IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Receive Buffer Operations CID 10031 10032 | | | | |
| Date: 2011-01-17 | | | | |
| 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

Addressing receive buffer operations to resolve CID 10031 and 10032 of TGmb sponsor ballot. Proposed edits are based on REVmb D6.04.

**CID 10031, 10032 proposed changes in letter ballot comment:**

Change the cache-filling instructions such that the first cache mentioned is only used for non-QOS receptions by any STA and the second cache is only used for QOS receptions by QoS STAs. E.g. "The receiving STA shall keep a cache of recently received <Address 2, sequence-number, fragment-number> tuples." becomes something like this: "The receiving STA shall keep a cache of recently received <Address 2, sequence-number, fragment-number> tuples from non-QoS MSDUs." In order to make the rejection text work properly, both caches are going to need a label as well. I suggest non-QoS duplicate cache and QoS duplicate cache and then use the appropriate reference wherever the noun cache is currently employed in this subclause.

I believe that the paragraph should read as follows: "A STA shall reject as a duplicate frame any non-QoS frame in which the Retry bit in the Frame Control field is 1 and that matches an <Address 2, sequence-number, fragment-number> tuple of an entry in the non-QoS cache. A receiver QoS STA shall also reject as a duplicate frame any QoS frame in which the Retry bit in the Frame Control field is 1 and that matches an <Address 2, TID, sequence-number, fragment-number> tuple of an entry in the QoS cache."

**CID 10031, 10032 proposed changes in this document:**

The suggested change is not really what was needed. The separate caches already exist, and the description of the caches and duplicate discarding are correct as they stand, but incomplete with respect to operation in the context of a block ack agreement. The proposed change modifies the block ack text to show how additional frames beyond those identified by the cache are discarded as either duplicate or old.

***TGmb editor: change the first paragraph of 9.3.2.11 as shown:***

**9.3.2.11 Duplicate detection and recovery(#1606)**

Because MAC-level acknowledgments and retransmissions are incorporated into the protocol, there is the

possibility that a frame may be received more than once. The procedures defined in this subclause attempt to

filter out these duplicates(#1630). Additional duplicate filtering is performed during Receive Buffer Operation for frames that are part of a Block Ack agreement as described in 9.20.4 and 9.20.7.

***TGmb editor: change 9.20.4 as shown:***

**9.20.4 Receive buffer operation**

For each Block Ack agreement, the recipient maintains a MAC variable NextExpectedSequenceNumber. The NextExpectedSequenceNumber is initialized to zero when a Block Ack agreement is accepted.

Upon the receipt of a QoS data frame from the originator for which a Block Ack agreement exists, the

recipient buffers(#10023) the MSDU regardless of the value of the Ack Policy subfield within the QoS Control

field of the QoS data frame, unless the sequence number of the frame is older than the NextExpectedSequenceNumber for that Block Ack agreement, in which case the frame is discarded because it is either old or a duplicate.

The recipient flushes received MSDUs from its receive buffer as described in this subclause.

If a BlockAckReq frame is received, all complete MSDUs and A-MSDUs(11n) with lower sequence numbers

than the starting sequence number contained in the BlockAckReq frame shall be passed up to the next MAC

process as shown in Figure 6-1.(11n) Upon arrival of a BlockAckReq frame, the recipient shall pass up the

MSDUs and A-MSDUs(11n) starting with the starting sequence number sequentially until there is an

incomplete or missing(#10) MSDU or A-MSDU(11n) in the buffer. If no MSDUs or A-MSDUs are passed up to the next MAC process after the receipt of the BlockAckReq frame and the starting sequence number of the BlockAckReq frame is newer than the NextExpectedSequenceNumber for that Block Ack agreement, then the NextExpectedSequenceNumber for that Block Ack agreement is set to the sequence number of the BlockAckReq frame.

If, after an MPDU is received, the receive buffer is full, the complete MSDU or A-MSDU(11n) with the

earliest sequence number shall be passed up to the next MAC process.(11n)

If, after an MPDU is received, the receive buffer is not full, but the sequence number of the complete MSDU or A-MSDU(11n) in the buffer with the lowest sequence number is equal to the NextExpectedSequenceNumber for that Block Ack agreement, then the MPDU shall be passed up to the next MAC process.

Each time that the recipient passes an MSDU or A-MSDU for a Block Ack agreement up to the next MAC process, the NextExpectedSequenceNumber for that Block Ack agreement is set to the sequence number of the MSDU or A-MSDU that was passed up to the next MAC process plus one.

The recipient shall(#1478) pass MSDUs and A-MSDUs up to the next MAC process(11n) in order of

increasing sequence number.

**9.20.7.6.2 Operation for each received data MPDU(11n)**

***TGmb editor: Under item a), change the text as shown:***

1) Store the received MPDU in the buffer, if no MSDU with the same sequence number is already

Present, otherwise discard the MPDU.(#10023)

***TGmb editor: Under item b), change the text as shown:***

1) Store the received MPDU in the buffer, if no MSDU with the same sequence number is already

Present, otherwise discard the MPDU.(#10023)

**References:**