 Setup and modification of the block ack parameters.

**9.3.1.8.3 Compressed BlockAckReq variant**

*Add the below after the second paragraph as follows*

In case of SAR BlockAck agreement, the BAR Information field of the compressed BlockAckReq frame is as shown in Figure 9-TBD2,

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Reserved  | MSDU Starting Sequence Number  | MPDU Starting Sequence Number  |
| Bits: |  2 | MSDU Modulo  | MPDU Modulo  |

**Figure 9-TBD2—Block Ack Starting Sequence Control field in case SAR is supported**

The MSDU Starting Sequence Number subfield of the Block Ack Starting Sequence Control field contains the sequence number of the first MSDU or A-MSDU for which the BlockAckReq frame is sent, the MPDU Starting Sequence Number subfield contain the sequence number of the first segment of the respective MSDU or A-MSDU for which this BlockAckReq frame is sent. The length of MSDU Sequence Number and MPDU Sequence Number subfields is set by the MSDU Modulo and MPDU Modulo subfields within the SAR Configuration element in the recipient ADDBA Response when SAR is enabled according to the procedure as described in 10.24.2 Setup and modification of the block ack parameters. Reserved subfield bits is set to zero.

**9.3.1.9.7 EDMG Compressed BlockAck variant**

*Add the below after the second paragraph as follows*

Under SAR BlockAck agreement, the Block Ack Starting Sequence Control subfield of the BA Information field of the compressed BlockAckReq frame is as shown in Figure 9-TBD2.

The MSDU Starting Sequence Number subfield of the Block Ack Starting Sequence Control field contains the sequence number of the first MSDU or A-MSDU for which the BlockAck frame is sent, the MPDU SSN subfield contains the sequence number of the segment of the respective MSDU or A-MSDU for which this BlockAck frame is sent. Reserved subfield bits are set to zero.

*Add the following section*

**9.4.2.139 Segmentation and Reassembly (SAR) Configuration element**

The Segmentation and Reassembly (SAR) Configuration element is shown in Figure 9-TBD3.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Element ID | Length  | SAR Configuration  |
| Octets: | 1 | 2 | 2 |

**Figure 9.TBD3—Segmentation and Reassembly Configuration element format**

The Element ID and Length fields are defined in 9.4.2.1 and should be filled with value 5.

The SAR Configuration field is shown in Figure 9-TBD3a.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | SAR Parameters  | MSDU Buffer Size  | MPDU Buffer Size  |
| Octets : | 1 | 2 | 2 |

**Figure 9.TBD3a—Segmentation and Reassembly Configuration element format**

The SAR Configuration element may be included in the ADDBA Request and Response frames in case the ADDBA originator requests the recipient to support segmentation and reassembly in the respective TID. SAR configuration set the specific segmentation parameters for the specific TID.

SAR Parameters field is described in Figure 9-TBD3b

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | SAR Enabled | MSDU Modulo  | MPDU Modulo  |
| Bits : | 1 | 3 | 4 |

**Figure 9.TBD3b—SAR Parameter field format**

SAR Enabled is one bit subfield indicating that Segmentation and Reassembly is enabled with specific parameters on respective TID. MSDU Modulo and MPDU Modulo subfields indicate the number of bits requested to be assigned for the MSDU Sequence Control and MPDU Sequence Control subfields in the 9.2.4.4.1 Sequence Control field structure and in the Block Ack Starting Sequence Control field respectively.

The MSDU Buffer Size subfield indicates the number of buffers available for this particular TID, each buffer is capable of holding a number of octets equal to the maximum MSDU size supported as indicated in the Segmentation and Reassembly Capability field.

The MPDU Buffer Size subfield indicates the number of buffers available for this particular TID, each buffer is capable of holding a number of octets equal to the Maximum MSDU size as indicated in Table 9-19—Maximum data unit sizes or to the value as agreed between the peers via ADDTS agreement on respective TID.

**9.4.2.250 EDMG Capabilities element**

**9.4.2.250.1 General**

*Add below row to table 2 – Capability IDs as follows*

**Table 2—Capabilities IDs**

|  |  |
| --- | --- |
| **Capability**  | **Capabilities ID** |
| Segmentation and Reassembly Capability  | TBD |

*Add the following section as follows*

**9.4.2.250.X Segmentation and Reassembly (SAR) Capability**

The Segmentation and Reassembly capability field is defined in Figure TBD

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Reserved | Maximum Segmented MSDU Exponent  | Segmentation and Reassembly support |
| Bits: | 3 | 4 | 1 |

**Figure TBD—Segmentation and Reassembly capability field format**

The Segmentation and Reassembly support subfield is defined in Table 9.TBD4

|  |  |  |
| --- | --- | --- |
| Subfield | Definition | Encoding |
| Segmentation and Reassembly support | Indicates that station supports the Segmentation and Reassembly as described in section 10.42 | ‘0’ - Segmentation and Reassembly (SAR) is not supported by STA‘1’ - Segmentation and Reassembly (SAR) is supported by STA |
| Maximum Segmented MSDU Exponent  | Indicates the maximum MSDU size supported when Segmentation is enabled. | This subfield is an integer in the range 0 to 9. The Maximum Segmented MSDU size that is defined by this subfield is equal to:2 (13 + Maximum Segmented MSDU Exponent) -1 octets |

**Table 9.TBD4—Segmentation and Reassembly support subfield**

Note: Fragmentation is not supported by the STA if Segmentation and Reassembly is set.

**9.6.5.2 ADDBA Request frame format**

*Change table 9-303 as follow*

**Table 9-303—ADDBA Request frame Action field format**

|  |  |
| --- | --- |
| **Order**  | **Information** |
| 1 | Category |
| 2 | Block Ack Action |
| 3 | Dialog Token |
| 4 | Block Ack Parameter Set |
| 5 | Block Ack Timeout Value |
| 6 | Block Ack Starting Sequence Control |
| 7 | GCR Group Address element (optional) |
| 8 | Multi-band (optional) |
| 9 | TCLAS (optional) |
| 10 | ADDBA Extension (optional) |
| 11 | SAR Configuration (optional) |

*Change paragraph 7 as follow*

In case originator want to enable SAR on respective TID(section 10.42 EDMG Segmentation and Reassembly), it shall include the SAR Configuration IE in the ADDBA Request. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control field (see Figure 9-TBD2) contains the sequence number of the first MSDU to be sent under this block ack agreement, the MPDU sequence number is set to the first MSDU segment of the respective indicated MSDU or A-MSDU. Otherwise, The Starting Sequence Number subfield of the Block Ack Starting Sequence Control field (see Figure 9-27) contains the sequence number of the first or next (in the case of a renegotiation of a block ack agreement) MSDU to be sent under this block ack agreement. The Fragment Number subfield is set to 0.

**9.6.5.3 ADDBA Response frame format**

*Change table 9-304 as follow*

**Table 9-304—ADDBA Response frame Action field format**

|  |  |
| --- | --- |
| **Order** | **Information** |
| 1 | Category |
| 2 | Block Ack Action |
| 3 | Dialog Token |
| 4 | Status Code |
| 5 | Block Ack Parameter Set |
| 6 | Block Ack Timeout Value |
| 7 | GCR Group Address element (optional) |
| 8 | Multi-band (optional) |
| 9 | TCLAS (optional) |
| 10 | ADDBA Extension (optional) |
| 11 | SAR Configuration (optional) |

*Add last paragraph as follow*

The SAR Configuration element is defined in 9.4.2.TBD and is included in case SAR Configuration field was included in the ADDBA Request.

**10.3.2.11.2 Transmitter requirements***Change the first paragraph as follows*

A STA maintains one or more sequence number spaces that are used when transmitting a frame to determine the sequence number for the frame. When multiple sequence number spaces are supported, the appropriate sequence number space is determined by information from the MAC control fields of the frame to be transmitted. Except as noted below, each sequence number space is represented by a modulo 4096 counter, starting at 0 and incrementing by 1, for each MSDU or MMPDU transmitted using that sequence number space.

NOTE – Under SAR BlockAck agreement, the MPDU sequence number is represented by a 2^MPDU Modulo counter and MSDU sequence number is represented by a 2^MSDU Modulo

**10.24 Block acknowledgment (block ack)**

**10.24.1 Introduction**

*Add below text after the following paragraph as follows*

All operations on sequence numbers are performed modulo 212. Comparisons between sequence numbers are circular modulo 212, 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 212).

Under SAR BlockAck agreement, operations on MSDU Sequence Number and MPDU Sequence Number are performed MSDU Modulo and MPDU Modulo respectively (section 10.42 EDMG Segmentation and Reassembly).

**10.24.2 Setup and modification of the block ack parameters**

*Add before last paragraph*

Originator may include the SAR Configuration element in the ADDBA Request only if the recipient set the Segmentation and Reassembly support subfield to ‘1’ within the Segmentation and Reassembly capability field.

If SAR Enabled was set in ADDBA Request, the originator shall include MSDU Modulo and MPDU Modulo values in the ADDBA Request, MPDU modulo shall be equal or smaller than 12 and the sum of MPDU Modulo and MSDU Modulo shall be equal to 14. The recipient may accept the ADDBA Request SAR configuration and set the SAR Enabled bit in the SAR Configuration element within the ADDBA Response to ‘1’. In this case, the recipient shall include the same MSDU Modulo and MPDU Modulo subfields of the ADDBA Request frame to confirm the size of the MPDU Sequence Number and MSDU Sequence Number subfield that in the Sequence Control field and the size of the MSDU Starting Sequence Number and MPDU Starting Sequence Number of the Block Ack Starting Sequence Control field respectively. Responder may reject the SAR configuration by resetting the SAR Enabled bit within the SAR Configuration element in the ADDBA Response.

When a SAR block ack agreement is established, the originator may change the size of its transmission window if the value in the MPDU Buffer Size or MSDU Buffer Size fields of the ADDBA Response frame are larger than the value in the ADDBA Request frame. If the value in the MSDU Buffer Size or the MPDU Buffer Size fields of the ADDBA Response frame is smaller than the value in the ADDBA Request frame, the originator shall change the size of its transmission window (WinSizeO or WinSizeOJ) so that it is not greater than the value in the respective field of the ADDBA Response frame and are not greater than the value of 1024. MPDU Buffer Size shall not exceed the value of 2MPDU\_modulo-2 and shall be equal or greater than the integer value of the Maximum Segmented MSDU size as indicated by the Maximum Segmented MSDU Exponent in the Segmentation and Reassembly Capability field divided by the Maximum MSDU size Maximum MSDU size as indicated in Table 9-19—Maximum data unit sizes or as agreed between the peers via ADDTS agreement on respective TID. Initiator may set the MSDU Buffer Size to be greater than 2^MSDU Modulo only if MPDU Modulo set to be greater than 9, in this case the receiver may receive multiple MSDUs with identical MSDU SN into it’s receive buffer.

**10.24.4 Receive buffer operation**

*Add after the first paragraph*

For each block ack agreement, the recipient maintains a MAC variable NextExpectedSequenceNumber. The NextExpectedSequenceNumber is initialized to the value of the Block Ack Starting Sequence Control field of the ADDBA Request frame of the accepted block ack agreement.

Under SAR BlockAck agreement, the NextExpectedSequenceNumber is initialized to the value of the MSDU Start Sequence Number subfields of Block Ack Starting Sequence Control field of the ADDBA Request frame of the accepted block ack agreement.

If, after an MPDU is received, the receive buffer is not full, but the sequence number of the complete MSDU or

A-MSDU in the buffer with the lowest sequence number is equal to the NextExpectedSequenceNumber for that block ack agreement, then the MSDU shall be passed up to the next MAC process.

**10.24.5 Teardown of the block ack mechanism**

*No changes in the subclause*

**10.24.6 Selection of BlockAck and BlockAckReq variants**

*No changes in the subclause.*

**10.24.7 HT-immediate block ack extensions**

**10.24.7.1 Introduction to HT-immediate block ack extensions**

*Add after the last paragraph*

HT-immediate block ack agreement in which the SAR Configuration element was included and the SAR Enable subfield in was set to ‘1’ in both the ADDBA Request and ADDBA Response is considered also as SAR block Ack agreement.

**10.24.7.2 HT-immediate block ack architecture**

*Changes in the subclause as follow*

*WinStartO* is the starting sequence number of the transmit window, and *WinSizeO* is the number of buffers negotiated in the block ack agreement.

In SAR BlockAck agreement, the originator contains a transmit buffer control that uses *WinStartOJ* and *WinSizeOJ* to submit MPDUs for transmission and releases transmit buffers upon receiving BlockAck frames from the recipient. *WinStartOJ* is the MSDU SSN value of the transmit window, and *WinSizeOJ* is the MSDU Buffer Size advertised by the responder in the SAR BlockAck agreement.

For each HT-immediate block ack agreement, the recipient chooses either full-state or partial-state operation (this choice is known only to the recipient). A STA may simultaneously use full-state operation for some agreements and partial-state operation for other agreements. The scoreboard context control stores an acknowledgment bitmap containing the current reception status of MSDUs or A-MSDUs for HT-immediate block ack agreements. Under SAR BlockAck agreement, the scoreboard context control stores an acknowledgment bitmap containing the current reception status of MPDUs. Under full-state operation, status is maintained in statically assigned memory. Under partial-state operation, status is maintained in a cache memory; therefore, the status information is subject to cache replacement. This entity provides the bitmap and the value for the Starting Sequence Number subfield to be sent in BlockAck frame responses to the originator.

**10.24.7.3 Scoreboard context control during full-state operation**

*Changes in the subclause as follow*

1. At HT-immediate block ack agreement establishment:
2. *WinStartR* = *SSN* from the ADDBA Request frame that elicited the ADDBA Response frame that established the HT-immediate block ack agreement.

In SAR BlockAck agreement, *WinStartR = MPDU SSN* from the ADDBA Request frame that elicited the ADDBA Response frame that established the SAR block ack agreement.

1. *WinEndR* = *WinStartR* + *WinSizeR* – 1
2. For each received Data frame that is related with a specific full-state operation HT-immediate block ack agreement, the block acknowledgment record for that agreement is modified as follows, where *SN* is the value of the Sequence Number subfield of the received Data frame.
3. For each received Data frame that is related with a specific full-state operation of SAR block ack agreement, the block acknowledgment record for that agreement is modified as follows, where *MPDU SN* is the value of the MPDU Sequence Number subfield of the received Data frame:
4. For each received BlockAckReq frame that is related with a specific full-state operation HTimmediate block ack agreement that is not a protected block ack agreement, the block acknowledgment record for that agreement is modified as follows, where *SSN* is the value from the Starting Sequence Number subfield of the received BlockAckReq frame

For each received BlockAckReq frame that is related with a specific full-state operation of SAR block ack agreement that is not a protected block ack agreement, the block acknowledgment record for that agreement is modified as follows, where *MPDU SSN* is the value from the MPDU Start Sequence Number subfield of the received BlockAckReq frame:

**10.24.7.4 Scoreboard context control during partial-state operation**

*Changes in the subclause as follow*

A STA implementing partial-state operation for an HT-immediate block ack agreement shall maintain the temporary block acknowledgment record for that agreement according to the following rules:

1. During partial-state operation, *WinStartR* is determined by the Sequence Number subfield value of received Data frames and by the Starting Sequence Number subfield value of received BlockAckReq frames as described below.

In SAR BlockAck agreement, *WinStartR = MPDU SSN* from the ADDBA Request frame that elicited the ADDBA Response frame that established the SAR block ack agreement.

1. For each received Data frame that is related with a specific partial-state operation HT-immediate block ack agreement, when no temporary record for the agreement related with the received Data frame exists at the time of receipt of the Data frame, a temporary block acknowledgment record is created as follows, where SN is the value of the Sequence Number subfield of the received Data frame:

For each received Data frame that is related with a specific partial-state operation SAR block ack agreement, when no temporary record for the agreement related with the received Data frame exists at the time of receipt of the Data frame, a temporary block acknowledgment record is created as follows, where *MPDU SN* is the value of the MPDU Sequence Number subfield of the received Data frame:

1. For each received BlockAckReq frame that is related with a specific partial-state operation HTimmediate block ack agreement that is not a protected block ack agreement, when no temporary record for the agreement related with the received frame exists at the time of receipt of the frame, a temporary block acknowledgment record is created as follows, where SSN is the starting value of the Sequence Number subfield of the received BlockAckReq frame:

For each received BlockAckReq frame that is related with a specific partial-state operation SAR block ack agreement that is not a protected block ack agreement, when no temporary record for the agreement related with the received frame exists at the time of receipt of the frame, a temporary block acknowledgment record is created as follows, where *MPDU SSN* is the value from the MPDU Start Sequence Number subfield of the received BlockAckReq frame:

**10.24.7.5 Generation and transmission of BlockAck frames by an HT STA or DMG STA**

*No changes in the subclause*

**10.24.7.6 Receive reordering buffer control operation**

**10.24.7.6.1 General**

*Add at end of the sub clause*:

The behavior described in this subclause, 10.24.7.6.2, and 10.24.7.6.3 applies to a STA that uses either partial-state operation or full-state operation for a SAR block ack agreement.

A receive reordering buffer shall be maintained for each SAR block ack agreement. Each receive reordering buffer includes a record comprising the following:

* Buffered MPDUs belongs to MSDUs that have been received, but not yet passed up to the next MAC process
* A *WinStartB* parameter, indicating the value of the MPDU Sequence Number subfield of the first (in order of ascending sequence number) MPDU that has not yet been received.
* A *WinEndB* parameter, indicating the highest MPDU sequence number expected to be received in the current reception window
* A *WinSizeB* parameter, indicating the size of the reception window
* 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

*WinStartB* is initialized to the MPDU Starting Sequence Number subfield value of the ADDBA Request frame that elicited the ADDBA Response frame that established the SAR block ack agreement.

*WinEndB* is initialized to *WinStartB* + *WinSizeB* – 1, where *WinSizeB* is set to the smaller of 1024 and the value of the MPDU Buffer Size field of the ADDBA Response frame that established the block ack agreement.

*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 SAR block ack agreement.

Any MSDU that has been passed up to the next MAC process shall be deleted from the receive reordering buffer.

The recipient shall always pass MSDUs or A-MSDUs up to the next MAC process in order of increasing MSDU

Sequence Number subfield value.

**10.24.7.6.2 Operation for each received Data frame**

*Add at end of the sub clause:*

For each received data MPDU that is related to a specific SAR Block Ack agreement, the receive reordering buffer record shall be modified as follows, where *MPDU\_SN* is the value of the Sequence Number subfield of the received MPDU:

1. If *WinStartB* ≤ *MPDU\_SN* ≤*WinEndB*,
2. Store the received MPDU in the buffer, if no MPDU with the same sequence number is already present; otherwise discard the MPDU.
3. Pass to the next MAC process any complete MSDUs in buffer which include all MPDUs with the same value of *MSDU\_SN* in increasing order of *MPDU\_SN* from the *MPDU\_SN* set with *Start of MSDUn*=true till MPDU with *End of MSDUn*=true and which their *MPDU\_SN* starting with *MPDU\_SN=WinStartB* and proceeding sequentially until there is no buffered MPDUs for the next *MPDU\_SN*.
4. Set *WinStartB* to the value of the *MPDU\_SN* subfield of the last MPDU of the MSDU that was passed up to the next MAC process plus one.

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.

5) Set *WinEndB* = *WinStartB* + *WinSizeB* – 1.

1. If *WinEndB* < *MPDU\_SN* *WinStartB* + 2MPDU\_Modulo-1,
2. Store the received MPDU in the buffer, if no MPDU with the same sequence number is already present; otherwise discard the MPDU.
3. Set *WinEndB = MPDU\_SN.*
4. Set *WinStartB = WinEndB – WinSizeB + 1.*
5. Pass to the next MAC process any complete MSDUs in buffer which include all MPDUs with the same value of *MSDU\_SN* in increasing order of *MPDU\_SN* from the *MPDU\_SN* set with *Start of MSDUn*=true till next MPDU with *End of MSDUn*=true and which their *MPDU\_SN* is lower than WinStartB. Gaps may exist in the *MPDU\_SN* subfield, in this case the recipient shall discard the MSDUs with missing *MPDU\_SN* and may inform the next MAC process for the incomplete or missing *MSDU SN*.
6. Set *WinStartB* to *MPDU\_SN* value of the last MSDU or A-MSDU that was passed up to the next MAC process plus one.
7. Set *WinEndB = WinStartB + WinSizeB – 1.*
8. If *WinStartB* + 2MPDU\_Modulo-1 *≤ MPDU\_SN*  *WinStartB* , discard the MPDU (do not store the MPDU in the buffer.

**10.24.7.6.3 Operation for each received BlockAckReq**

*Add at end of the sub clause:*

For each received BlockAckReq frame that is related with a specific SAR block ack agreement, the receive reordering buffer record is modified as follows, where *MPDU\_SSN* and *MSDU\_SSN* are in the Starting Sequence Number subfield value of the received BlockAckReq frame:

1. If *WinStartB* < *MPDU\_SSN* < *WinStartB* + 2MPDU\_Modulo-1,
2. In a block ack agreement that is not a protected block ack agreement, set *WinStartB* = *MPDU\_SSN* and follow rules in the sub clause*.* See 10.24.9 for a protected block ack agreement.
3. Set *WinEndB = WinStartB – WinSizeB - 1.*
4. Pass to the next MAC process any complete MSDUs in buffer which include all MPDUs with the same value of *MSDU\_SN* in increasing order of *MPDU\_SN* from the *MPDU\_SN* set with *Start of MSDUn*=true till MPDU with *End of MSDUn*=true and which their MPDU\_SN is lower than *WinStartB*. Gaps may exist in the *MPDU\_SN* subfield, in this case the recipient shall discard the MSDUs with missing *MPDU\_SN* and may inform the next MAC process for the incomplete or missing MSDUs.
5. Pass to the next MAC process any complete MSDUs in buffer which include all MPDUs with the same value of *MSDU\_SN* in increasing order of *MPDU\_SN* from the *MPDU\_SN* set with *Start of MSDUn*=true till MPDU with *End of MSDUn*=true and which their *MPDU\_SN* starting with *MPDU\_SN=WinStartB* and proceeding sequentially until there is no buffered MPDUs for the next *MPDU\_SN*.
6. Set *WinStartB* to *MPDU\_SN* value of the last MSDU or A-MSDU that was passed up to the next MAC process plus one.
7. Set *WinEndB = WinStartB + WinSizeB – 1.*
8. If *WinStartB* + 2MPDU\_Modulo-1 ≤ *MPDU\_SSN* < *WinStartB* , do not make any changes to the receive reordering buffer record

**10.24.7.7 Originator’s behavior**

*Add as indicate:*

The originator may transmit QoS Data frames with a TID matching an established block ack agreement in any order provided that their sequence numbers lie within the current transmission window. The originator may transmit an MPDU with a sequence number that is beyond the current transmission window (*SN* > *WinStartO + WinSizeO – 1*), in which case the originator’s transmission window (and the recipient’s window) is moved forward. The originator should not transmit MPDUs that are lower than (i.e., *SN* < *WinStartO*) the current transmission window.

Under SAR block ack agreement the originator shall not transmit MPDU with a *MPDU\_SN* beyond the current MPDU transmission window (*WinStartO* <*MPDU\_SN =*< *WinStartO +WinSizeO – 1*) and shall not transmit MPDU with a *MSDU\_SN* beyond the current MSDU transmission window (*WinStartOJ =<MSDU\_SN =*< *WinStartOJ +WinSizeOJ – 1*) . All MPDU\_SN and MSDU\_SN shall be assigned with increasing sequential values.

The originator may send a BlockAckReq frame for block ack agreement that is not a protected block ack agreement or a robust ADDBA Request frame for protected block ack agreement when a Data frame that was previously transmitted within an A-MPDU that had the Ack Policy field equal to Normal Ack is discarded due to exhausted MSDU lifetime. The purpose of this BlockAckReq or robust ADDBA Request frame is to shift the recipient’s *WinStartB* value past the hole in the sequence number space that is created by the discarded Data frame and thereby to allow the earliest possible passing of buffered frames up to the next MAC process.

Under SAR block ack agreement the BlockAckReq or robust ADDBA Request frame shall contain only MPDU\_SSN and MSDU\_SSN fields of MPDU set with Start of MSDUn = true.

**10.24.9 Protected block ack agreement**

*Add as indicate:*

A STA that has successfully negotiated a protected block ack agreement shall obey the following rule as a block ack originator in addition to rules specified in 10.24.7.7 and 10.24.7.8:

* To change the value of WinStartB at the receiver, the STA shall use a robust ADDBA Request frame.

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 10.24.7.3 to 10.24.7.6:

* 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*. The Block Ack Starting Sequence Control subfield value may be utilized for the purposes of updating the value of *WinStartR*. If the Block Ack Starting Sequence Control subfield value is greater than *WinEndB* or less than *WinStartB* or in case of SAR block ack agreement if MPDU Start Sequence in the Block Ack Starting Sequence Control subfield value is greater than *WinEndB* or less than *WinStartB*, than dot11PBACErrors shall be incremented by 1.
* Upon receipt of a valid robust ADDBA Request frame for an established protected block ack agreement whose TID and transmitter address are the same as those of the block ack agreement, the 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 10.24.7.3, 10.24.7.4, 10.24.7.6.1, and 10.24.7.6.3, while treating the starting sequence number as though it were the SSN or in case of SAR block ack agreement treating the MPDU starting sequence number as though it were the MPDU SSN of a received BlockAckReq frame. Values in other fields of the ADDBA Request frame shall be ignored

*Add the following section as follows***10.42 EDMG Segmentation and Reassembly (SAR)**

**10.42.1 General**

EDMG STA may segment and reassemble large size MSDUs received from the upper layer, MSDU segments are carried in individually addressed MPDUs. The segmentation and reassembly mechanisms allow the source STA to receive from the upper layer a MSDU in size supported by its communicating peer and to deliver it segmented into MPDUs over the wireless link. The recipient STA is responsible to reassemble the segmented MSDUs into their original size of the MSDU and forward it to the upper layer. Collectively the mechanism allow sending large datagrams over the wireless link efficiently and without further L3/L4 segmentation processing.

EDMG STA that supports Segmentation and Reassembly capability may set the Segmentation and Reassembly capability subfield in the EDMG Capabilities element to ‘1’ and advertise the Maximum Segmented MSDU size it supports. An originator that intends to use the segmentation mechanism for the transmission of MSDUs to an intended recipient should first check that the recipient respective Segmentation and Reassembly capability subfield in the EDMG Capabilities element is set to ‘1’. If the intended recipient STA is capable of supporting segmentation, the originator may enable the SAR functionality on specific TID and initiate with the peer a Segmentation and Reassembly (SAR) BlockAck Agreement to set the SAR configuration.

Below diagram depict and example of Segmented MSDU delivered between two non-segmented MSDUs, the segmented MSDU is comprised of four segments.



**Figure TBD— Segmentation and Reassembly Procedure**

**10.42.2 Segmentation Operation**

The originator segments the MSDU into one or several MPDUs, each MPDU frame body carry MSDU segment which may be all or a portion of an MSDU. Once a segment is transmitted for the first time, its frame body content and length shall be fixed until it is successfully delivered to the immediate receiving STA.

MPDU contains the first segment of MSDU shall be assigned with Start of MSDUn equal to ‘0’ and shall be set to ‘1’ otherwise. MPDU contain the last segment of MSDU shall be assigned with End of MSDUn equal to ‘0’ and shall be set to ‘1’ otherwise. MPDUs set with End of MSDUn equal to ‘0’ and Start of MSDUn equal to ‘0’ contain whole single MSDU. MPDUs set with End of MSDUn equal to ‘0’ and Start of MSDUn equal to 1 shall be in size of Maximum MSDU size as indicated in Table 9-19—Maximum data unit sizes or as agreed between the peers via ADDTS agreement on respective TID. All MPDUs assigned MSDU Sequence Number and MPDU Sequence Number values shall be set with continuously increasing sequential values.

**10.42.3 Reassembly Operation**

The destination STA reassembles the segmented MSDU according to Start of MSDUn, End of MSDUn, MSDU Sequence Number and MPDU Sequence Number indications and the Sequence Number subfield as described in 10.24.7.6.2 Operation for each received Data frame. Recipient shall reconstruct the segmented MSDU with the MAC Header of the first segmented MSDU contained in the MPDU assigned with Start of MSDUn equal to ‘0’ and shall remove the MAC Header field from all the received MPDU assigned with Start of MSDUn equal to ‘1’. If security encapsulation has been applied to the MPDU segment, it shall be de-encapsulated and decrypted before the segment is used for reassembly of the MSDU. Destination STA that supports segmentation and reassembly shall be capable to support a reception of segments of multiple MSDUs containing all or part of an MSDU, plus any security encapsulation overhead, MAC header and FCS and to reassemble it back to the original MSDU size.

**SP: Do you agree to integrate the text suggested in “11-17-1414-00-00ay-EDMG Segmentation and Reassembly.docx” into the 11ay draft ?**