IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolutions for MAC Operation Adrian comments on 11mc/D4.0 | | | | |
| Date: 2015-09-08 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Graham SMITH | SRT Wireless | Davie, FL, USA. | 916 799 9563 | gsmith@srtrl.com |

Abstract

This submission proposes resolutions for MAC Comments on 11mc/D4.0.

Green indicates material agreed to in the group,

yellow material to be discussed, red material rejected by the group and

cyan material not to be overlooked.

The “Final” view should be selected in Word.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5050  Adrian  9.35.7  1447.35 | It is not clear if this subclause replaces 9.3.2.12, or is an extension to it. | If a replacement, exclude mesh STAs from 9.3.2.12. If an extension, move the mesh duplicate detection logic to become two new rows in 9.3.2.12.\* |

Discussion:

**9.35 Mesh Forwarding framework**

**9.35.7 Detection of duplicate MSDUs/MMPDUs**

“The receiving mesh STA shall keep a cache of recently received <Mesh SA, Mesh Sequence Number> tuples. The Mesh Source Address (Mesh SA) is contained in Address 4 for individually addressed mesh Data frames and Multihop Action frames. The Mesh Source Address (Mesh SA) is contained in Address 3 for group addressed mesh Data frames.

A mesh STA shall reject an MSDU/MMPDU with a Mesh Control field as a duplicate if it matches a <Mesh SA, Mesh Sequence Number> tuple of an entry in the cache.

**9.3.2.12 Duplicate detection and recovery**

Table 9-4 lists the Receiver caches

In Table 9-4 we already have Mesh GCR so it might seem logical to also include the ‘standard’ mesh STA.

Adrian: I think you just need to cite RR5, because all frames transmitted by a Mesh STA should have a mesh control field.

However, thinking about it now, I think the two might be different.

I think the 9.3.2.12 cache operates on MPDUs and the mesh cache on MSDUs, after reassembly.

So I’d suggest rejecting this:

Rejected. The mesh cache operates on MSDUs/MMPDUs and is logically at a layer above the cache in 9.3.2.12, which also applies to the fragments of those MSDUs/MMPUDs.

Proposed resolution:

REJECT.

The mesh cache operates on MSDUs/MMPDUs and is logically at a layer above the cache in 9.3.2.12, which also applies to the fragments of those MSDUs/MMPUDs.

|  |  |  |
| --- | --- | --- |
| CID 5163  Stephens, Adrian  9.35.9  1448.13 | "is not able to forward the frame" -- forwarding is at the level of MSDU or MMPDU, not frame. | change "frame" to "MSDU".  Make the same change at 1448.14 |

Discussion:

“If the source mesh STA is not able to forward the frame because its destination is unknown, the mesh STA shall assume that the destination is outside the MBSS and shall forward the frame to known mesh gates in the MBSS as an individually addressed frame according to the procedures for frame addressing and data forwarding of individually addressed frames at source mesh STAs in an MBSS (9.35.4.1 (At source mesh STAs (individually addressed))).”

We read futher that

— Address 5: The address of the destination end mesh STA, which is the unknown destination address of the MSDU

Hence, I agree with the comment that “MSDU” is the correct term, not “frame”.

Proposed Resolution

ACCEPT

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5162  Stephens, Adrian  9.35.7  1447.44 | "An RSNA defines a number of security features in addition to wired equivalent privacy (WEP) and IEEE Std 802.11 authentication. These features include the following" -- if the mesh requirements are in addition to 9.3.2.12, the best place to state them is in that subclause | Replace 9.35.7 with an additional row for transmitter sequence number and receiver cache in 9.3.2.12. |

Discussion:

The reference text is wrong, the text reference comes from 4.3.5.3 P67.3

**“4.3.5.3 Robust security network association (RSNA)**

An RSNA defines a number of security features in addition to wired equivalent privacy (WEP) and IEEE Std 802.11 authentication. These features include the following:

— Enhanced authentication mechanisms for STAs

— Key management algorithms

— Cryptographic key establishment

— Enhanced data cryptographic encapsulation mechanisms, such as counter mode with cipher-block

chaining message authentication code protocol (CCMP), Galois Counter Mode protocol (GCMP), and, optionally, temporal key integrity protocol (TKIP).

— Fast basic service set (BSS) transition (FT) mechanism

— Enhanced cryptographic encapsulation mechanisms for robust Management frames

So forget this?

Agreed. Commenter is an idiot. But see my comments above. I think this can be rejected.

Proposed resolution:

REJECT

The mesh cache operates on MSDUs/MMPDUs and is logically at a layer above the cache in 9.3.2.12, which also applies to the fragments of those MSDUs/MMPUDs.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5157  Stephens, Adrian  9.28.4  1399.08 | This list of permitted MPDUs does not include those used by a DMG STA. | Add Extended Compressed BlockAck and BlockAckReq to the list |

Discussion:

The list is:

“An RD responder shall not transmit an MPDU (either individually or aggregated within an A-MPDU) that is not one of the following frames:

— Ack

— Compressed BlockAck

— Compressed BlockAckReq

— QoS data

— Management

Note that “RD” is Reverse Direction. Basically an RD responder can immediately come back with a packet during a TXOP i.e. packets in both directions during a TXOP.

The Extended Compressed BlockAck and BlockAckReq certainly exist so why were they not included? Maybe the author only intended the compressed versions to be used in RD? I don’t know, I need to find out.

Adrian: The author of the RD text wrote it 10 years before DMG existed :0).

The authors of .11ad intend to use RD, but failed to update the list.

But the commenter is an idiot and produced an ambiguous change. You need at least to fully expand the terms.

Sent email to Carlos.

1. Reverse direction was defined as part of 11n. We’ve extended it in 11ad to meet our needs.
2. Section (**8.3.1.8.1 Overview**) states: “DMG STAs use only the Compressed BlockAckReq variant and the Extended Compressed BlockAckReq variant”. So, a DMG STA can use both the compressed and the extended compressed.
3. As a consequence of (2), a DMG STA uses the Compressed BlockAck and Extended Compressed BlockAck.

Therefore, on the basis of (2) and (3) above, the commenter is correct.

Thanks,

Carlos.

Proposed resolution:

REVISE

At 1399.1 make changes as shown:

“An RD responder shall not transmit an MPDU (either individually or aggregated within an A-MPDU) that is not one of the following frames:

— Ack

— Compressed BlockAck

— Compressed BlockAckReq

— Extended Compressed BlockAck

— Extended Compressed BlockAckReq

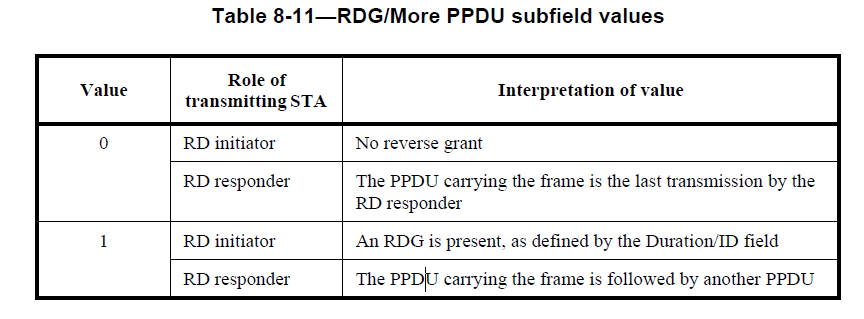
— QoS data

— Management

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5156  Stephens, Adrian  9.28.3  1398.24 | "NOTE 6--After transmitting a PPDU requiring a response but not containing an RDG, the state of the RDG/More PPDU subfield in the response does not affect the behavior of the RD initiator" -- this circumstances described in this note cannot occur if the peer is a compliant implementation.    We do not attempt to describe how the protocol responds to non-compliant peers. | Remove cited note. |

Discussion:

The HT Control field contains the RDG/More PPDU subfield and is interpreted differently depending on whether it is transmitted by an RD initiator or an RD responder.



This is within the clause “**Rules for RD initiator**.

Hence, in the case of Note 6 (no reverse grant) one would assume that the value of the field is 0.

So is Note 6 saying that if there is no RD Grant, then ignore the field?

It does seem that the Note 6 introduces confusion at the least because I for one have no idea what it is really trying to say and it definitely confuses the values in Table 8-11 indicating that if there is no Grant, then a 1 could appear.

Hence I agree with the commenter

Proposed resolution:

ACCEPT

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5155  Stephens, Adrian  9.27.6  1395.34 | This statement about ignoring unknown values should be extended to over the Element ID extension case. | Add statement that a STA that supports any Element ID that implies a n Element ID extension shall parse the Element ID extension, and skip any elements it identifies that it does not understand. |

Discussion:

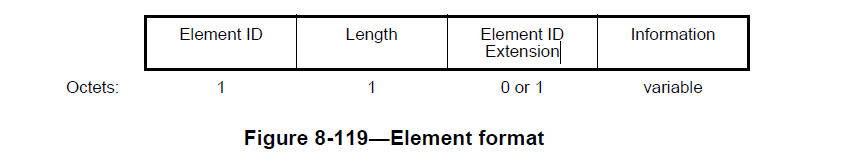
**“9.27.6 Element parsing**

A STA that encounters an unknown or reserved element ID value in a Management frame received without error shall ignore that element and shall parse any remaining management frame body for additional elements with recognizable element ID values.

The MME of a Vendor-Specific Protected Action frame is located at the end of the frame body.

NOTE—It is not necessary to be able to parse the Vendor-Specific Content to locate the MME.”

The optional Element ID extension is in the Element format 8.4.2.1



It is “N/A” for all Element IDs up to 254 and only Element ID 255 uses the Element ID Extension

This is new and therefore does need to be included in this cited BUT at the moment it is reserved and its format is not given – I would assume it will take the same form as the Element format.

So if the STA does not recognize 255, no problem but if it does recognize 255 but not the Element ID extension, then it needs to skip just that part, no change required. We do not know what goes in the Information field yet (or do we?).

BUT, if a STA identifies the 255 Element ID notes the Length, then does not recognize the Element ID extension, it should skip the next (Length -1) octets. Element ID “255” is (at the moment) “Reserved for elements using the Element ID Extension field”, which could be interpreted as “reserved’ but I suppose we had better try and cover it as the commenter proposes. So now it comes down to how to say it…

Adrian Comment (to my original text)

Our capitalization rules say this has to be lower case unless it’s quoting the name of the field, in which case “field” has to be present.

An alternavite is “… reservied value of the Element ID field, or the Element ID field equal to 255 and unknown value of the Element ID Extension field “ more words, less editorial surprise.

Proposed resolution:

REVISED

At 1395.34

Replace

“A STA that encounters an unknown or reserved element ID value in a Management frame received without error shall ignore that element and shall parse any remaining management frame body for additional elements with recognizable element ID values.”

With

“A STA that encounters an unknown or reserved element ID value, or if the value of the Element ID field is 255 and the element ID extension value is unknown or reserved, in a Management frame received without error, shall ignore that element and shall parse any remaining management frame body for additional elements with recognizable element ID values.”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5153  Stephens, Adrian  9.26.3.3  1388.24 | "A STA shall not transmit PPDUs separated by a RIFS unless the RIFS Mode field of the HT Operation element is equal to 1." -- this has nothing to do with protection | Move cited sentence to 9.3.2.3.2.    Remove any references to 9.26.3.3 that do not reference RIFS protection. |

Discussion:

True, this has nothing to do with protection**.**

**At** 1249.37 **RIFS is obsolete so should we change anything??**

Assuming the answer is “yes”.

Taking second part first.

9 other instances of “9.26.3.3” and all seem reasonably to do with RIFS protection’

890.22

1249.61

1386.21

1298.45

1401.2

1401.25

1401.30

2753.26

2753.28

Could be an ACCEPT but commenter did not specify where to insert the sentence in 9.3.2.3.2.

Proposed resolution:

REVISED

At 1388.24 delete

“A STA shall not transmit PPDUs separated by a RIFS unless the RIFS Mode field of the HT Operation element is equal to 1."

Insert new paragraph at 1249.59

“A STA shall not transmit PPDUs separated by a RIFS unless the RIFS Mode field of the HT Operation element is equal to 1."

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5150  Stephens, Adrian  9.24.10.3  1376.36 | "and 9.3.2.8 (Dual CTS protection))" -- I don't see there is any way that GCR could be used at ranges where STBC is required to close the link. There is no way to negotiate this in GCR. So this inclusion is misleading, and probably wouldn't do anything useful anyway. | Remove cited text. |

Discussion:

**9.24.10.3 GCR block ack BlockAckReq and BlockAck frame exchanges**

A protective mechanism (such as transmitting an HCCA CAP, MCCA, or RTS/CTS; setting the Duration field in the first frame and response frames to update the NAVs of STAs in the BSS and OBSS(s); or using another mechanism described in 9.26 (Protection mechanisms) and 9.3.2.8 (Dual CTS protection)) should be used to reduce the probability of other STAs transmitting during the GCR TXOP.

**9.3.2.8.1 Dual CTS protection procedure**

“If the Dual CTS Protection field of the HT Operation element has value 1 in the Beacon frames transmitted by its AP, a non-AP HT STA shall start every TXOP with an RTS frame addressed to the AP. **The RTS frame shall be an STBC frame if the STBC transmit and receive capabilities of the non-AP HT STA allow it to receive and transmit STBC frames using a single spatial stream; otherwise, the RTS frame shall be a non- STBC frame**.”

So the commenter may well be right that use of STBC and GCR are not compatable, but the RTS frame need not be STBC, so therefore, in theory Dual CTS protection may be used.

Proposed resolution:

REJECT

The commenter asserts that these mechanisms will not work, but has not shown any specific issue.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5148  Stephens, Adrian  9.24.1  1359.42 | "A DMG STA shall support the HT-immediate block ack extension." -- this statement is out of context, because 9.24.1 doesn't mention HT-immediate. | Either merge 9.24.7.1 into 9.24.1, or move the cited sentence into 9.24.7.1. The latter is my preference. |

Discussion:

Agreed, the added sentence at 1359.42

“A DMG STA shall support the HT-immediate block ack extension. A DMG STA shall not use the HT-delayed block ack extension.”

Is out of place. This is the Introduction to Block Ack so why tell us about DMG STAs here? It is not correct. Agree with commenter.

Proposed resolution:

REVISED

At 1359.42 delete

“A DMG STA shall support the HT-immediate block ack extension. A DMG STA shall not use the HT-delayed block ack extension.”

At 1366.31 insert

“A DMG STA shall support HT-immediate block ack.”

At 1374.04 insert

“A DMG STA shall not use HT-delayed block ack.”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5147  Stephens, Adrian  9.24.3  1361.57 | "Separate the Block and Basic BlockAckReq frames" - there is no such thing as a "Block" or "Block frame"  Likewise at 1361.59. | Possibly: replace "Block" by "Data". Likewise at 1361.59, replace "Block frame" with "series of Data frames related to a single block ack agreement"  But the third list item appears to duplicate the new 2nd one, so I don't know if I'm missing something. |

Discussion:

Subject to any constraints in this subclause about permitted use of TXOP or SP according to the channel access mechanism used, the originator may

— **Separate the Block and Basic BlockAckReq frames into separate TXOPs or SPs**

**— Split a Block frame across multiple TXOPs or SPs**

— Split transmission of Data frames sent under block ack policy across multiple TXOPs or SPs

— Interleave MPDUs with different TIDs within the same TXOP or SP

— Sequence or interleave MPDUs for different RAs within a TXOP or SP

Yes, agree wth commenter that no idea what a Block frame is,. This is the only use of the term “Block frame”.

The clue may be earlier in the clause, line 1361.43,

“…if necessary, the originator may transmit a block of QoS Data frames separated by SIFS,…”

OR

It may mean “Block Ack”. We also have GCR block ack mentioned. Maybe the use of “the” and “a” are a clue?

Can we split a Block Ack frame? I don’t think so.

Let’s take the first bullet.

— **Separate the Block and Basic BlockAckReq frames into separate TXOPs or SPs**

Could it be

1. Separate the Block ACK and the Basic BlockAckReq frames into separate TXOPs or SPs

OR

1. Separate the block of QoS data frames and the Basic BlockAckReq frames into separate TXOPs or SPs

I think I am right that the BA should come SIFS after the Basic BlockAckReq, hence I think #2 is what is intended. This is different from bullet 3 and seems to make sense.

Second bullet:

**— Split a Block frame across multiple TXOPs or SPs**

Could it be

— Split a block of QoS data frames across multiple TXOPs or SPs

This does seem to be the same as bullet 3. There ain’t such a thing as a Block frame, so I am inclined to delete this.

Proposed resolution:

REVISED

1361.57 edit as shown

“Separate the block of QoS data frames and the Basic BlockAckReq frame into separate TXOPs or SPs”

At 1361.59 Delete

“— Split a Block frame across multiple TXOPs or SPs”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5145  Stephens, Adrian  9.23.3.3  1347.56 | "The MCCA enabled neighbor mesh STAs that could cause interference to transmissions during these reserved time periods, or that would experience interference from them, shall not transmit during these reserved time periods." How does a neighbor mesh STA know when it might cause interference? | Either precisely define "could cause interference" or "would experience interference", or turn the shall into some other verb. |

Discussion:

“MCCA enabled mesh STAs use Management frames to make reservations for transmissions. …. The MCCAOP owner and the MCCAOP responders advertise this MCCAOP reservation to their neighbors via an MCCAOP advertisement. The MCCA enabled neighbor mesh STAs that could cause interference to transmissions during these reserved time periods, or that would experience interference from them, shall not transmit during these reserved time periods.”

The intent of the text is clear, it referes to mesh STAs that would interfere if they transmitted, i.e. they are close enough that tif they transmitted at the same time as their neighbour, they would mutiually block each other. The underlying “shall” is that neighbour STAs shall not transmit during MCCAOPs between neighbors.

As the commenter found this confusing I suppose we should make it clearer.

Proposed resolution:

REVISED

At 1347.56 change

The MCCA enabled neighbor mesh STAs that could cause interference to transmissions during these reserved time periods, or that would experience interference from them, shall not transmit during these reserved time periods.”

MCCA enabled neighbor mesh STAs might cause interference to transmissions between the MCCAOP owner and the MCCAOP responders, an MCCA enabled neighbor mesh STA shall not transmit during these reserved time periods.

Consult Guido on this. If a STA hears an MCCAOP shall it obey?

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5135  Stephens, Adrian  9.16  1316.10 | "Further restrictions on TXVECTOR parameter values may apply due to rules found in 9.26 (Protection mechanisms) and 9.7 (Multirate support)."   What is this telling me? Is it specific to FEC\_CODING or not? | Either make this specific to FEC\_CODING or remove it. |

Discussion

**“9.16 LDPC operation**

An HT STA shall not transmit a frame with the TXVECTOR parameter FORMAT set to HT\_MF or HT\_GF and the TXVECTOR parameter FEC\_CODING set to LDPC\_CODING unless the RA of the frame corresponds to a STA for which the LDPC Coding Capability subfield of the HT Capabilities element received from that STA contained a value of 1 and dot11LDPCCodingOptionActivated is true.

Further restrictions on TXVECTOR parameter values may apply due to rules found in 9.26 (Protection mechanisms) and 9.7 (Multirate support).

So the first thing we see is that two TXVECTOR parameters, FORMAT and FEC\_CODING are referred to in the first paragraph. So why the commenter chose FEC\_CODING is not clear. Maybe he meant was “is it specific to LDPC”?

Does the cited sentence have anything to do with LDPC? YES

It refers to 9.26 and in **9.26.5.2**, 1393.48 we read:

“The first PPDU transmitted after a successful initial handshake (i.e., upon reception of a response frame with L-SIG TXOP protection addressed to the TXOP holder) shall have the TXVECTOR FORMAT parameter set to HT\_MF.”

Also:

**9.7.6.7 Control frame TXVECTOR parameter restrictions**

A STA shall not transmit a Control frame that initiates a TXOP with the TXVECTOR parameter FEC\_CODING set to a value of LDPC\_CODING.

Also, **9.7.6.5.5 Control response frame TXVECTOR parameter restrictions** (1298 61)

“A STA shall not transmit a control response frame with TXVECTOR parameter FEC\_CODING set to LDPC\_CODING unless it is in response to a reception of a frame with the RXVECTOR parameter FEC\_CODING equal to LDPC\_CODING..

A STA shall not transmit a control response frame with the TXVECTOR parameter FORMAT set to HT\_GF.” 1299.01

Proposed resolution:

REJECT

In 9.7.6.7, 9.7.6.5.5 and 9.26.5.2 there are direct references to TXVECTOR FORMAT and FEC\_CODING settings.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5134  Stephens, Adrian  9.15  1315.39 | "Otherwise, the STA is A-PPDU incapable." This is awkward.  There is no need to define "incapable". | Replace with "Otherwise, the STA is not A-PPDU capable." |

Discussion

“A DMG STA is aggregate PPDU (A-PPDU) capable if the A-PPDU supported field within the STA’s DMG Capabilities element is 1. Otherwise, the STA is A-PPDU incapable.

Proposed resolution:

ACCEPT

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5129  Stephens, Adrian  9.12  1310.19 | The use of "per" is often ambiguous. It can mean "through", or it can mean "for each".  In the case of: " agreement is determined per block ack agreement.", it could mean that agreement is determined for each block ack agreement, or through the process of block ack agreement.  The ambiguity exists when the thing following "per" can be viewed as a noun describing an object, or a noun describing a process. | Review all uses of "per" and replace ambiguous uses by "for each" or "through the process of" (or some less wordy unambiguous alternative). |

Discussion

“The use of an A-MSDU carried in a QoS Data frame under a block ack agreement is determined per block ack agreement.”

In this case it seems obvious that if ‘under a block ack agreement’, the process uses the ‘block ack agreement’. So how to say this better? I am tempted to say “as per” which is unambiguous and can not mean “each”. Then do we need “as per the process of”.

Or should it be:

“The use of an A-MSDU carried in a QoS Data frame under a block ack agreement is determined as per the corresponding block ack agreement.”

Looking for similar phrases I find (there are 621 instances or “per”)

964.02 “…then this field will be structured per the Vendor Specific element…”

1593.40 “…the frame is not discarded per management frame protection processing,”

1602.10 “…the frame is not discarded per management frame protection processing”

1603.11 “…the frame is not discarded per management frame protection processing,”

1713.36 “…shall be protected using the TPKSA, per the procedures defined in Clause 11 (Security).”

1737.27 “…transmit FTMs per Burst Fine Timing Measurement frames before…”

1999.24 “If the contents of the FTE are not as per specified for this message, then the TDLS responder STA shall reject the TDLS Setup Request frame with status code STATUS’ (This is a bit different)

2031.24 “Upon creation of the PMK, an AEK shall be created per 13.5.7 (Keys and key derivation algorithm…”

I think I will go with the ‘as per’ approach and then the committee loose to chime in with 101 variants.

Proposed changes:

REVISED

1310.19

From

“The use of an A-MSDU carried in a QoS Data frame under a block ack agreement is determined per block ack agreement.”

To

“The use of an A-MSDU carried in a QoS Data frame under a block ack agreement is determined for each block ack agreement.”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5128  Stephens, Adrian  9.12  1309.40 | "An A-MSDU contains only MSDUs whose DA parameter values map to a single RA value (see 8.3.2.2 (Aggregate MSDU (A-MSDU) format)). An A-MSDU contains only MSDUs whose SA parameter values map to a single TA value (see 8.3.2.2 (Aggregate MSDU (A-MSDU) format)). "  This text is "commoner" than the detail about PTP TSPECs and Short A-MSDUs that occurs before it. | Move cited text to the start of the subclause. Merge following sentence into previous para. |

Discussion

The commenter is right that the clause on A-MSDU starts with stuff on TIDs,. PTP TSPEC, PBSS and PCP and then Short A-MSDU. Then eventually we get to an important aspect. I agree with the commenter.

I think I would also move the next two paras as well.

Proposed resolution:

REVISED

Move Text at 1309.18 to 1309.35 to the end of the sub clause 1310.53.

Move text at 1309.35 to follow para beginning at 1309.40 and start new paragraph at “For the Short A-MSDU…”

Hence, now reads:

**“9.12 A-MSDU operation**

An A-MSDU contains only MSDUs whose DA parameter values map to a single RA value (see 8.3.2.2 (Aggregate MSDU (A-MSDU) format)). An A-MSDU contains only MSDUs whose SA parameter values map to a single TA value (see 8.3.2.2 (Aggregate MSDU (A-MSDU) format)).

For the Short A-MSDU case, an A-MSDU contains only MSDUs whose SA and DA parameter values are the same. The Short A-MSDU subframe structure is used only between a pair of STAs that communicate directly (see 8.3.2.1 (Format of Data frames)). The Short A-MSDU subframe structure cannot be used for frame forwarding.

The constituent MSDUs of an A-MSDU shall all have the same priority parameter value from the corresponding MA-UNITDATA.request primitive.

An A-MSDU shall be carried, without fragmentation, within a single QoS Data frame.”

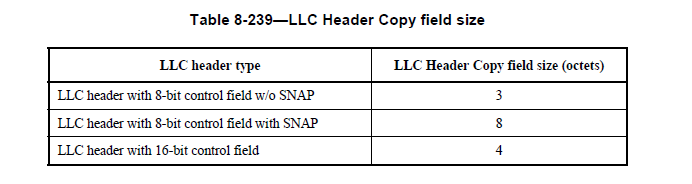
|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5127  Stephens, Adrian  9.11  1309.06 | "A STA that participates in a successful ADDTS exchange that included a U-PID element with the No-LLC field equal to 1 shall strip the LLC header from an MSDU corresponding to the TID indicated in the ADDTS exchange before transmission of the MSDU."  The MSDU is an opaque data type to the MAC. This is a statement for the operation of the MAC that implies the MAC knows how to strip an LLC header.  This breaks layering and assumptions made in the system architecture about the MAC. | Identify the architectural entity responsible for this behaviour. If this entity is within the scope of the 802.11 STA architecture, move this normative statement to the behavioural description of that entity.  If the entity is not within scope of the 802.11 architecture, ensure that the interface from the STA (UNITDATA) can indicate the No-LLC field so that higher layers can perform this function.  Likewise adjust the next sentence so that operations on MSDUs are not performed within the MAC. |

Discussion

**“9.11 MSDU processing**

A STA can use the U-PID element transmitted in ADDTS Request and ADDTS Response frames to indicate the protocol responsible for handling MSDUs corresponding to the TID indicated within the frame carrying the U-PID element (see 10.4.4.4 (TS setup procedures for both AP and non-AP STA initiation)). A STA that participates in a successful ADDTS exchange that included a U-PID element with the No-LLC field equal to 1 shall strip the LLC header from an MSDU corresponding to the TID indicated in the ADDTS exchange before transmission of the MSDU.A STA that participates in a successful ADDTS exchange that included a U-PID element with the No-LLC field equal to 1 and that receives an MSDU corresponding to the TID indicated in the ADDTS exchange shall add the header indicated by the U-PID element **before delivery of the MSDU at the MAC-SAP**.”

The U-PID element can be included in any variant of the ADDTS Request or Response. “No-LLC, is set to 1 to indicate MSDUs do not contain the LLC header.” Also includes LLC Header Copy,



We read in ADDTS Request and Response:

“When present in the ADDTS (Request) frame, the Upper Layer Protocol Identification (U-PID) element indicates the upper layer protocol associated with the TID/TSID specified within the TSPEC element contained in this frame. If a TSPEC element is not present in the frame, the U-PID element is not included inthe frame.”

Quick refresher on TID/TSID. It identifies the TC or TS to which the MSDU belongs. It is simply a value of 0-7 is EDCA and 8-15 for HCCA (HEMM SEMM).

So back to the cited clause. If an MSDU is received with “No-LLC” set to 1, then the STA adds in the LLC header as indicated in the LLC Copy field. So this is simply a way to avoid sending the LLC header over the air. So what about the transmitting? Where is the IP Datagram constructed? Does the MAC receive it already with the LLC header?

So if the “no-LLC field” option is set in the U-PID element of an ADDTS Request, the STA is supposed to strip off the LLC header, then transmit the MSDU, and then on reception, put it back in.  Hence, the datagram received from on high and communicated forward definitely does still have the LLC header.  If the commenter is right, and this is not a MAC function, then we need to say so and I suppose it becomes “outside the standard” .  My thought is that a compliant STA would know what this is and would carry out the stripping and adding in its driver, which is (is it not?) the MAC function?  In addition it definitely says (see the relevant para above) that the adding back in is done before the MAC-SAP.  Hence stating that the MAC does need to do this?

Reached out to Mark H.

Hi, Graham,

So, first, I will claim that this is an optimization, within the MAC, to save bytes on the air.  Normally, processing the LLC header would be outside the scope of the MAC, in that I agree with Adrian.  But, this is a special case of an operation which is transparent to both service users (the LLC sublayers on each end), and which is entirely something the MAC (under direction of the SME) does for its own optimization reasons.  The LLC header (the U-PID) is entirely opaque to the MAC, so I claim there is no layer violation going on here.

The comment asks to define the architectural entity that does this, and that seems like a reasonable request, so I wouldn’t reject the comment outright.  Of course, that said, we don’t have an entity identified that would fit this behavior.  The closest, I believe, is the A-MSDU aggregation/deaggregation facility, since the LLC header stripping/adding clearly has to be done before/after aggregating/deaggregating, respectively.  And, since this layer is “manipulating the MSDU” (in the sense of working with the bytes, even though they are opaque bytes), it makes sense to me to do the LLC header stripping/adding in this same spot.  Conveniently, 9.11 is just before 9.12, so this is already fine, from a text organization point-of-view (to address the part of the comment about moving the text – I don’t think we move it).  Do we add an “MSDU processing” “layer” to Figure 5-1, just above A-MSDU aggregation/deaggregation?  I suggest no for now, on the grounds that Figure 5-1 does not show every single processing element, only the major ones.  Let the commenter take this problem to ARC if he doesn’t like that answer.  (Dick’s thing will have the same problem, I note, as will TGak.)

I do note that the text in 9.11 is a bit broken.

First, it is vague about how and when the stripping happens, although it’s pretty good about the adding back in part.  But, for example, is the transmitting side (the stripping side) supposed to understand all formats of “LLC header” so that it “shall strip the LLC header from an MSDU”, or does it assume the first *n* bytes are the LLC header (where *n* was specified with the U-PID in the ADDTS.request)?  Is it supposed to compare the actual *n* bytes in the MSDU with what the SME told it would be there, in the ADDTS.request?  I’ll claim that for this to be a function done inside the MAC, and not be a layer violation, the MAC really shouldn’t understand all this stuff, and it should be just blindly doing what it was told.  (This will get me in trouble with Dick’s EDP<->LDP translation stuff.  But that stuff also needs some architectural arguments and sorting out, so I’ll worry about that later.)  And, I think “what it was told” is to strip *n* bytes (where *n* is known from the U-PID element in the ADDTS) without regard for the content or format of those bytes.  Then, at the receiver (as 9.11 already says) the supplied *n* bytes are blindly added in, and it’s someone else’s responsibility (not the MAC) to make sure this didn’t break anything.

Second, the text in 9.11 talks about what happens when the No-LLC field is set to 1.  Curiously, this field is defined in 8.4.2.153 to “indicate that MSDUs do not contain the LLC header.”  Is that, “do not contain \_*before they are given to the MAC SAP*\_” or “do not contain \_*when the bytes are sent OTA*\_”?  One assumes the bytes are there when the MSDU is given to the MAC SAP, and stripped before the MSDU actually flies over the air, otherwise why does 9.11 talk about the STA stripping the bytes?  But, this is not at all clear from the juxtaposition of 9.11 and 8.4.2.153.  I think it would be good to pick one, and make the text clear in both places.

Then, we should say something about what it means/what happens if the No-LLC field is set to 0.  I can’t find anything to help with this.   I guess you could suggest that the MAC shall bring world peace in this case.  (I’d go to an 11ad expert, and ask how the MAC accomplishes this. )

While you’re at it, I also note that 8.4.2.153 never says what the ‘format’ or content of the LLC Header Copy field is.  It only gives its size, dependent on what “LLC header type” it is.  (And, the latter is not indicated, so something has to figure that out – perhaps based on the Length of the element, but then it doesn’t (necessarily) match the actual content’s type/format.)  This is a bit of a mess.  I’d suggest that we say that the header bytes are opaque octets, and the length is understood from the Length of the element.  Then, Table 8-242 is really just an informative “NOTE” if you will, about common LLC formats that would have the mentioned lengths.  I’m not sure how we make a Table an informative NOTE, but I think it should be one.

So, for a comment that we basically Reject, there is quite a lot you could do to fix things in here.  Does that help?

Proposed resolution:

REJECT

But we need a cunning reason

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5126  Stephens, Adrian  9.1  1308.61 | "NOTE--A STA supporting the HT Control field can reset the NAV set by a wrapped RTS frame following the rules defined in 9.3.2.4 (Setting and resetting the NAV)."  Given the previous para, note is a statement of the obvious put there to "reassure" participants in the standards process as to what the previous para \*really really\* meant.  Such reassurance should be unnecessary if the previous para is written \*really really\* clearly enough. | Delete cited note. |

Discussion

**“9.10 Control Wrapper operation**

A STA supporting the HT Control field that receives a Control Wrapper frame shall process it as though it received a frame of the subtype of the wrapped frame.

NOTE—A STA supporting the HT Control field can reset the NAV set by a wrapped RTS frame following the rules defined in 9.3.2.4 (Setting and resetting the NAV).”

So, is the Note stating the obvious? Obviously it is to the commenter, but then he is probably cleverer than the average mere mortal who reads this Standard. I suppose that in this case I would say that the NOTE is much more an example than a note.

Two options come to mind:

A – Delete the NOTE (as proposed by commenter), i.e. ACCEPT

B – Change NOTE into an example i.e. REVISE

“A STA supporting the HT Control field that receives a Control Wrapper frame shall process it as though it received a frame of the subtype of the wrapped frame. For example, a STA supporting the HT Control field can reset the NAV set by a wrapped RTS frame following the rules defined in 9.3.2.4 (Setting and resetting the NAV).”

Straw Poll / Discussion

Proposed resolution:

Accept

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5125  Stephens, Adrian  9.9  1308.25 | "shall set the +HTC Support subfield " - there is no such subfield | Change "+HTC" to "+HTC-HT".  Make similar changes at 1397.41 |

Discussion

Agreed (see 883.44)

ACCEPT

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5124  Stephens, Adrian  9.8  1308.17 | "For MSDUs or A-MSDUs belonging to the service class of QoSNoAck when the receiver is a QoS STA"  Either there is a third condition to follow this one "service class of QoSNoAck and the receiver is not a QoS STA" or the constraint "when the receiver is a QoS STA" is unnecessary. | Either remove "when the receiver is a QoS STA", or add the missing third condition stated in the comment. |

Discussion

Here is complete paragraph. Note similar text at beginning (underlined)

**9.8 MSDU transmission restrictions**

“An A-MSDU shall contain only MSDUs of a single service class and inherits that service class for the purpose of the following rules. For MSDUs or A-MSDUs belonging to the service class of QoSAck when the receiver is a QoS STA, the QoS Data frames that are used to send these MSDUs or A-MSDUs shall have the Ack Policy subfield in the QoS Control field set to Normal Ack, Block Ack, Implicit Block Ack Request, or PSMP Ack. For MSDUs or A-MSDUs belonging to the service class of QoSNoAck when the receiver is a QoS STA, the QoS Data frames that are used to send these MSDUs or A-MSDUs shall have the Ack Policy subfield in the QoS Control field set to No Ack.”

It does seem superfluous to state that “when the receiver is a QoS STA”, as it may be obvious that “QoS Ack” and “QoSNoAck” apply only to QoS STAs. Let’s firstly confirm that QoSAck applies only to QoS STAs.

We read in 5.1.1.1 127.34

“There are two service classes available in a QoS STA: QoSAck and QoSNoAck.”

128.55

“In QoS STAs, the value of the service class parameter in the MAC service primitive (see 5.2 (MAC data service specification)) may be a noninteger value of QoSAck or QoSNoAck.”

However, I also read

129.19

“When a non-QoS Data frame is received from a STA, the service class parameter in the MA-UNITDATA.indication primitive is set to

— QoSAck, if the frame is an individually addressed frame and is acknowledged by the STA.

— QoSNoAck, if the frame is a group addressed frame and is not acknowledged by the STA.”

MA-UNITDATA.indication primitive defines the transfer of an MSDU from the MAC sublayer entity to the LLC sublayer entity, or entities in the case of group addresses.

Sooo…from this maybe we do need to still state “QoS STA” so as to be safe?

Having said all that, the sentence is awkward and as the commenter states, does seem to leave the reader hanging.

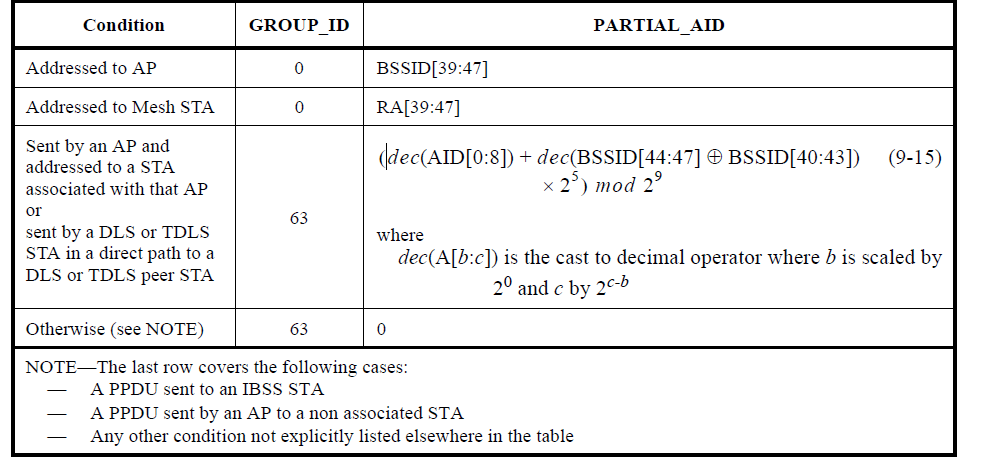
Proposed resolution:

REJECTED

See 129.4 which describes operation when the recipient is a non-QoS STA.

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5082  Stephens, Adrian  9.20  1318.24 | "dec(A[b:c]) is the cast to decimal operator" -- this is misleading terminology. There is nothing here special about base 10 arithmetic (decimal). | Replace "dec" with "value" in these equations and replace "is the cast to decimal" with "is the value". |

Discussion



“As an example of the GROUP\_ID and PARTIAL\_AID setting, consider the case of a BSS with BSSID 00-21-6A-AC-53-5232 that has as a member a non-AP STA assigned AID 5. In VHT PPDUs sent to an AP, the GROUP\_ID is set to 0 and the PARTIAL\_AID is set to 164. In VHT PPDUs sent by the AP to the non-AP STA associated with that AP, the GROUP\_ID is set to 63 and PARTIAL\_AID is set to 229.”

I have to admit that in carrying out the example to get the PARTIAL\_AID to be 229, I got totally confused at first because I assumed least significant is transmitted first, but then there is another reversal caused by the “where *dec*(A[b:c]) statement. So one simply takes the binaries as they are in order to do the calculation rather than the order they are transmitted in..

Doing that I was able to calculate the complicated formula result of 229, but then I failed miserably to get BSSID[39:47] to be 164 (I get 338). Maybe someone else could confirm or deny my answer?

53=0101 0011, 52=0101 0010. Hence last 9 bits [39:47] are 101010010 = 338 (in decimal)

BUT, back to the comment, as the commenter states, the *dec* defined as “cast to decimal operator” at first seems to be fancy way of saying, binary to decimal convertion, but it is not, because it inverts the order due to the “where” sentence.

*“dec*(A[*b:c*]) is the cast to decimal operator where *b* is scaled by 20 and *c* by 2c-b “

Hence, they have defined this special operator *dec* term.

For example, an AID of 0000101 is 5, but this is transmitted 1010000 which would be 160 but *dec*(AID[0:8]) is indeed 5. One could well object to the term “**the** cast to decimal operator” as it seems to be “**an** operator” invented for this.

NOW, is this confusing? Yes, it confused me for sometime.

They could easily have simply defined the bits differently on line 38 – 46 and saved all this trouble. Something like

“AID[*b*:*c*] represents bits *b* to *c* inclusive of the AID of the recipient STA with bit *b* being the more significant bit and *c* being the lesser significant bit.

BSSID[*b*:*c*] represents bits *b* to *c* inclusive of the BSSID, with bit *b* being the more significant bit and *c* being the lesser significant bit. In this representation, BSSID[47] is the least significant bit.

RA[*b*:*c*] represents bits *b* to *c* inclusive of the RA field, with bit *b* being the more significant bit and *c* being the lesser significant bit.. In this representation RA[47] is the least significant bit.

Then we can delete the “where” statement and the inversion.

So we have choice, if we do as little as possible we must keep the “where” statement in some form but is the term “cast to decimal operator” clear or not? The commenter suggest replacing *dec* with *value*. Not sure if that helps as *dec* is definitely a special sort of operator.

OPTIONS

A – Basically leave it as is but still confusing. Maybe change “cast to decimal operator” to

“A[*b:c*] represents the value of bits *b* to *c* of bit string A, in which bit position *b* is scaled by 20 and *c* by 2c-b  ”

B – Replace *dec* with *value. Value* is still a special operator so we don’t really make anything clearer. *Dec* is not so much a decimal conversion as an inverter and converter.

C - as per below

Alas, I feel that this will could a lot of discussion and time unless the others lose interest quickly. If one accepts, however, that for an AID of 5, AID[0:8] is a binary number of 00000101, then I think we are good.

Proposed resolution:

REVISED

1318.19 edit Table 9-9 to read

“((AID[0:8] + (BSSID[44:47] http://www.symbols.com/gi.php?type=1&id=1088&i=1 BSSID[40:43]) x 25) *mod* 29

where *A*[*b:c*] represents the value of bits *b* to *c* of bit string *A*, in which bit position *b* is scaled by 20 and *c* by 2c-b  ”

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5081  Stephens, Adrian  9.15  1315.61 | "NOTE--An A-PPDU capable DMG STA that receives an A-PPDU responds with an acknowledgment frame, if appropriate, only after it receives the last PPDU in the A-PPDU. "  -- this should not need to be stated, because it will only receive a frame requiring an immediate response in the last A-MPDU of the A-PPDU. It is therefore hinting at how to handle non-compliant behaviour, which is outside the scope of the standard. | Remove cited note. |

Discussion

One does wonder why this NOTE was added. What could have been the incentive? Why would a DMG STA want to transmit an ACK before it receives the last PPDU in the A-PPDU?

Maybe the clue is a couple of lines prior to the NOTE.

“If a frame that requires an immediate response is present within an A-PPDU, it shall be transmitted in the last A-MPDU of the A-PPDU.”

Even then, it is very clear.

Yes I agree with the commenter.

Proposed resolution:

ACCEPT

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5078  Stephens, Adrian  9.7.7.5  1302.20 | "BRP packets transmitted during beam refinement should use MCS 1 and shall not use any MCS greater than MCS 12. BRP packets transmitted during beam refinement should use MCS 0 if the BRP packet is sent at start of an SP as defined in 9.36.6.2 (Service period (SP) allocation)."  -- this creates conflicting recommendations | Reword thus: "A BRP packet transmitted during beam refinement at the start of a SP (as defined in 9.36.6.2 (Service period (SP) allocation)) should use MCS 0 , otherwise a BRP packet transmitted during beam refinement should use MCS 1. A BRP packet transmitted during beam refinement shall not use any MCS greater than MCS 12. " |

Discussion

Yep, can’t fault the comment logic.

Proposed resolution:

ACCEPT

|  |  |  |
| --- | --- | --- |
| Identifiers | Comment | Proposed change |
| CID 5046  Stephens, Adrian  9.24  1359.64 | "The delayed block ack mechanism is primarily intended to allow existing implementations" -- this may have been true at the time of writing the footnote in 2002, but it is surely no longer true. | Remove cited footnote. |

Discussion

“The block ack mechanism improves channel efficiency by aggregating several acknowledgments into one frame. There are two types of block ack mechanisms: immediate and delayed. Immediate block ack is suitable for high-bandwidth, low-latency traffic while the delayed block ack is suitable for applications that tolerate moderate latency.43

In delayed Block Ack the recient responds to a BlockAckReq with a standard Ack, then it sends its B,ockAck later (using highest priority) AC. Not to be encouraged.

The NOTE is definitely out of date, and serves no real purpose as for as the Standard is concerned

Proposed resolution:

ACCEPT