Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**ARQ for TG4 Systems**
Date Submitted	**2000-05-10**
Source(s)	Subbu Ponnuswamy Malibu Networks 1035 Suncast Lane El Dorado Hills, CA 95762 Voice: 916-941-8815 Fax: 916-941-8850 mailto:subbu@malibunetworks.com Jacob Jorgensen Malibu Networks 1035 Suncast Lane El Dorado Hills, CA 95762 Voice: 916-941-8810 Fax: 916-941-8850 mailto:jacob@malibunetworks.com Juha Salokannel Nokia Visiokatu 1 Tampere, FINLAND, FIN-33720 Voice: +358 3 272 5494 Fax: mailto: juha.salokannel@nokia.com Mika Kasslin Nokia Itamerenkatu 11-13 Helsinki, FINLAND, FIN-00180 Voice: +358 71 8036294 Fax: +358 71 8036856 mailto: mika.kasslin@nokia.com Chet Shirali Vyyo Voice: 408-330-0300 Ext 22 Fax: mailto: cshirali@vyyo.com
Re:	In response to “Call for Comments on 802.16 TG4 Straw-man Document 802.16.4-01/10”.
Abstract	In this contribution, we propose a variation of Selective Repeat ARQ for TG4 systems and describe specific changes to the 802.16 TG4 (straw man).
Purpose	Approve portions of this document described in section “Specific comments” as additions to IEEE 802.16 TG4 (straw man).
Notice	
Release	

Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures (Version 1.0) <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:r.b.marks@ieee.org> as early as possible, in written or electronic form, of any patents (granted or under application) that may cover technology that is under consideration by or has been approved by IEEE 802.16. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.

ARQ for TG4 Systems

Subbu Ponnuswamy and Jacob Jorgensen
Malibu Networks
Juha Salokannel and Mika Kasslin
Nokia
Chet Shirali
Vyyo

Introduction

In this submission, we propose an ARQ mechanism for TG4 MAC. The additions to the 802.16 TG4 (straw man) document including the formats of new messages, are describeds. Our proposal includes a variation of the Selective Repeat ARQ that has been shown to outperform other well-known ARQ schemes in terms of higher channel efficiency and low mean delay. The proposed approach not only optimizes the path from transmitter to the receiver by selectively retransmitting only the lost MPDUs, but also provides an efficient and scalable bitmap-based ACK management mechanism to optimize the ACK traffic flowing in the reverse direction. Similar bitmap-based ACK management for Selective Repeat schemes have been successfully used by many point-to-multipoint wireless systems, including HIPERLAN/2 [2], GPRS [4] and Wireless ATM [8]. Since the link conditions could change at any time, a block level retransmission approach using MPDU-Sequence Numbers (MSN) is presented. The block level retransmission scheme allows the MPDU size be changed between retransmissions. However, a system that does not require support for changing MPDU sizes between retransmissions incurs no additional overhead. The default retransmission mechanism is based on simple MPDU sequence numbering and MPDU-level retransmission.

In summary:

- Each MPDU is given a sequence number. This sequence number is used for re-assembly as well as ARQ
- Selective Repeat ARQ algorithm is used to selectively retransmit only lost MPDUs
- Block-level retransmission allows the MPDU size be changed between retransmissions, if necessary
- Cumulative ACK and bitmap-based ACK are used to manage ACK traffic
- ARQ ACKs are allowed to piggyback with data as well as use a separate ACK connection
- ARQ works efficiently with various combinations of optional features such as fragmentation and packing
- Incorporating ARQ into the standard does not increase overhead for connections that do not want to use ARQ

Acronyms

The following acronyms are used in this document:

ACK	Acknowledgement
ACKC	ACK Congestion
ARQ	Automatic Repeat reQuest
BBN	Bitmap Block Number
BM	BitMap
BSN	Block Sequence Number
CACK	Cumulative Acknowledgement
MPDU	MAC Protocol Data Unit
MSN	MPDU-Sequence Number
MSDU	MAC Service Data Unit (Higher Layer Packet)
PSH	Packing Sub-Header
SR	Selective Repeat

Summary

In this section we present the summary of the proposed ARQ approach, retransmission units and message formats for ARQ. The following sections describe specific text to be added to the 802.16 TG4 straw man [11].

Sequence numbering and Fragmentation

The 802.16 MAC [1] defines two fields, Fragmentation Control (FC) and Fragment Sequence Number (FSN) to control fragmentation and re-assembly. These fields are part of optional sub-headers in 802.16.1. The 3-bit FSN not only restricts the number of allowed fragments per-MSDU to eight, but also creates ambiguity in re-assembly, when a set of consecutive MPDUs is lost. Therefore, we propose that an MSN be used instead of the FSN to unambiguously re-assemble MPDUs for TG4 systems. Since the MSN is not a per-MSDU sequence number, gaps in MSN can be easily detected to re-assemble correctly received MSDUs and to discard MSDUs with missing fragments. Since the Fragmentation sub-header (FC and FSN) requires at least one additional byte to the generic MAC header, the per-MSDU FSN + FC of TG1 optional sub-header is replaced by the MSN (6 bits) + FC (2 bits), without any additional overhead, for TG4. This does not require any changes to the baseline 802.16 MAC, as the fragmentation sub-headers are optional. With the help of 2-bit FC fields and MSN, the receiver unambiguously determines the first and last fragments of MSDUs and missing MSDUs respectively. The MSN combined with FC, enables low overhead re-assembly and ARQ mechanisms for TG4 systems.

MSN and Retransmission Unit

Since the ARQ is implemented at the MAC layer, the natural choice for the retransmission unit is the MPDU. The MPDUs are given a sequence number when they are created [9]. The sequence numbers are based on the number of blocks that go into an MPDU. The default block size is dynamically set to the MPDU size for every MPDU. The default mechanism is a simple MPDU sequence numbering scheme, where each MPDU is given a sequence number, irrespective of the number of bytes in the MPDU. Since the smallest unit of retransmission is the block, the default scheme does not allow the MPDU size be changed between retransmissions. The section on Block Sequence Numbering describes the details of the block concept.

The receivers and transmitters use MSN to track and retransmit lost MPDUs and to advance their receiving and sending windows based on positive acknowledgements. The MSN-based ARQ makes no assumptions about any optional features of 802.16 MAC such as packing and fragmentation. The MSN will be assigned to all MPDUs irrespective of whether ARQ is enabled or not. When fragmentation is allowed for a connection, the MSN is used to re-assemble packets, even if ARQ is not enabled.

This also allows us to implement ARQ (with small window size) with no additional overhead than what is required for fragmentation control. The same sub-header that is used for re-assembly is used by the ARQ protocol. If bigger window size is desired, an additional byte is added to the sub-header. The TYPE field in 802.16 Generic MAC Header is used to control the different types of sub-headers present.

Packing Sub-header

Another optional feature of 802.16 MAC that can be used with ARQ and/or fragmentation is packing. When packing is allowed for a connection, a single MPDU may have one or more full or fragmented MSDUs. Since only the MPDUs are given a sequence number, no additional overhead is introduced for such packed MPDUs. 802.16 MAC defines an optional packing sub-header similar to the one shown in **Figure 2**. The only difference between this and the one in the baseline 802.16.1 is the TYPE field. Since FSN is not needed for TG4, the three FSN bits are used for the TYPE field in **Figure 2**. As shown in a later section, the TYPE field is used to piggyback ARQ messages (ACK, Discard and Reset Messages).

Selective Repeat ARQ

Selective Repeat (SR) has been proposed for many wireless and mobile standards such as HIPERLAN/2, IMT-2000. Since wireless resources are scarce and SR provides higher channel efficiency, SR has been the preferred algorithm for many wireless systems compared to other ARQ protocols such as Go-Back-N (GBN). Unlike GBN, the SR ARQ algorithm selectively retransmits only the lost PDUs. There are many variations of SR.

However, in this submission we consider a variation of SR that uses CACK and bitmap-based ACK. A bitmap-based ACK combines the benefits of a cumulative ACK and a set of positive and negative ACKs into one message. It is important that TG4 standard specify a unified ARQ approach, with configurable and negotiable parameters that can be used to control a variety of ARQ features. It would very difficult to maintain interoperability, if entirely different approaches are specified to support different flavors of ARQ.

Adaptive Modulation and ARQ

Based on link conditions, the PHY modulation may have to be changed dynamically. When the modulation changes, the capacity of the channel also changes on a per-frame basis. However, it is possible to keep the MPDU size the same, by varying the number of PHY slots required to transmit a given MPDU, when modulation changes. However, changing the MPDU size between retransmissions may be desired under certain types of PHY and/or bandwidth management schemes. The proposed approach in this submission specifies the MPDU as the default retransmission unit, while supporting generic block level retransmission.

The following sections describe the specific texts to be added to the 802.16 TG4 MAC straw man document.

MPDU Sequence Number for Fragmentation and ARQ

Each MPDU is assigned a sequence number (MSN) The MSN along with the FC bits is sufficient for unambiguous re-assembly. The basic ARQ does not require any addition overhead (**Figure 1**(a)), as the same MSN is used for reassembly. This basic mode supports window sizes of up to 32 blocks. If more than 32 blocks is desired, an optional sub-header shown in 1(b) should be included instead of the one in **Figure 1**(a). The maximum sequence number space is limited to 2047 and the ARQ window size must be less than 1024. Similar to option 1(a), option 1(b) is also used for reassembly when ARQ is enabled for a connection. The window size always refers to the number of blocks, where the number of blocks is the same as the number of MPDUs for the default case.

Table 1: TG4 options with different sub-headers (Ref: Figure 1)

NO ARQ (with Fragmentation)	ARQ with window size <= 32 (with or without fragmentation and/or packing)	ARQ with window size >32 and <= 1024 (with or without fragmentation and packing)
1(a)	1(a)	1(b)

FC (2)	MSN (6)

a) Default Fragmentation Sub-header (Window <= 32)

FC (2)	RSVD (3)	MSN (11)

(b) For 32 < Window <= 1024

Figure 1: Fragmentation/ARQ Sub-headers for 802.16 TG4 MAC

TYPE (3)	FC (2)	Length (11)

Figure 2: Packing Sub-header for 802.16 TG4 MAC

Automatic Repeat Request Mechanism

Selective Repeat Algorithm

Selective Repeat with bitmap-based Cumulative Acknowledgments is specified as the ARQ mechanism for 802.16 TG4. To avoid ambiguity, the window size of the SR ARQ must be less than half the size of the maximum possible sequence number. The transmitters are required to retransmit only the MPDUs that are lost. The following subsections describe various options and the actions to be taken at the transmitter and receiver for an ARQ enabled connection. The ACK, Discard and Reset message formats and the methods for handling ACK messages are described.

ARQ Options and Sub-headers

Two MAC sub-headers are defined to support ARQ. Since it is important to optimize the use of optional sub-headers, the options that are most likely to be used together are combined to define sub-headers. **Figure 1** shows the supported options. 1(a) is the default sub-header for 802.16 TG4 MAC, when fragmentation is supported for a connection. If ARQ is enabled for the same connection, the same sub-header is may be used for window sizes up to 32 blocks. The next subsection describes the concept of blocks and Block Sequence Numbering used by TG4 MAC. By default, the fragmentation and ARQ use a per-MPDU sequence number, where each MPDU is given a sequence number irrespective of the number of bytes in the MPDU. However, the block level retransmission can be enabled on a per-connection basis to support a finer control over the retransmission unit at the expense of complexity in implementation. The second option 2(c) is used by connections that require window sizes up to 1024. If fragmentation is not supported by a connection that uses this option, the FC bits should be ignored. The TYPE field of the Generic Header is used to control these options. The appropriate TYPE values are TBD. Table 1 summarizes these options and their applications.

Blocks and Block Sequence Numbering

A block is the minimum retransmission unit for the ARQ and each MPDU may contain one or more blocks. The block size for a connection should be specified between 1 byte and the minimum MPDU size desired for that connection. The default block size is “0”, which means that the block size is obtained dynamically. Some blocks may have less number of bytes than the block size, as there may be gaps in MSDU delivery time, for a particular connection. **Figure 3** shows an example with block size of 64 bytes. Note that blocks 2 and 3 have less number of bytes than the block size (64 bytes).

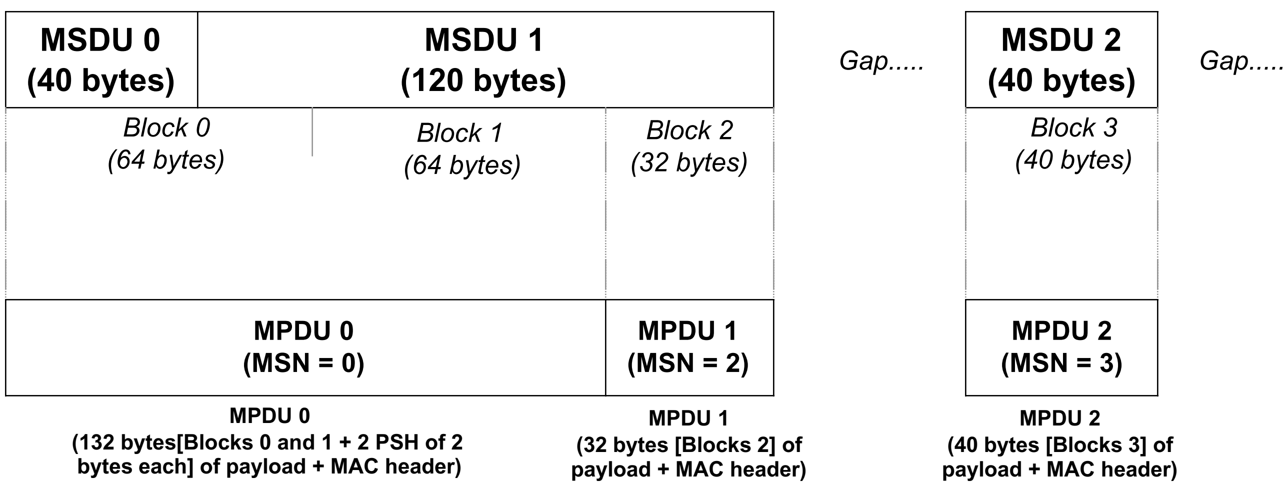


Figure 3: Block Sequence Numbering Example (Block Size = 64 bytes)

Except for the default case, the packing sub-headers and piggybacked control information are not counted towards the block size. A block may span multiple MSDUs, but a block cannot span multiple MPDUs. The number of blocks in an MPDU must be calculated using the simple formula, $\left\lceil \frac{MPDU\_Size}{Block\_Size} \right\rceil$. For the default case, where the Block_Size is dynamic, the MPDU_Size is the total number of bytes in the MPDU, including the packing sub-headers and piggybacked data. For other block sizes, the MPDU_Size is the actual number of MSDU bytes in the MPDU, where an MPDU may contain one or more full or fragmented MSDU.

Blocks are sequentially numbered. The block number and the number of bytes in a block do not change, during the lifetime of the block. The MSN of an MPDU is the BSN of the first block (**Figure 3**). When there is only one block in each MPDU (default case), the MSNs are sequential. For other block sizes, the MSNs are not sequential. When an MPDU is lost, the MPDU may be split into multiple MPDUs during retransmission, as long as the blocks do not span multiple MPDUs and the block size is not dynamic. In other words, the proposed approach allows block-level retransmission. The block size can be negotiated during connection setup and it is entirely up to the implementation and configuration to choose the appropriate block sizes for different connections. For example, a block size of 1 byte provides finer control over the unit of retransmission, but restricts the window size. On the other hand, a block size that is equal to the MPDU size (dynamic block size) requires that the lost MPDUs be retransmitted in its entirety, but supports a bigger window size for the same sequence number space, in addition to its simplicity.

In the default case of dynamic block size, when a receiver correctly receives an MPDU with sequence number *N*, it expects to receive the next MPDU with sequence number *N + 1*. This greatly simplifies the receiver logic and makes the ARQ implementation scalable. The default MPDU sequence numbering is equivalent to a *dynamic* definition of blocks, where the block size of each MPDU is dynamically obtained from the length field in the MPDU header for every MPDU. Therefore, each MPDU carries exactly one block, irrespective of the number of bytes, number of MSDUs and the number of packing sub-headers carried in each MPDU. The ARQ parameter negotiation process specifies a block size of “0” to indicate the default case. Block sizes of greater than zero may be negotiated between SS and BS, if block-level retransmission is desired. Similar to the default case, for block sizes greater than zero, the receiver keeps track of next expected sequence number. When an MPDU is received in order, the receiver updates the next expected sequence number to (Current Sequence Number + $\left\lceil \frac{MPDU\_Size}{Block\_Size} \right\rceil$).

Though the number of bytes within a block must not change between retransmissions, the packing sub-header field should be updated between retransmissions to reflect the changes in fragment size and fragmentation control bits, if an MPDU is split in multiple MPDUs between retransmissions. If fragmentation is not used for a connection, only full MSDUs should be packed into MPDUs. Similarly, the packing can also be disabled on a per-connection basis. The ARQ works the same with or without packing and/or fragmentation.

Acknowledgement Format

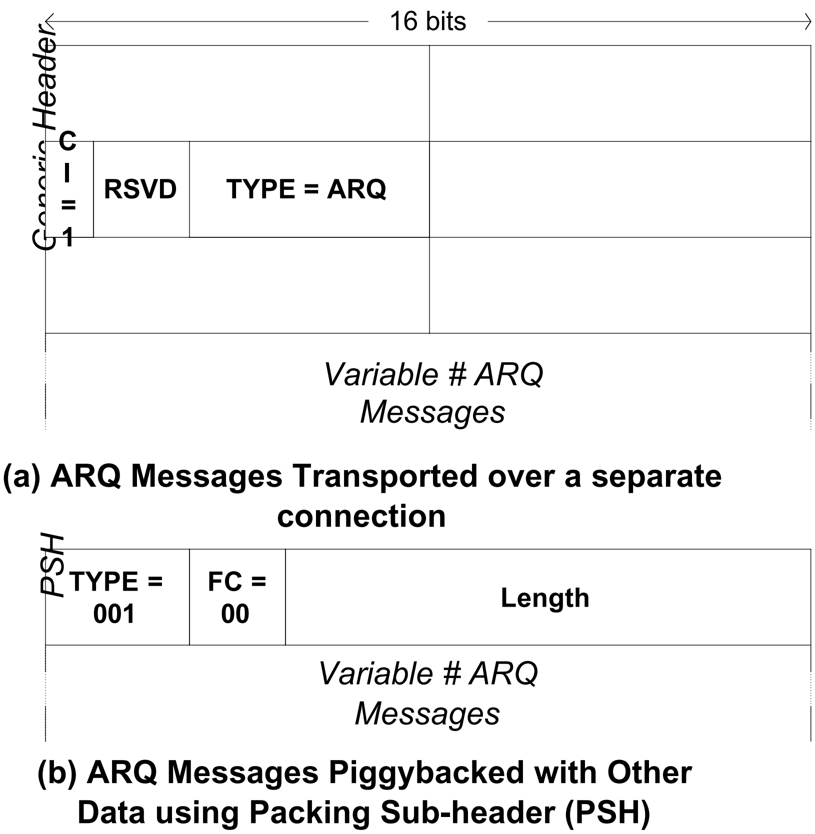
The ACK message is specified as a bitmap as shown in **Figure 5**. A variable length ACK message has the following fields:

- **Connection ID:** The connection ID of the connection for which this ACK message is being generated.
- **Type:** Indicates the type of the message. TYPE = 1 for ARQ bitmap.
- **CACK Flag:** Indicates if the first ACK MAP entry in the message is a Cumulative Acknowledgement. It is not necessary for all ACK messages to have this flag set. However, cumulative ACK must be sent frequently enough to advance the transmitter’s sliding window. Specifying a frequency for CACK is beyond the scope of the standard, however the frequency should be considerably smaller than the window size.
- **ACKC Flag:** This is valid only in the upstream direction. The SS sets this flag if there is ACK message backlog and the messages cannot be piggybacked on to other data MPDUs going upstream. The exact behavior of the BS scheduler upon receipt of a message with this flag set is not defined here, as it is beyond the scope of the standard. The BS scheduler may allocate slots for the management connection for that CPE to carry these ACKS.
- **Length:** The total length of the ACK message in number of ACK MAPs, where each ACK MAP consumes two bytes.
- **ACK MAP (BBN + BM):** The BBN and BM together describe a one-byte bitmap. A single ACK message can have more than one BBN + BM combination. The BBN and BM fields are described in detail below.

An ACK MAP consists of a Bitmap Block Number (BBN) and a Bitmap (BM). The basic idea behind the BBN + BM representation is to create bitmaps in 8-bit increments to support efficient hardware implementations. Since the maximum possible sequence number is 2047, the sequence number space is divided into 256 8-bit blocks. A BBN indicates the Bitmap Block Number within this 256 possible blocks and the 8-bit BM indicates the 8 sequence numbers within a BBN. For example, a BBN of 25 and a BM of 11100101 indicate the status of eight MPDUs stating from sequence number 200 (= 8 * 25). This ACK MAP indicates that the receiver has correctly received MPDUs 200, 201, 202, 205, 207 correctly, whereas MPDUs 203, 204 and 206 were lost. If the CACK bit was set in this ACK message and this ACK MAP is the first MAP in the message, then this also serves as a Cumulative Acknowledgement for MPDU with sequence number 202.

The number of ACK MAPS in a single ACK Message is variable subject to a vendor-specified limit. An ACK PDU is a collection of one or more ACK messages. In order to reduce the impact of ACK traffic on the reverse direction, the ACK PDUs are allowed to piggyback on data flowing towards the transmitter. The scheduler is also allowed to allocate specific slots for the ACK PDUs in both upstream and downstream, in case piggybacking is not possible.

No assumptions should be made about the correctness of packets that fall between two ACK MAPS of the same message. Only the first ACK MAP in an ACK message with CACK bit set should be considered as a positive acknowledgment. When the CACK bit is set for an ACK message, the first ACK MAP's BM portion should have the first (MSB) bit set. A receiver can send bitmaps only for BBNs that address the last correctly received block or before.



(a) ARQ Messages Transported over a separate connection

(b) ARQ Messages Piggybacked with Other Data using Packing Sub-header (PSH)

Figure 4: ARQ Message Transport

ARQ ACK messages can be transported over a separate connection or piggybacked with other data (**Figure 4**). The TYPE field in the packing sub-header indicates whether the PDU that follows the packing sub-header is another full or fragmented MSDU or an ACK PDU. A separate ACK connection is also defined to carry ACK traffic. If the ACK message is transported over this separate connection, a CRC-16 should be appended.

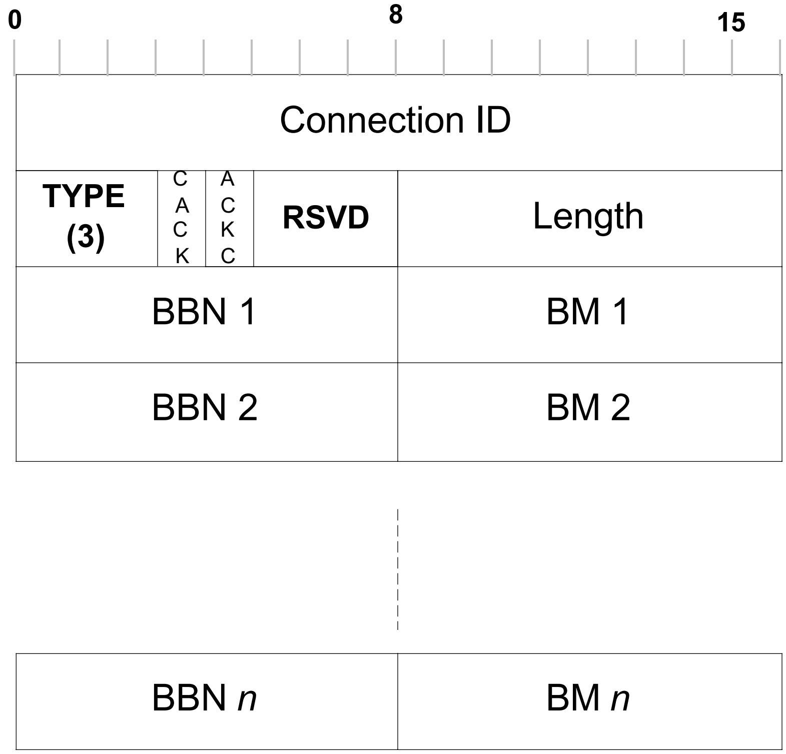


Figure 5: ARQ ACK Message Format with Cumulative ACKS and Bitmap NACKS

Discard and Reset Messages

Discard messages (**Figure 6**) contain the MPDU/Block sequence number up to which the transmitter requests the receiver to skip. When a receiver receives a discard message, it advances its window to the sequence number specified in the discard message. This message informs the receiver that the transmitter wants to discard MPDUs/Blocks with sequences numbers up to but not including the discard MSN specified in the discard message. The conditions for generating discard messages are not specified here.

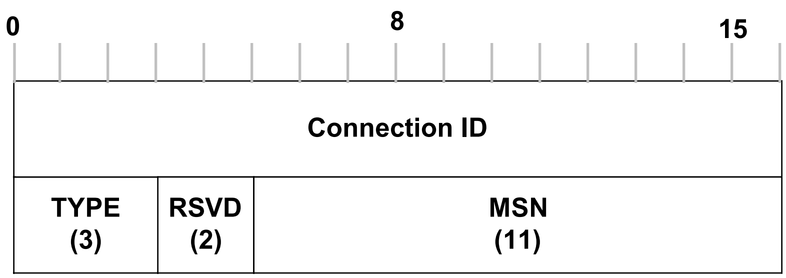


Figure 6: ARQ Discard Message Format

Reset messages (**Figure 7**) are used to reset the ARQ state of a connection. The receiver should acknowledge a reset message and the bottom of windows should be set to zero on both sides upon synchronization. All state information should be cleared and the receiver should discard all data in its buffer. Specifying the exact conditions for generating these messages is beyond the scope of a standard.

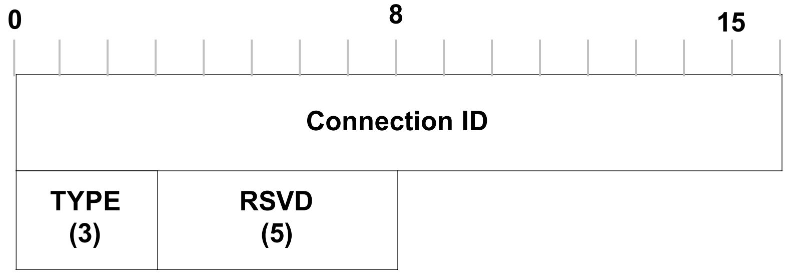


Figure 7: ARQ Reset Message Format

Receiver Actions

- The bottom of the receiver's window is the lowest MSN (L-MSN) not yet correctly received. The receiver should maintain this value as an 11-bit MSN.
- Should check if the MPDU is received error free (CRC Check)
- If the MPDU is corrupted, discard the MPDU and generate an ACK MAP
- If the MPDU is not corrupt and if the MSN is within the receiver's window - keep the MPDU and update appropriate variables.
- Must send periodic CACKs even when the channel is error free. While the exact frequency of this message is not specified in the standard, the frequency should be considerably smaller than the window size of the connection.
- A timer should be used to send duplicate CACK messages.
- If a discard message is received and it is within the receiver's window, advance the window to the sequence number specified in the discard message. The receiver may deliver correctly received packets from its current start of the window up to the discard sequence number to the Convergence Sub-layer.
- If the receiver is an SS, set the ACKC bit in the outgoing ACK messages, if there is a backlog of ACK messages.

Transmitter Actions

- Hold copies of transmitted MPDUs until cumulatively acknowledged by the receiver.
- Set a timer every time the transmitter's window is advanced. This timer should be set to TBD (~5) seconds. Since this is a fairly large timeout value, if the timer expires before any CACK is received, transmitter should send a reset.
- If the transmitter is a BS, and ACKC bit is set fro ACK messages from an SS, the scheduler may allocate slots for the ACK connection of that SS.

ARQ Parameters and Negotiation

The following ARQ parameters need to be specified for connections that have ARQ enabled:

- **Window size:** The window size to be used for this connection. Should be less than or equal to 1024
- **Maximum retry limit:** The maximum number of retransmissions allowed for an MPDU of this connection. The actual number of retries can be made adaptive based on the QoS and other properties of the connection, subject to this maximum limit.

These parameters are communicated to the SS during connection setup.

Specific Comments on IEEE 802.16 TG4 Straw man

We suggest that the following changes be made to the "Straw man Amendment to Standard Air Interface for Fixed Broadband Wireless Access Systems in License Exempt Bands", IEEE 802.16.4.01/10 [11].

1. Include the Section on “Acronyms” on page 1 of this document under section 4 of [11].
2. Replace section 6.2.8 of [11] with section “Automatic Repeat Request Mechanism“ and all subsections under the same of this document
3. Include the section on “MPDU Sequence Number for Fragmentation and ARQ” of this document under section 6 of [11]

References

1) “Draft Standard Air Interface for Fixed Broadband Wireless Access Systems”, *IEEE 802.16/D2-2001*

2) “Automatic Repeat Request (ARQ) Mechanism in HIPERLAN/2”, by H. Li, J. Lindskog, G. Malmgren, G. Miklos, F. Nilsson, and G. Rydnell, *The IEEE Semiannual Vehicular Technology Conference* – VTC2000-Spring, 2000

3) “Broadband Radio Access Networks (BRAN): HIPERLAN Type 2 – Data Link Control (DLC) Layer – Part 1: Basic Data Transport Functions”, *ETSI TS 101 761-1 v1.2.1 (2000-11)*

4) “Digital cellular telecommunication system (Phase 2+): General Packet Radio Service (GPRS) – Overall description of the GPRS Radio Interface Stage 2”, *ETSI TS 101 350 v6.1.0 (1998-10)*

5) “Performance Evaluation of Retransmission Mechanisms in GPRS Networks”, by Q. Pang, A. Bigloo, V. C. M. Leung, and C. Scholefield, *IEEE WCNC'00*, Sep. 2000

6) “Improved Selective Repeat ARQ Scheme for Mobile Multimedia Communications”, by W. S. Jeon, and D. G. Jeong, *IEEE Communications Letters*, Vol. 4, No. 2, 2000

7) “A selective repeat ARQ scheme for point-to-multipoint communications and its throughput analysis”, by S. R. Chandran, *ACM SIGCOMM conference on Communications architecture & protocols*, 1986

8) “Wireless ATM: Limits, Challenges, and Proposals”, by E. Ayanoglu, K. Y. Eng, and M. J. Karol, *IEEE Personal Communications*, Aug 1996

9) “A Proposal for the Enhancement of the 802.16.1 MAC to Provide a per-Service Flow Reliable Connection Service”, *IEEE 802.16 Session 11 (802.16.4c-01/10)*, by Jacob Jorgenson, Ken Pierce and Subbu Ponnuswamy, Malibu Networks

10) Data Integrity in 802.16.4 MAC, *IEEE 802.16 Session 11 (802.16.4c-01/06)*, by Naftali Chayat, Vladimir Yanover, and Inbar Anson, BreezeCOM Ltd

11) Straw man Amendment to Standard Air Interface for Fixed Broadband Wireless Access Systems in License Exempt Bands, *IEEE 802.16.4.01/10*