IEEE P802.11
Wireless LANs

|  |
| --- |
| **Comment Resolution for CID 9**  |
| **Date:** 2013-08-01 |
| **Author(s):** |
| **Name** | **Affiliation** | **Address** | **Phone** | **email** |
| Alfred Asterjadhi | Qualcomm Inc. | 5775 Morehouse Dr San Diego, CA 92109 | +1-858-658-5302 | aasterja@qti.qualcomm.com |

Abstract

This document provides comment resolution for TGah Draft 0.1 Comment Collection 9 with these CIDs: 9.

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 TGah Draft. This introduction is not part of the adopted material.

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

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **P.L** | **SC** | **Comment** | **Proposed Change** | **Resolution** |
| 9 | 135.135 | 9.21 | There is a BlockAck procedure but there is not a fragment ack procedure. 11ah supports sensors which need an efficient way of acknowledging fragments. | Define a Fragment Ack mechanisms to acknowledge multiple fragments with a single response frame with signaling similar to HT-Immediate BA. Will submit a document with the resolution. | Revised – TGah editor to make changes shown in 11-13- 0974-00-00ah under the heading for CID 9. |

Discussion: *The “Table 8-301c—Ack Policy field in the FC field for Short frames” which was motioned in IEEE July meeting (Reference doc 898r0) needs to be updated. Given that the changes related to that document are not reflected in D0.2, a new version of 898r1 has been uploaded to the server (and ready for pre-motion) where the only modification to the document that was motioned was to add the following text that enables fragment BA for short frames:*

*“In a short frame that is a fragment: When both the originator and the addressed recipient support the Fragment BA procedure, the addressed recipient returns an NDP BlockAck frame after a SIFS period, according to the procedure defined in 9.3.2.9a (Fragment BA procedure).”*

**Instruction to Editor: *Please add the following subclause immediately after subclause 9.3.2.9:***

**9.3.2.9a Fragment BA procedure**

A S1G STA may partition an MSDU or an MMPDU into multiple fragments as described in 9.5 (Fragmentation) and send the MPDUs resulting from the fragmentation of the MSDU or MMPDU as independent transmissions.

A S1G STA indicates support of Fragment BA using the Fragment BA Support subfield of the S1G Capabilities Info field in the S1G Capabilities element. A S1G STA shall set the Fragment BA Support subfield to 1 in S1G Capabilities element if the dot11FragmentBAOptionImplemented is true. Otherwise, the S1G STA shall set the Fragment BA Support subfield to 0. A S1G STA (known as the originator STA) with dot11FragmentBAOptionImplemented set to true sending frames to another S1G STA may use the Fragment BA procedure described in this section if it has received from the STA (known as the recipient STA) a frame that included a S1G Capabilities element with the Fragment BA Support subfield set to 1. Otherwise an S1G STA shall not use the Fragment BA procedure described in this section. Non-S1G STAs shall not use the Fragment BA procedure described in this section.

An originator STA may send MPDUs containing an MSDU or MMPDU fragment and set the Ack policy of the MPDU to Block Ack. A recipient STA shall not send any frame as an immediate response to an MPDU containing an MSDU or MMPDU fragment with the Ack Policy of the MPDU set to Block Ack. An originator STA may solicit an immediate response following the MDPU that includes a fragment by setting the ACK Policy to Implicit Block Ack Request. The receiving STA that is the intended receiver of an MPDU, containing a fragment, with the Ack Policy set to Implicit Block Ack Request shall send an NDP BlockAck frame after a SIFS time, without regard of the idle/busy state of the medium. The receiving STA shall include the receipt status of each fragment of the MSDU in the BlockAck Bitmap field of the NDP BlockAck frame.

The originator STA that elicits a NDP BlockAck (1MHz) as a response should request individual acknowledgements for each of the MSDUs with Fragment Number greater than 7. The originator STA shall not elicit NDP BlockAck 1MHz as a response for MSDUs with Fragment Number greater than 7.

Note: NDP BlockAck 1MHz can acknowledge only a limited number of consecutive fragments because its BlockAck Bitmap field size is 8. Instead, NDP BlockAck 2MHz can acknowledge up to the maximum number of fragments because its BlockAck Bitmap field size is 16.

The originator STA shall consider a NDP BlockAck (1 (or ≥2) MHz) frame as successfully received if the BlockAck ID field value equals the 2 (or 6) LSBs of the Scrambler Initialization value in the Service field and the Starting Sequence Control field value equals the Sequence Number of the fragment MPDU that elicited the response.

If the originator STA does not receive an NDP BlockAck frame as an immediate response, it may retransmit the last fragment MPDU to re-solicit an immediate NDP BlockAck response.

**9.2.7 Fragmentation/defragmentation overview**

**Instruction to Editor: *Please add the following Note immediately after Note 1 in subclause 9.2.7 (@REVmc D1.1):***

NOTE —A fragmented MSDU or MMPDU transmitted by an S1G STA to another S1G STA can be acknowledged either using immediate acknowledgment (i.e., transmission of an (NDP) ACK frame after a SIFS) or using the Fragment BA procedure described in (9.3.2.9a Fragment BA procedure).

**8.2.4.5.4 Ack Policy subfield**

**Instruction to Editor: *Please modify Table 8-10 as follows (@REVmc D1.1):***

|  |
| --- |
| * **Ack Policy subfield in QoS Control field of QoS (#100)Data frames**
 |
| **Bits in QoS Control field** | **Meaning** |
| **Bit 5** | **Bit 6** |
| 0 | 0 | Normal Ack or Implicit Block Ack Request. * ***Merge of .11ad change with CID 225 resolution.***

In a frame that is a non-A-MPDU frame:The addressed recipient returns an (#1198)Ack or QoS +CF-Ack frame after a short interframe space (SIFS) period, according to the procedures defined in 9.3.2.8 ((#1198)Ack procedure) and 9.20.3.5 (HCCA transfer rules). A non-DMG STA sets the Ack Policy subfield for individually addressed QoS Null (no data) frames(11ad) to this value.(#225) In a frame that is part of an A-MPDU:The addressed recipient returns a (#192)BlockAck frame, either individually or as part of an A-MPDU starting a SIFS after the PPDU carrying the frame, according to the procedures defined in 9.3.2.9 (Block Ack(Ed) procedure), 9.22.7.5 (Generation and transmission of BlockAck frames(#192) by an HT STA or DMG STA(11ad)), 9.22.8.3 (Operation of HT-delayed Block Ack), 9.26.4 (Rules for RD initiator), 9.26.5 (Rules for RD responder), and 9.30.3 (Explicit feedback beamforming).In a frame that is a fragment:When both the originator and the addressed recipient support the Fragment BA procedure, the addressed recipient returns an NDP BlockAck frame after a SIFS period, according to the procedure defined in 9.3.2.9a (Fragment BA procedure). |
| … |  |  |

* **NDP BlockAck**

**Instruction to Editor: *Please make the following changes to subclause 8.3.4a.1.5:***

NDP MAC frame body of NDP BlockAck frame contains the information listed in Table 8-33l (NDP MAC frame body of NDP BlockAck (1MHz)) and Table 8-33m (NDP MAC frame body of NDP BlockAck (≥2MHz)).

|  |
| --- |
| **Table 8-33l -- NDP MAC frame body of NDP BlockAck (1MHz)** |
| Field | Size (bits) | Description |
| NDP MACFrame Type | 3 | NDP MAC Frame Type field is set to 4. |
| BlockAck ID | TBD | The BlockAck ID field is TBD bits in length and contains the identifier of the NDP BlockAck frame. It is set to the TBD LSBs of the Scrambler Initialization value in the Service field (as defined in 24.3.9.2 (Service field)) of the PSDU that carries the soliciting frame.  |
| Starting Sequence Control | 12 | The Starting Sequence Control field is 12 bits in length and contains the sequence number of the first MSDU or A-MSDU for which the NDP BlockAck frame is sent. The value of this field is defined in 9.21.7.5 (Generation and transmission of BlockAck by an HT STA) when the NDP BlockAck is used during a BlockAck session and is set to the sequence number of the MSDU being fragmented when it is used during a Fragment BA session (see 9.3.2.9a (Fragment BA procedure)).  |
| BlockAck Bitmap | TBD | The BlockAck Bitmap field of the NDP BlockAck frame is TBD bits in length and is used to indicate the received status of up to TBD MSDUs and A-MSDUs when the NDP BlockAck is used during a BlockAck session. Each bit that is equal to 1 in the NDP BlockAck bitmap acknowledges the successful reception of a single MSDU or A-MSDU in the order of sequence number, with the first bit of the NDP BlockAck bitmap corresponding to the MSDU or A-MSDU with the sequence number that matches the value of the Starting Sequence Control field.When the NDP BlockAck is used during a Fragment BA session (see 9.3.2.9a (Fragment BA procedure)) each bit that is equal to 1 in the BlockAck Bitmap acknowledges the successful reception of a single fragment of an MSDU, in the order of the fragment number, with the first bit of the BlockAck Bitmap corresponding to the MPDU with fragment number equal to 0. |

The NDP MAC frame body of NDP BlockAck for >=2MHz has the structure defined in Table 8-33m (NDP MAC frame body of NDP BlockAck(≥2MHz)).

|  |
| --- |
| **Table 8-33m -- NDP MAC frame body of NDP BlockAck (≥2MHz)** |
| Field | Size (bits) | Description |
| NDP MACFrame Type | 3 | NDP MAC Frame Type field is set to 4. |
| BlockAck ID | TBD | The BlockAck ID field is TBD bits in length and contains the identifier of the NDP BlockAck frame. It is set to the TBD LSBs of the Scrambler Initialization value in the Service field (as defined in 24.3.9.2 (Service field)) of the PSDU that carries the soliciting frame. |
| Starting Sequence Control | 12 | The Starting Sequence Control field is 12 bits in length and contains the sequence number of the first MSDU or A-MSDU for which the NDP BlockAck frame is sent. The value of this field is defined in 9.21.7.5 (Generation and transmission of BlockAck by an HT STA) when the NDP BlockAck is used during a BlockAck session and is set to the sequence number of the MSDU being fragmented when it is used during a Fragment BA session (see 9.3.2.9a (Fragment BA procedure)).  |
| BlockAck Bitmap | TBD | The Block Ack Bitmap field of the NDP BlockAck frame is TBD bits in length and is used to indicate the received status of up to TBD MSDUs and A-MSDUs. Each bit that is equal to 1 in the NDP BlockAck bitmap acknowledges the successful reception of a single MSDU or A-MSDU in the order of sequence number, with the first bit of the NDP BlockAck bitmap corresponding to the MSDU or A-MSDU with the sequence number that matches the value of the Starting Sequence Control field. When the NDP BlockAck is used during a Fragment BA session (see 9.3.2.9a (Fragment BA procedure)) each bit that is equal to 1 in the BlockAck Bitmap acknowledges the successful reception of a single fragment of an MSDU, in the order of the fragment number, with the first bit of the BlockAck Bitmap corresponding to the MPDU with fragment number equal to 0. |

* **S1G Capabilities info field**

**Instruction to Editor: *Please change the following subclause as follows:***

The structure of the S1G Capabilities Info field is defined in Figure 8-401dg (S1G Capabilities Info field).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | B0 | B1 | B2 | B3 | B4 | B5 | B6 B7 | B8 |
|  | UplinkSynchCapable | DynamicAID | BATSupport | TIM ADESupport | Non-TIMSupport | TWTSupport | STATypeSupport | Fragment BA Support |
| Bits: | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| * **S1G Capabilities Info field**
 |

The subfields of the S1G Capabilities Info field are defined in Table 8-191d (Subfields of the S1G Capabilities Info field).

|  |
| --- |
| * **Subfields of the S1G Capabilities Info field**
 |
| Subfield | Definition | Encoding |
| … |  |  |
| Fragment BA Support | This bit indicates support of Fragment BA procedure | Set to 1 if dot11FragmentBAOptionImplemented is true.Set to 0 otherwise. |