IEEE P802.11
Wireless LANs

|  |
| --- |
| A-MSDU Fragmentation |
| Date: 2016-11-05 |
| Author(s): |
| Name | Affiliation | Address | Phone | Email |
| Matthew Fischer | Broadcom Ltd. | 190 Mathilda Pl, Sunnyvale, CA 94086 |  | matthew.fischer@broadcom.com |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Abstract

This submission proposes the addition of a capability bit to signal support for receipt of fragmented A-MSDUs.

**REVISION NOTES:**

**R0: initial**

Interpretation of a Motion to Adopt

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

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

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

**Discussion:**

* **Various levels of dynamic fragmentation defined**
	+ **Intended to allow implementation choice of tradeoff**
		- **Complexity/cost**
		- **Performance**
* **Dynamic Fragmentation Support level is advertised**
	+ **To allow full interoperability at the highest common level**
* **AMSDU is currently NOT allowed to be fragmented**
	+ **See later slides for details**
* **Desire to optionally support AMSDU Fragmentation**
	+ **To allow more flexible design**

**CURRENT STATUS**

* **D0.5**
* **25.3 Fragmentation**
	+ **Language always refers to MSDU and MMPDU, not AMSDU**
	+ **i.e. AMSDU fragmentation is implicitly forbidden**
* **Baseline (REVmcD8.0)**
* **10.12 A-MSDU Operation**
	+ An A-MSDU shall be carried, without fragmentation, within a single QoS Data frame

**Rationales for existing restriction of A-MSDU fragmentation**

* **Just aggregated something, now fragment – seems counter productive**
* **Dynamic fragmentation includes a minimum fragment size**
	+ **Applied only to FIRST fragment**
	+ **Allows recipient to examine specific minimum number of bytes of MAC payload to identify L3/L4 header information**
		- **I.e. first N bytes of the MSDU**
		- **Allows MAC to select RX Buffer queue that is appropriate for the L3/L4 protocol at first fragment arrival**
	+ **This technique fails if the first fragment size applies to MPDU which contains more than one MSDU (i.e. A-MSDU)**
		- **Non-first MSDUs in the A-MSDU will not necessarily have the desired minimum fragment size of bytes lined up with the first bytes of MSDU payload**

**Contradiction with dynamic fragmentation**

* **For Level0, 1, 2, the restriction is effective**
* **For Level3 the restriction does not matter**
	+ **Level3 allows 4 fragments for a single MSDU to appear within a single AMPDU**
	+ **It is possible for a receiver to lose the first fragment (F0) of an MSDU but receive at least one of: F01, F02, F03 for the same MSDU**
		- **In this case, the receiver must buffer F01, F02, F03 in an intermediate location because it does not have L3/L4 header information available to identify the correct RX Buffer for this MSDU**
* **Level3 fragmentation already allows the behavior that the AMSDU restriction is intended to prevent**

**A-MSDU Fragmentation Proposal**

* **AMSDU fragmentation would be a recipient choice**
* **Complementary to Level0,1,2,3 selection**
* **Optional support**
* **Allows more choices at transmitter**
* **Transmitter can apply AMSDU aggregation without having to prevent fragmentation that might occur due to dynamic RU allocations**
* **If RU allocations are large, then transmitter prefers to use AMSDU to maximize performance**
	+ **E.g. small L3/L4 MSDUs, e.g. TCP ACK**
* **Because a transmitter cannot predict allocations**
	+ **The transmitter is forced to not use AMSDU**

**PROPOSED CHANGES:**

**9.4.2.218 HE Capabilities element**

**TGax Editor: *Add a new subfield to the HE Capabilities information element “A-MSDU Fragmentation Support”***

**TGax Editor: *Add a new row to Table 9-262z – Subfields of the HE MAC Capabilities Information field of subclause 9.4.2.218 HE Capabilities information element as shown:***

|  |  |  |
| --- | --- | --- |
| **Subfield** | **Definition** | **Encoding** |
| A-MSDU Fragmentation Support | The A-MSDU Fragmentation Support subfield indicates support for the reception of fragmented A-MSDUs.  | Set to 1 to indicate support for the receipt of fragmented A-MSDUs. Set to 0 to indicate that reception of fragmented A-MSDUs is not supported. |

**10.5 Fragmentation**

**TGax Editor: *Modify subclause 10.5 Fragmentation as shown:***

The MAC may fragment and reassemble MSDUs or MMPDUs that are carried in individually addressed MPDUs. The fragmentation and defragmentation mechanisms allow for fragment retransmission. The MAC may fragment A-MSDUs when the recipient has indicated support for the receipt of fragmented A-MSDUs.

A STA that has a value of true for dot11AMSDUFragmentationOptionImplemented shall set the A-MSDU Fragmentation Support subfield to 1 in transmitted HE Capability elements. Otherwise, the STA shall set the A-MSDU Fragmentation Support subfield to 0 in transmitted HE Capability elements.

With static fragmentation, t~~T~~he length of each fragment shall be an equal number of octets for all fragments except the last, which may be smaller. The length of each fragment shall be an even number of octets, except for the last fragment of an MSDU or MMPDU, which may be either an even or an odd number of octets. The length of a static fragment shall never be larger than dot11FragmentationThreshold unless security encapsulation is invoked for the MPDU. If security encapsulation is active for the MPDU, then the MPDU shall be expanded by the encapsulation overhead and this may result in a fragment larger than dot11FragmentationThreshold. Unless the conditions described in 25.3.2 (Support and requirements for dynamic fragmentation) are met, static fragmentation is used.

A fragment is an MPDU, the Frame Body field of which carries all or a portion of an MSDU, A-MSDU or MMPDU. When data are to be transmitted, the number of octets in the fragment (before processing by the security mechanism) shall be limited by dot11FragmentationThreshold and the number of octets in the MPDU that have yet to be assigned to a fragment at the instant the fragment is constructed for the first time. Once a fragment 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.

A STA shall be capable of receiving fragments, containing all or part of an MSDU, of arbitrary length that is less than or equal to the maximum MSDU size as specified in Table 9-19 (Maximum data unit sizes (in octets) and durations (in microseconds)).~~defined in 9.2.3 (General frame format), plus any security encapsulation overhead, plus MAC header and FCS~~. A STA with a value of true for dot11AMSDUFragmentationOptionImplemented shall be capable of receiving fragments containing all or part of an A-MSDU of arbitrary length that is less than or equal to the maximum A-MSDU size as specified in Table 9-19 (Maximum data unit sizes (in octets) and durations (in microseconds)).

A STA shall be capable of receiving fragments, containing all or part of an MMPDU, of arbitrary length that

is less than or equal to the minimum of

— The maximum MMPDU size as defined in 9.3.3.2 (Format of Management frames), plus any security encapsulation overhead, plus MAC header and FCS

— Any maximum MPDU length advertised by the STA

If a fragment requires retransmission, its frame body content and length shall remain fixed for the lifetime of the MSDU, A-MSDU or MMPDU at that STA. Each fragment shall contain a Sequence Control field, which is comprised of a sequence number and fragment number. When a STA is transmitting an MSDU, A-MSDU or MMPDU, the sequence number shall remain the same for all fragments of that MSDU, A-MSDU or MMPDU. The fragments shall be sent in order of lowest fragment number to highest fragment number, where the fragment number value starts at 0, and increases by 1 for each successive fragment. The Frame Control field also contains a bit, the More Fragments bit, that is equal to 0 to indicate the last (or only) fragment of the MSDU, A-MSDU or MMPDU.

The source STA shall maintain a transmit MSDU timer for each MSDU being transmitted. The attribute dot11MaxTransmitMSDULifetime specifies the maximum amount of time allowed to transmit an MSDU. The timer starts on the initial attempt to transmit the first fragment of the MSDU or A-MSDU containing the MSDU. If the timer exceeds dot11MaxTransmitMSDULifetime, then all remaining fragments are discarded by the source STA and no attempt is made to complete transmission of the MSDU.

NOTE—A STA might interleave fragments of MSDUs or A-MSDUs with different TIDs sent to the same receiver, subject to any constraint caused by the number of replay counters.

**TGax Editor: *Modify subclause 10.6 Defragmentation as shown:***

**10.6 Defragmentation**

Each fragment contains information to allow the complete MSDU, A-MSDU or MMPDU to be reassembled from its constituent fragments. The header of each fragment contains the following information that is used by the destination STA to reassemble the MSDU, A-MSDU or MMPDU:

— Frame type

— Address of the sender, obtained from the Address 2 field

— Destination address

— *Sequence Control field:* This field allows the destination STA to check that all incoming fragments belong to the same MSDU, A-MSDU or MMPDU, and the sequence in which the fragments should be reassembled. The sequence number within the Sequence Control field remains the same for all fragments of an MSDU, A-MSDU or MMPDU, while the fragment number within the Sequence Control field increments for each fragment.

— Traffic identifier, for frames with a QoS Control field.

— *More Fragments indicator:* Indicates to the destination STA that this is not the last fragment of the MSDU, A-MSDU or MMPDU. Only the last or sole fragment of the MSDU, A-MSDU or MMPDU shall have this bit set to 0. All other fragments of the MSDU or MMPDU shall have this bit set to 1.

The destination STA shall reconstruct the MSDU, A-MSDU or MMPDU by combining the fragments in order of fragment number subfield of the Sequence Control field. If security encapsulation has been applied to the fragment, it shall be deencapsulated and decrypted before the fragment is used for defragmentation of the MSDU, A-MSDU or MMPDU. If the fragment with the More Fragments bit equal to 0 has not yet been received, then the destination STA knows that the MSDU, A-MSDU or MMPDU is not yet complete. As soon as the STA receives the fragment with the More Fragments bit equal to 0, the STA knows that no more fragments may be received for the MSDU, A-MSDU or MMPDU.

A STA shall support the concurrent reception of fragments of at least three MSDUs, A-MSDUs or MMPDUs. A STA should support the concurrent reception of fragments of at least one MSDU or A-MSDU per access category. An AP should support the concurrent reception of at least on MSDU or A-MSDU per access category per associated STA. Note that a STA receiving more than three fragmented MSDUs, A-MSDUs or MMPDUs concurrently might experience a significant increase in the number of frames discarded.

NOTE—The three MSDUs, A-MSDUs or MMPDUs might be from different peers (e.g., in an IBSS or MBSS).

The destination STA shall maintain a Receive Timer for each MSDU or MMPDU being received, for a minimum of three MSDUs or MMPDUs. The STA may implement additional timers to be able to receive additional concurrent MSDUs or MMPDUs. The receiving STA shall discard all fragments that are part of an MSDU or MMPDU for which a timer is not maintained. There is also dot11MaxReceiveLifetime, that specifies the maximum amount of time allowed to receive an MSDU. The receive MSDU or MMPDU timer starts on the reception of the first fragment of the MSDU or MMPDU. If the receive MSDU timer exceeds dot11MaxReceiveLifetime, then all received fragments of this MSDU or MMPDU are discarded by the destination STA. If additional fragments of an individually addressed MSDU or MMPDU are received after its dot11MaxReceiveLifetime is exceeded, those fragments shall be acknowledged and discarded.

To properly reassemble MPDUs into an MSDU, A-MSDU or MMPDU, a destination STA shall discard any duplicated fragments received. A STA shall discard duplicate fragments as described in 10.3.2.11 (Duplicate detection and recovery). However, an acknowledgment shall be sent in response to a duplicate fragment of an individually addressed MSDU, A-MSDU or MMPDU.

**TGax Editor: *Modify the third paragraph of subclause 10.12 A-MSDU operation as shown:***

**10.12 A-MSDU operation**

An A-MSDU shall be carried, without fragmentation, within a single QoS Data frame, when the recipient has not indicated support for reception of fragmented A-MSDUs. An A-MSDU may be fragmented and each fragment transmitted within a single QoS Data frame, when the recipient has indicated support for reception of fragmented A-MSDUs. Support for reception of fragmented A-MSDUs is indicated with the Fragmented A-MSDU Support bit of the HE Capabilities element.

**TGax Editor: *Add a new MIB variable in C.3 MIB Detail within the dot11HEStationConfigEntry group as shown:***

**C.3 MIB Detail**

dot11HEDynamicFragmentationImplemented OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is a capability variable. Its value is determined by device capabilities.

This attribute, when true, indicates that the STA implementation is capable of receiving dynamic fragments. The capability is disabled, otherwise"

DEFVAL { false }

::= { dot11HEStationConfigEntry <XX>}