IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LB1000 CID5962 Super BA | | | | |
| Date: 2015-09-01 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Matthew Fischer | Broadcom | 190 Mathilda Place, Sunnyvale, CA 94086 | +1 408 543 3370 | [mfischer@broadcom.com](mailto:mfischer@broadcom.com) |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Abstract

This document proposes a resolution for CID 5962 of LB1000 (the first sponsor ballot of the TGmc draft), a comment on TGm Draft 4.0 suggesting an increase in the maximum BA window from 64 to 256 MSDUs in order to allow an increase in efficiency when wide bandwidth, high constellation PPDUs are exchanged.

**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 TGmc Draft. This introduction is not part of the adopted material.

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

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

**CID LIST:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5962 | Matthew Fischer | 602.25 | 8.3.1.9.1 | With increasing PHY rates and PPDU BW values, the efficiency of medium utilization continues to drop unless PPDU durations can be maintained. PPDU durations at the higher PHY rates are limited by the maximum number of MPDUs that can be included in a single AMPDU which is in turn currently limited by the maximum BA window size. The maximum BA window size needs to be increased to allow medium efficiency to increase. | Update the BA mechanism to allow for a maximum BA window size of 256 MPDUs. This requires modification to the BA frame, the BA behavioral subclause and other areas. | Revise - generally agree with commenter, TGmc editor to execute proposed changes from 11-15-1024r0 found under all headings which include CID5962 |

**Discussion:**

As per the comment, increasing bandwidth of PPDUs, increasing constellation sizes and higher coding rates can combine to cause system-level inefficiencies during TXOPs even when the maximum count of 64 MSDUs are contained in a single AMPDU within a single PPDU. Hard-fought PHY-based throughput increases are lost when the combined overheads of PHY headers, IFS and control frame exchanges are significant when compared to the total amount of time spent for PHY payload symbol transmission within a single PPDU. By increasing the BA window, larger payloads can be placed within a single PPDU, thereby restoring efficiency to more respectable numbers for high-constellation/bandwidth scenarios. Combined AMSDU and AMPDU aggregation can alleviate some of the efficiency losses, but those returns are mitigated somewhat by the increase in PER that arises from higher bit counts per FCS. Still, it is expected that for the highest rates and efficiencies in the future, both a combined AMSDU/AMPDU aggregation policy and an increase in the BA window will be needed.

The term Extended is already used by DMG in the context of BlockAck (to allow for the addition of one byte to the BA frame to signal receiver buffer capacity), so the modifier “super” has been chosen for this extension. Within DMG it is possible to have a BlockAck or BlockAckReq frame that is both Super and Extended.

Note that there are 8 reserved bits available in the BAR and BA control fields.

The determination of which type of BlockAck frame to transmit following the receipt of an AMPDU is based on examination of the TA of the AMPDU. I.e. there is no explicit signalling for which type of BlockAck to use.

**Proposed changes**

The document of reference for baseline text is REVmc Draft 4.2.

**CID 5962**

**8.3.1.8.1 Overview**

***TGmc editor: in Figure 8-26 BAR Control field found within subclause 8.3.1.8.1 Overview, replace the reserved bit in position B11 of the BAR Control field with a new bit Super Bitmap and modify some of the text within the subclause and modify Table 8-22 BlockAckReq frame variant encoding as shown:***

The values of the Multi-TID, Compressed Bitmap, Super Bitmap and GCR subfields determine which of seven possible BlockAckReq frame variants is represented, as indicated in Table 8-22 (BlockAckReq frame variant encoding).

**Table 8-22 – BlockAckReq frame variant encoding**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Multi-TID subfield value** | **Compressed Bitmap subfield value** | **GCR subfield value** | **Super Bitmap subfield value** | **BlockAckReq frame variant** |
| 0 | 0 | 0 | 0 | Basic BlockAckReq |
| 0 | 1 | 0 | 0 | Compressed BlockAckReq |
| 1 | 0 | 0 | 0 | Extended Compressed BlockAckReq |
| 1 | 1 | 0 | 0 | Multi-TID BlockAckReq |
| 0 | 0 | 1 | 0 | Reserved |
| 0 | 1 | 1 | 0 | GCR BlockAckReq |
| 1 | 0 | 1 | 0 | Reserved |
| 1 | 1 | 1 | 0 | Reserved |
| 0 | 0 | 0 | 1 | Reserved |
| 0 | 1 | 0 | 1 | Super BlockAckReq |
| 1 | 0 | 0 | 1 | Reserved |
| 1 | 1 | 0 | 1 | Extended Super BlockAckReq |
| 0 | 0 | 1 | 1 | Reserved |
| 0 | 1 | 1 | 1 | Reserved |
| 1 | 0 | 1 | 1 | Reserved |
| 1 | 1 | 1 | 1 | Reserved |

**8.3.1.9.1 Overview**

***TGmc editor: in Figure 8-32 BA Control field found within subclause 8.3.1.9.1 Overview, replace the reserved bit in position B11 of the BA Control field with a new bit Super Bitmap and modify some of the text within the subclause and modify Table 8-24 BlockAck frame variant encoding as shown:***

The values of the Multi-TID, Compressed Bitmap, Super Bitmap and GCR subfields of the BA Control field determine which of the BlockAck frame variants is represented, as indicated in Table 8-24 (BlockAck frame variant encoding).

**Table 8-24 – BlockAck frame variant encoding**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Multi-TID subfield value** | **Compressed Bitmap subfield value** | **GCR subfield value** | **Super Bitmap subfield value** | **BlockAck frame variant** |
| 0 | 0 | 0 | 0 | Basic BlockAck |
| 0 | 1 | 0 | 0 | Compressed BlockAck |
| 1 | 0 | 0 | 0 | Extended Compressed BlockAck |
| 1 | 1 | 0 | 0 | Multi-TID BlockAck |
| 0 | 0 | 1 | 0 | Reserved |
| 0 | 1 | 1 | 0 | GCR BlockAck |
| 1 | 0 | 1 | 0 | Reserved |
| 1 | 1 | 1 | 0 | Reserved |
| 0 | 0 | 0 | 1 | Reserved |
| 0 | 1 | 0 | 1 | Super BlockAck |
| 1 | 0 | 0 | 1 | Reserved |
| 1 | 1 | 0 | 1 | Extended Super BlockAck |
| 0 | 0 | 1 | 1 | Reserved |
| 0 | 1 | 1 | 1 | Reserved |
| 1 | 0 | 1 | 1 | Reserved |
| 1 | 1 | 1 | 1 | Reserved |

***TGmc editor: add two new subclauses after subclause 8.3.1.9.6 GCR Block Ack variant as shown:***

**8.3.1.9.6a Super BlockAck variant**

The TID\_INFO subfield of the BA Control field of the Super BlockAck frame contains the TID for which this BlockAck frame is sent.

The BA Information field of the Super BlockAck frame comprises the Block Ack Starting Sequence Control subfield and the Block Ack Bitmap subfield, as shown in Figure 8-37a (BA Information field (Super BlockAck)). The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield contains the sequence number of the first MSDU or A-MSDU for which this BlockAck frame is sent. The value of this subfield is defined in 9.24.7.5 (Generation and transmission of BlockAck frames by an HT STA or DMG STA). The Fragment Number subfield of the Block Ack Starting Sequence Control subfield is set to 0.

The Block Ack Bitmap subfield of the BA Information field of the Super BlockAck frame is 32 octets in length and is used to indicate the received status of up to 256 entries, where each entry represents an MSDU or an A-MSDU. Each bit that is equal to 1 in the super Block Ack Bitmap field acknowledges the successful reception of a single MSDU or A-MSDU in the order of sequence number, with the first bit of the Block Ack Bitmap field corresponding to the MSDU or A-MSDU with the sequence number that matches the value of the Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield.

|  |  |  |
| --- | --- | --- |
|  | Block Ack Starting Sequence Control | Block Ack Bitmap |
| Octets: | 2 | 32 |

**Figure 8-37a BA Information field (Super BlockAck)**

***TGmc editor: add a new subclause after subclause 8.3.1.9.6 GCR Block Ack variant as shown:***

**8.3.1.9.6b Extended Super BlockAck variant**

The TID\_INFO subfield of the BA Control field of the Extended Super BlockAck frame contains the TID for which this BlockAck frame is sent.

The BA Information field of the Extended Super BlockAck frame comprises the Block Ack Starting Sequence Control subfield and the Block Ack Bitmap subfield, as shown in Figure 8-37b (BA Information field (Extended Super BlockAck)). The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield contains the sequence number of the first MSDU or A-MSDU for which this BlockAck frame is sent. The value of this subfield is defined in 9.24.7.5 (Generation and transmission of BlockAck frames by an HT STA or DMG STA). The Fragment Number subfield of the Block Ack Starting Sequence Control subfield is set to 0.

The Block Ack Bitmap subfield of the BA Information field of the Extended Super BlockAck frame is 32 octets in length and is used to indicate the received status of up to 256 entries, where each entry represents an MSDU or an A-MSDU. Each bit that is equal to 1 in the super Block Ack Bitmap field acknowledges the successful reception of a single MSDU or A-MSDU in the order of sequence number, with the first bit of the Block Ack Bitmap field corresponding to the MSDU or A-MSDU with the sequence number that matches the value of the Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Block Ack Starting Sequence Control | Block Ack Bitmap | RBUFCAP |
| Octets: | 2 | 32 | 1 |

**Figure 8-37b BA Information field (Extended Super BlockAck)**

The RBUFCAP field contains an unsigned integer that is the number of MPDU buffers available to store received MPDUs at the time of transmission of the Extended Super BlockAck frame (9.39 (DMG link adaptation)).

**8.4.2.138 ADDBA Extension element**

***TGmc editor: within subclause 8.4.2.138 ADDBA Extension element replace one of the seven reserved bits of the ADDBA Capabilities field in Figure 8-527 – ADDBA Capabilities field format with a new subfield named “Super Bitmap”, and add a new paragraph to the subclause as shown:***

The Super Bitmap subfield is used during the set up of a Block Ack agreement to negotiate the use of the Super Bitmap as found in the Super BlockAck and Extended Super BlockAck. When this subfield is set to 1 in the ADDBA Request frame, it indicates that the originator is requesting the use of Super BlockAck or Extended Super BlockAck variant. When this subfield is set to 1 in the ADDBA Response frame, it indicates that the recipient is accepting the use of the Super BlockAck or Extended Super BlockAck variant.

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

***TGmc editor: within subclause 9.24.2 Setup and modification of the block ack parameters, modify some of the text as shown:***

When the Block Ack Policy subfield value is set to 1 by the originator of an ADDBA Request frame between HT STAs, then the ADDBA Response frame accepting the ADDBA Request frame shall contain 1 in the Block Ack Policy subfield.

An HT STA or DMG STA with dot11SuperBlockAckOptionImplemented equal to true may set the Super Bitmap subfield of the ADDBA Extension element in an ADDBA Request frame to 1. An HT STA or DMG STA with dot11SuperBlockAckOptionImplemented equal to true may set the Super Bitmap subfield of the ADDBA Extension element in an ADDBA Response frame to 1 if the Super Bitmap subfield of the ADDBA Request frame to which this frame is a response was equal to 1.

For each accepted block ack agreement, the originator shall set the sequence number of the first frame transmitted under the agreement to the value of the Block Ack Starting Sequence Control field of the ADDBA Request frame of the accepted block ack agreement.

When a block ack agreement is established between two HT STAs or two DMG STAs, the originator may change the size of its transmission window if the value in the Buffer Size field of the ADDBA Response frame is larger than the value in the ADDBA Request frame. If the value in the Buffer Size field 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) so that it is not greater than the value in the Buffer Size field of the ADDBA Response frame and is not greater than the value *BitmapSize*, where *BitmapSize* is equal to 256 when the Super Bitmap subfield of the ADDBA Response frame is equal to 1 and 64, otherwise.

**9.24.6 Selection of BlockAck and BlockAckReq variants**

***TGmc editor: within subclause 9.24.6 Selection of BlockAck and BlockAckReq variants, add a new paragraph at the end of the subclause as shown:***

The Super Bitmap subfield of the BA Control field or BAR Control field may be set to 1 in BlockAck and BlockAckReq frames sent as part of a BA agreement for which the ADDBA Response frame contained a value of 1 in the Super Bitmap subfield of the ADDBA Extension element.

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

***TGmc editor: within subclause 9.24.7.3 modify the text as shown:***

For each HT-immediate block ack agreement that uses full-state operation, a recipient shall maintain a block acknowledgment record as defined in 9.24.3 (Data and acknowledgment transfer using immediate block ack policy and delayed block ack policy). This record includes a bitmap, indexed by sequence number; a 12-bit unsigned integer starting sequence number, *WinStartR*, representing the lowest sequence number position in the bitmap; a variable *WinEndR*; and the maximum transmission window size, *WinSizeR*, which is set to the smaller of *BitmapSize*and the value of the Buffer Size field of the associated ADDBA Response frame that established the block ack agreement, where *BitmapSize* is equal to 256 when the Super Bitmap subfield of the ADDBA Response frame is equal to 1 and 64, otherwise. *WinEndR* is defined as the highest sequence number in the current transmission window. A STA implementing full-state operation for an HT-immediate block ack agreement shall maintain the block acknowledgment record for that agreement according to the following rules:

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

***TGmc editor: within subclause 9.24.7.4 modify the text as shown:***

For an HT-immediate block ack agreement that uses partial-state operation, a recipient shall maintain a temporary block acknowledgment record as defined in 9.24.3 (Data and acknowledgment transfer using immediate block ack policy and delayed block ack policy). This temporary record includes a bitmap, indexed by sequence number; a 12-bit unsigned integer WinStartR (the lowest sequence number represented in the bitmap); a 12-bit unsigned integer WinEndR (the highest sequence number in the bitmap); the originator address; TID; and the maximum transmission window size, WinSizeR, which is set to the smaller of *BitmapSize*and the value of the Buffer Size field of the associated ADDBA Response frame that established the block ack agreement, where *BitmapSize* is equal to 256 when the Super Bitmap subfield of the ADDBA Response frame is equal to 1 and 64, otherwise.

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

***TGmc editor: within subclause 9.24.7.5 Generation and transmission of BlockAckc frames by an HT STA or DMG STA, modify the text as shown:***

When responding with a BlockAck frame to either a received BlockAckReq frame or a received A-MPDU with Ack Policy equal to Normal Ack (i.e., implicit block ack request) during either full-state operation or partial-state operation, any adjustment to the value of *WinStartR* according to the procedures defined within 9.24.7.3 (Scoreboard context control during full-state operation) and 9.24.7.4 (Scoreboard context control during partial-state operation) shall be performed before the generation and transmission of the response BlockAck frame. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield of the BlockAck frame shall be set to any value in the range from (*WinEndR* – *BitmapSizeMinus1*) to *WinStartR*, where *BitmapSizeMinus1* is equal to 255 when the Super Bitmap subfield is equal to 1 and 63, otherwise. The values in the recipient’s record of status of MPDUs beginning with the MPDU for which the Sequence Number subfield value is equal to *WinStartR* and ending with the MPDU for which the Sequence Number subfield value is equal to *WinEndR* shall be included in the bitmap of the BlockAck frame.

When responding with a BlockAck frame to either a received BlockAckReq frame or a received A-MPDU with Ack Policy equal to Normal Ack (i.e., implicit block ack request) during either full-state or partial-state operation, if the adjusted value of *WinEndR* is less than the value of the starting sequence number of the BlockAck frame plus *BitmapSizeMinus1*, within the bitmap of the BlockAck frame, the status of MPDUs with sequence numbers that are greater than the adjusted value of *WinEndR* shall be reported as unsuccessfully received (i.e., the corresponding bit in the bitmap shall be set to 0). *BitmapSizeMinus1* is equal to 255 when the Super Bitmap subfield is equal to 1 and 63, otherwise.

**9.24.7.6.1 General**

***TGmc editor: within subclause 9.24.7.6.1 General, modify the text as shown:***

*WinEndB* is initialized to *WinStartB* + *WinSizeB* – 1, where *WinSizeB* is set to the smaller of *BitmapSize*and the value of the Buffer Size field of the ADDBA Response frame that established the block ack agreement, where *BitmapSize* is equal to 256 when the Super Bitmap subfield of the ADDBA Response frame is equal to 1 and 64, otherwise.

**10.5.2.4 Procedure common to both originator and recipient**

***TGmc editor: within subclause 10.5.2.4 Procedure common to both originator and recipient, modify Table 1-6 – Types of block ack agreement based on capabilities and ADDBA conditions for DMG STAs as shown:***

**Table 10-6—Types of block ack agreement based on capabilities and ADDBA conditions for DMG STAs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Capabilities condition** | **ADDBA condition** | **Type of BlockAckReq and BlockAck variant** | **Typeof block ack agreement** |
| Both STAs have the BlockAck with Flow Control field in the DMG Capabilities element equal to 1. | Block Ack Policy subfield set to 0, Super Bitmap subfield set to 0 | Compressed | HT-immediate |
| Both STAs have the BlockAck with Flow Control field in the DMG Capabilities element equal to 1. | Block Ack Policy subfield set to 0, Super Bitmap subfield set to 1 | Super | HT-immediate |
| At least one of the STAs has the BlockAck with Flow Control field in the DMG Capabilities element equal to 0. | Block Ack Policy subfield set to 0, Super Bitmap subfield set to 0 | Compressed | HT-immediate |
| At least one of the STAs has the BlockAck with Flow Control field in the DMG Capabilities element equal to 0. | Block Ack Policy subfield set to 0, Super Bitmap subfield set to 1 | Super | HT-immediate |
| At least one of the STAs has the BlockAck with Flow Control field in the DMG Capabilities element equal to 0. | Block Ack Policy subfield set to 1, Super Bitmap subfield set to 0 | Extended Compressed | HT-immediate |
| At least one of the STAs has the BlockAck with Flow Control field in the DMG Capabilities element equal to 0. | Block Ack Policy subfield set to 1, Super Bitmap subfield set to 1 | Extended Super | HT-immediate |
| Both STAs have the BlockAck with Flow Control field in the DMG Capabilities element equal to 1. | Block Ack Policy subfield set to 1, Super Bitmap subfield set to 0 | Extended Compressed | HT-immediate + flow control |
| Both STAs have the BlockAck with Flow Control field in the DMG Capabilities element equal to 1. | Block Ack Policy subfield set to 1, Super Bitmap subfield set to 1 | Extended Super | HT-immediate + flow control |

***TGmc editor: add the following new MIB variable to the dot11StationConfig group and add a corresponding value in the group’s SEQUENCE definition and add an appropriate entry to the dot11VHTMACAdditions Object-group and an entry to the dot11SMTbase13 group:***

**C.3 MIB Detail**

dot11SuperBlockAckOptionImplemented 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 IEEE 802.11 Super BlockAck option is implemented."

DEFVAL { false }

::= { dot11StationConfigEntry <ANA> }

**References:**