Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Comments on the use of the TG1 MAC for TG3 Purposes**
Date Submitted	**2000-10-26**
Source(s)	Subir Varma Aperto Networks 1637 South Main Street Milpitas, CA 95035 Voice: (408) 719 9977 Fax : (408) 719 9970 mailto: svarma@apertonet.com
Re:	IEEE 802.16.100/01r4, September 2000
Abstract	The objective of this contribution is to comment on certain aspects of the TG1 MAC protocol, from the point of view of its suitability for TG3. Sections 1 to 3 provide some background and motivation behind the suggested changes, while Section 4 has the concise comments on the TG1 specification
Purpose	
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 comittee 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 an 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>.

Comments on the Use of the TG1 MAC for TG3 Purposes

Subir Varma

Aperto Networks

1.0 Introduction

The proposed TG1 Wireless MAC protocol has two main parts:

- The Wireless Transport Layer which is responsible for the actual transfer of bits from one end of the wireless link to the other. Topics such as encapsulation of higher layer packets, ARQ, Request-Grant mechanisms in the upstream and frame level control fall within its boundaries.
- The Wireless Link Management Protocol, which is responsible for overall control of the wireless link itself. Topics such as system initialization, encryption, flow management, registration, security etc. fall within its boundaries.

The current TG1 MAC protocol is quite adequate for TG3 in the area of the Link Management Protocol, but lacks several features that are essential for a wireless transport operating in the lower frequency bands. The sub-11 GHz bands have the following characteristics that are not found in the higher bands:

- The wireless link is subject to larger delay spreads and thus a higher degree of multipath related fading, as opposed to the higher bands that are affected more due to longer term fading caused by environmental factors. This has obvious implications on the PHY layer, but even the MAC layer needs to be more agile, and have the ability to recover quickly from transient channel error conditions.
- There is a greater degree of co-channel interference due to the fact that the wireless signals propagate farther in the lower bands. The actions taken by the wireless transport layer have a profound effect on the amount of interference that is caused by a transmitter, as well as techniques that the system can use to recover from interference.
- The system is oriented more towards residential and SOHO markets, so that each wireless channel has to support a large number of users, each of whom may be generating sporadic traffic, with low long term average bit rate. Web Surfing would be dominant application running on these systems, so the protocol should be able to handle the traffic generated by TCP applications efficiently.

The suggestion made in this contribution is to adopt the current TG1 Wireless Link Management protocol for TG3, but enhance the TG1 Wireless Transport Protocol in the following areas:

- Provide for greater link robustness: Transient link error conditions should be hidden from higher layer protocols, and the system should be able to recover from them in a transparent manner.
- Provide for greater link agility: The system should have the ability to control and modify a number of MAC and PHY level parameters, in a very dynamic manner.
- Provide for the ability for the system to efficiently handle bursty traffic of the type generated by TCP based applications.

2.0 TG3 MAC Transport Layer

2.1 Encapsulation

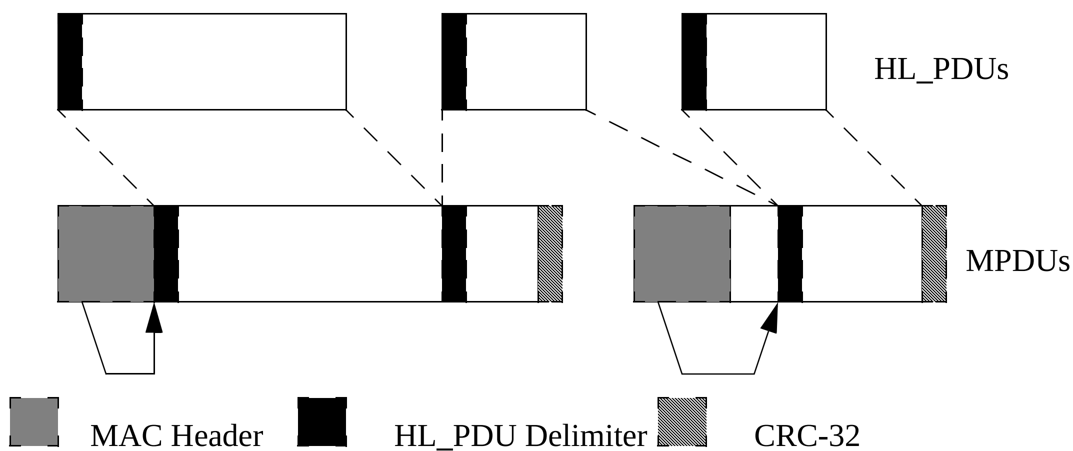


FIGURE 1. Encapsulation of HL_PDUs into MPDUs

The MAC Transport Layer encapsulates all Higher Layer PDUs (CS-PDUs) into MAC layer PDUs (MPDU) at the transmitter. At the receiver, the CS-PDUs s are extracted by means of pointer fields that are inserted into the header of the MPDU, as well as length indicator fields at start of each CS-PDU. The size of an MPDU is not fixed, but is subject to a maximum which is a function of current link conditions.

The proposed transport layer design decouples the MAC layer from higher layers, so that the MAC layer control entity is able to control the size of the MPDU as a function of the current link conditions, and independent of the PDU sizes of the higher layer protocols it is actually carrying. This ability is especially important with the addition of ARQ, since in low error rate environments the optimal MPDU size is larger as compared to high error rate environments, and vice-versa (note that re-transmissions are made at the MPDU level). In the current TG1 MAC scheme, the MPDU size can reduced under control of the BS, but it cannot be increased, since an MPDU cannot contain more than one CS-PDU.

Note that the MAC Header overhead appears only once per MPDU in the proposal, as opposed to once per CS-PDU, thus reducing link overhead. The scenario under which there are a large number of small IP packets waiting for upstream transmission, is quite common, as the following example illustrates: In a TDD based system, for any downstream traffic that uses the TCP protocol, such as FTP or HTTP, there are a large number of 60 byte ACK packets that are generated in the upstream direction (TCP sends one ACK back for every 2 packets that it receives). Under the current TG1 protocol, the BS can give a large grant that can accommodate a large number of these ACK packets, but each of them will have to have their own MAC header. Under the scheme proposed above on the other hand, a large number of ACK packets can be packed in the same MPDU, thus simplifying the protocol and making it more efficient. Since TCP is the most common protocol running in data networks, treating TCP ACKS inefficiently will have a negative effect on system efficiency. In general, since the BS does not know the sizes of the individual CS-PDUs at the SS, the situation in which more than one MAC header appears in an upstream burst can be quite common under the current TG1 protocol.

Another benefit of this approach, which will become more apparent when we discuss the ARQ scheme, is that the proposed encapsulation scheme enables the size (and contents) of the MPDU to change between re-transmissions, thus enabling the protocol to carry varying number of bytes of the CS-PDUs from one transmission to another. The size of an MPDU can change for various reasons in between re-transmissions, for example if the link parameters change, or because of the constraints of the framing structure, the MPDU will not fit within the space left in the frame etc. The current TG1 protocol rigidly constraints the contents of the MPDU to remain the same between re-transmissions, thus reducing the flexibility that the scheduler has in efficiently utilizing the available frame space. Also combined with the point made above, that the BS does not have complete control over the MPDU size in the TG1 MAC, it may mean that there are a lot of smaller size MPDUs being re-transmitted, that the BS has to keep track of.

Note that the proposed scheme does not prevent the BS from giving Grants per Terminal, the only difference is that each MPDU may now contain several CS-PDUs, as opposed to a single one.

2.2 Request-Grant Mechanism

The following Request/Grant mechanism is proposed:

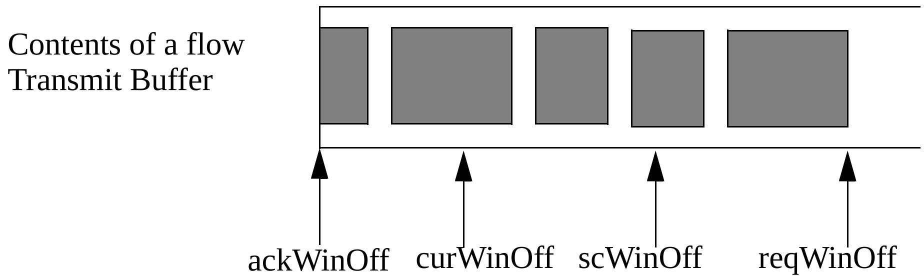
- The SS communicates its current backlog in bytes, to the BS, by means two fields in the MPDU header, namely the reqWinOff and the curWinOff. The reqWinOff counter is updated when a higher layer packet arrives for transmission, and is a running count of the total number of bytes that have arrived into the queue. The curWinOff counter is updated when a MPDU is transmitted and is a running count of the total number of bytes that have been transmitted so far. Both these fields are present in the REQ packet as well as every MPDU sent from the SS. The BS computes the current SS backlog as the difference between these two fields.

- The BS gives grants to the SS, also in bytes, by means of fields in the MAP packet. When the SS receives a grant, it generates an MPDU, with the number of bytes in the payload portion of the MPDU, equal to the grant size.

The curWinOff and reqWinOff fields are used in the Request-Grant process, as well as in the ARQ mechanism, as described in the next section. The proposed Request-Grant mechanism differs from TG1 Request-Grant mechanism in the following ways:

- Due to the presence of the full size piggyback fields in each MPDU, the BS has complete knowledge of the current number of backlogged bytes in each upstream flow. In the TG1 scheme, the SS has limited ability to convey its current backlog to the BS due to the small size (8 bits) of the piggyback field, and has to make a new REQ to do so in most cases. As an example consider the case when the SS makes an initial request for 1500 bytes, and the BS proceeds to give it three grants over the course of the next few frames in order to satisfy this request. If 1000 additional bytes arrive before the final grant, then under the current TG1 MAC, the SS will have to generate an additional REQ packet and then steal some BW from one of the grants, in order to transmit it. This ends up consuming extra BW for the REQ packet, as well as leads to the creation of an additional packet fragment. Under the proposed scheme, the request for the additional 1000 bytes can be piggybacked with one of the MPDUs that are sent upstream.
- The BS gives grants to the SSs not only in terms of number of time units allocated, but also in terms of number of payload bytes in the MPDU. This allows the BS and the SS to maintain very tight control over the transfer of data across the channel, and enables the ARQ mechanism to work. Having a byte based grant mechanism also simplifies the design of the hardware on both the BS and SS ends, since it does not have to do any sophisticated calculations to figure out how much data it can transmit within a certain time interval, since that information is readily available in the MAP.

2.3 ARQ



reqWinOff - ackWinOff = Current Byte Backlog in Transmit Buffer
curWinOff - ackWinOff = Bytes transmitted but not yet ACKed
reqWinOff - curWinOff = Bytes received for transmission, but not yet transmitted
reqWinOff - scWinOff = Bytes received for transmission, but not yet scheduled
scWinOff - curWinOff = Bytes scheduled for transmission but not yet transmitted

FIGURE 2. Counters Maintained at the Transmitter on a per flow basis

Objectives of the ARQ protocol:

- It should be possible to support different levels of ARQ on a per flow basis, for example:

1. No ARQ for voice traffic
2. Limited ARQ for TCP traffic - limited number of re-transmissions, such that the number of re-transmissions can be changed.

- The ARQ protocol should not un-necessarily constrain the peak BW for the flow (by limiting the number of MPDUs per frame, for example).
- The ARQ protocol should avoid the use of timers to control re-transmissions.
- The ARQ protocol should enable the link layer parameters and/or size of the MPDU to change between re-transmissions.
- The ARQ protocol should be robust and recover from various error events, such as loss of ACK packets etc.

2.3.1 Downstream ARQ Protocol

- The BS maintains the reqWinOff, scWinOff, curWinOff and the ackWinOff counters for each flow, at the transmitting end. The reqWinOff counter is incremented when a new CS-PDU arrives, the scWinOff counter is incremented when bytes from the transmit buffer are scheduled, the curWinOff counter is incremented when the bytes actually get transmitted and the ackWinOff counter is incremented when an ACK is received from the receiver. When an CS-PDUs gets scheduled for transmission, the BS creates the MPDU and inserts the curWinOff field into the MPDU header.

- The SS maintains an ackWinOff counter, on a per flow basis. The value of this counter is set to the sequence number of the next byte that the SS expects to receive. If a MPDU is received correctly, then this counter is incremented by the number of bytes contained in the MPDU. If the MPDU is lost or received in error, then the counter is not incremented.
- As long as there are bytes in the flow transmit queue that have not been acked, the BS schedules a special ACK packet in the upstream (on a per flow basis). The SS returns the ackWinOff value in the ACK packet. The SS also indicates in the ACK packet whether the last MPDU in the downstream frame was received correctly or in error.
- If an MPDU is lost, then the SS drops all subsequent MPDUs on that flow, until it receives the one with the expected sequence number. When the BS receives a NACK, it re-transmits all the bytes in the queue with sequence numbers of ackWinOff and greater.
- If one or more MPDUs are not able to get across after N re-transmissions, then the BS drops the first CS-PDU in its transmit queue. It then continues by sending the next HL-PDU, with the same Sequence Number (curWinOff) as the on that the SS is expecting. When the SS starts receiving a new CS-PDU, it drops the incomplete CS-PDUs that it was trying to re-assemble.

2.3.2 Upstream ARQ Protocol

The upstream ARQ protocol that is described in this section has the desirable property that all re-transmissions are controlled directly by the BS. This facilitates the operation of the ARQ protocol, since the BS can allocate upstream BW for re-transmissions, without having to be prompted to do so by the SS.

- The BS updates its own copy of the reqWinOff field by examining the MAC header of REQ and data packets coming from the SSs. It gives upstream data slot allocations in the MAP packet, and updates the scWinOff counter with every grant allocation, by the number of bytes in the payload portion of the grant.
- On receiving an allocation, the SS creates and transmits the MPDUs, and increments its own copy of the curWinOff counter by the number of bytes in the transmission payload. On receiving an CS-PDU, it increments its copy of the reqWinOff counter by the size of the HL_PDU. It puts the curWinOff and reqWinOff counters in the appropriate fields in the MPDU header.
- If an MPDU is lost, then the BS detects this and sends a NACK back to the SS. It also allocates BW for re-transmission of the lost MPDUs. When the SS receives a NACK, it rolls back its curWinOff counter and sets it equal to the ackWinOff counter value received from the BS, and re-transmits the data.
- If an MPDU is not able to get across after N re-transmissions, then the BS sets the flush flag in the ACK. When the SS gets the flush, it drops the CS-PDU at the head of its transmit queue. If there are additional packets in the transmit queue, then it requests BW for them by using the REQ slots.

3.0 Link Layer Parameter Control

The wireless link is subject to greater number of impairments, as compared to wired transmission media. One of the objectives of the MAC and PHY layers is to protect the applications running in the higher layers from these problems. The link layer ARQ scheme described in the previous section offers a first level protection against errors, but does not work very well under extreme conditions. In such situations, the system should have the ability to appropriately change other parameters, at the MAC, PHY or Radio layers, in order to increase the robustness of the transmissions. Among the various parameters that can be varied, the TG1 specification incorporates the ability to control two, namely the modulation and the FEC. However, in general the protocol should have the flexibility to be able to control more than these parameters. An example of a parameter that can be controlled in a more dynamic manner is Transmit Power level. Others include various diversity related parameters such an polarization, antennae etc.

We propose that the MAP packet incorporate a separate field in each IE, upstream and downstream, that describes the set of parameters that are applicable for that burst. This field can be parsed by both the transmitter as well as the receiver, and can be used to appropriately set the link parameters, on a burst by burst basis. The presence of this field will enable the link control algorithms in the BS to react very quickly to changing link conditions, and vary link parameters without the need to exchange messages in advance of doing so.

4.0 Specific Comments on Section 2.5

Replace the contents of Section 2.5 by the contents of Section 4.1 of this document.

4.1 MAC Header Formats

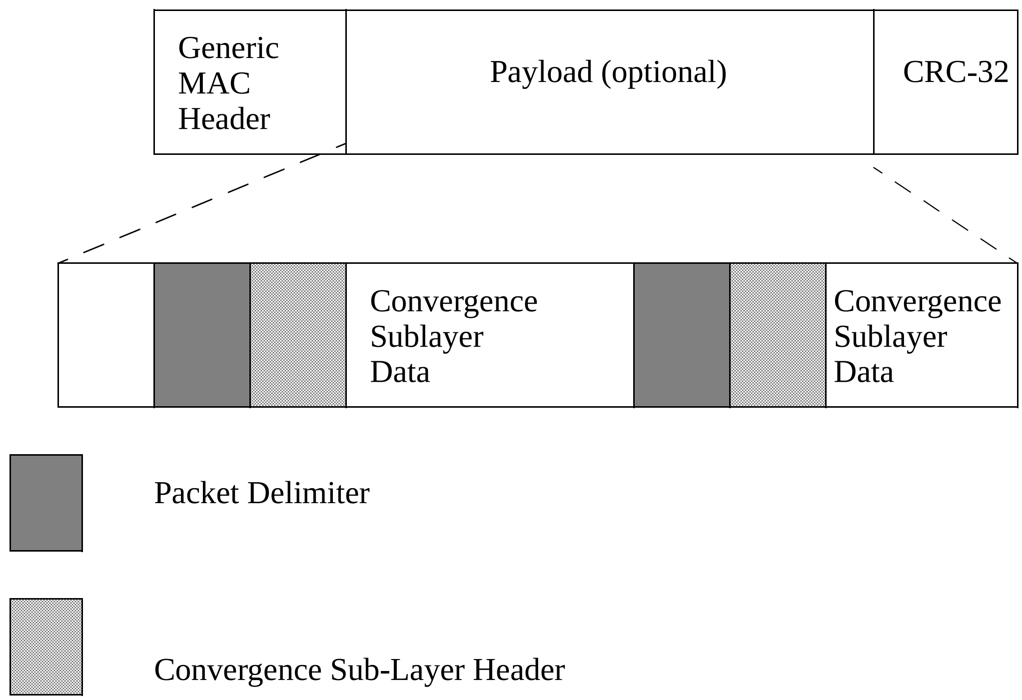


FIGURE 3. MAC PDU Format

MAC Protocol Data Units (PDU) shall be of the form illustrated in Figure 3. Each PDU is preceded by a fixed length generic MAC header. The PDU may contain optional payload information. The payload information can vary in length, so that a MAC PDU will represent a variable number of bytes. The payload information is divided into three parts: A two byte packet delimiter field (Figure 4), an yet to be defined convergence sublayer header and the data portion. This allows the MAC to tunnel various higher layer traffic types without knowledge of the formats or bit patterns of those messages.

A 32-bit CRC is appended to the MAC PDU if the payload size is non-zero. Messages are always transmitted in order: Most-Significant-Byte first with the Most-Significant-Bit first in each byte.

Five MAC Header formats are defined. The first two are generic headers that precede each MAC data message, while the other headers precede MAC management, Bandwidth

Request and Upstream ACK messages respectively. There is a bit field in the Frame Control Byte that is used to distinguish between the various MAC message types.

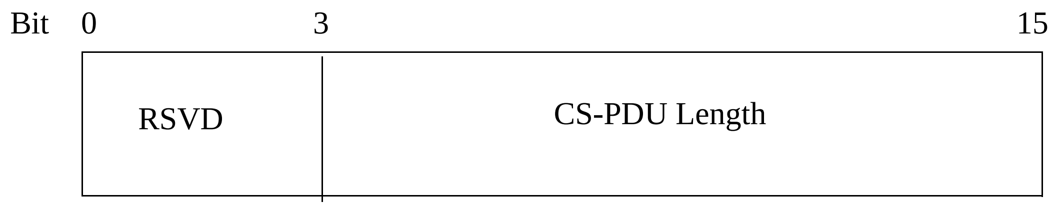


FIGURE 4. Packet Delimiter field

All Higher Layer PDUs (CS-PDUs) are preceded by a two byte delimiter field, whose format is shown in Figure 4. The size of the CS-PDU, in bytes, is inserted into this field.

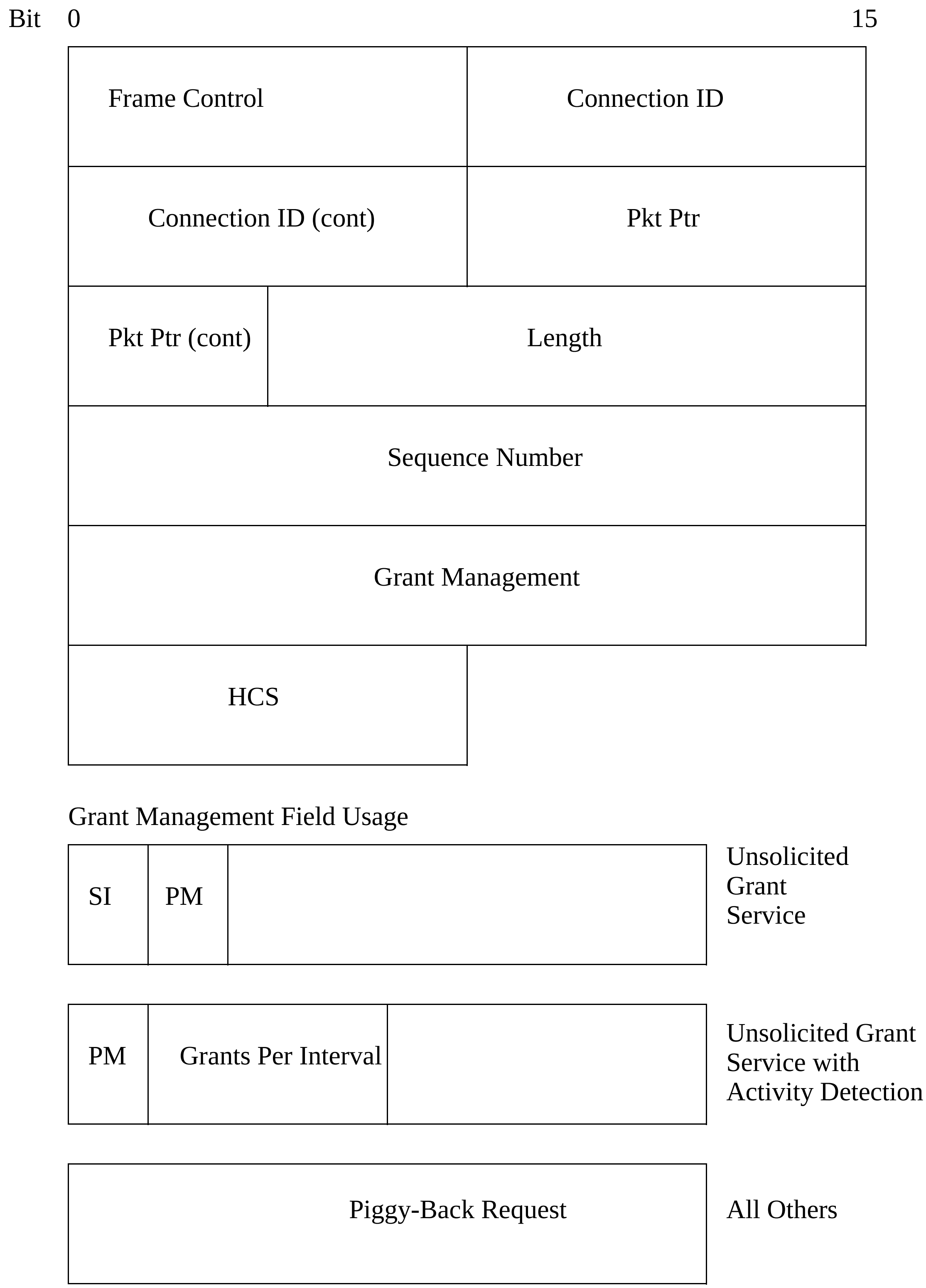


FIGURE 5. Upstream MAC Header

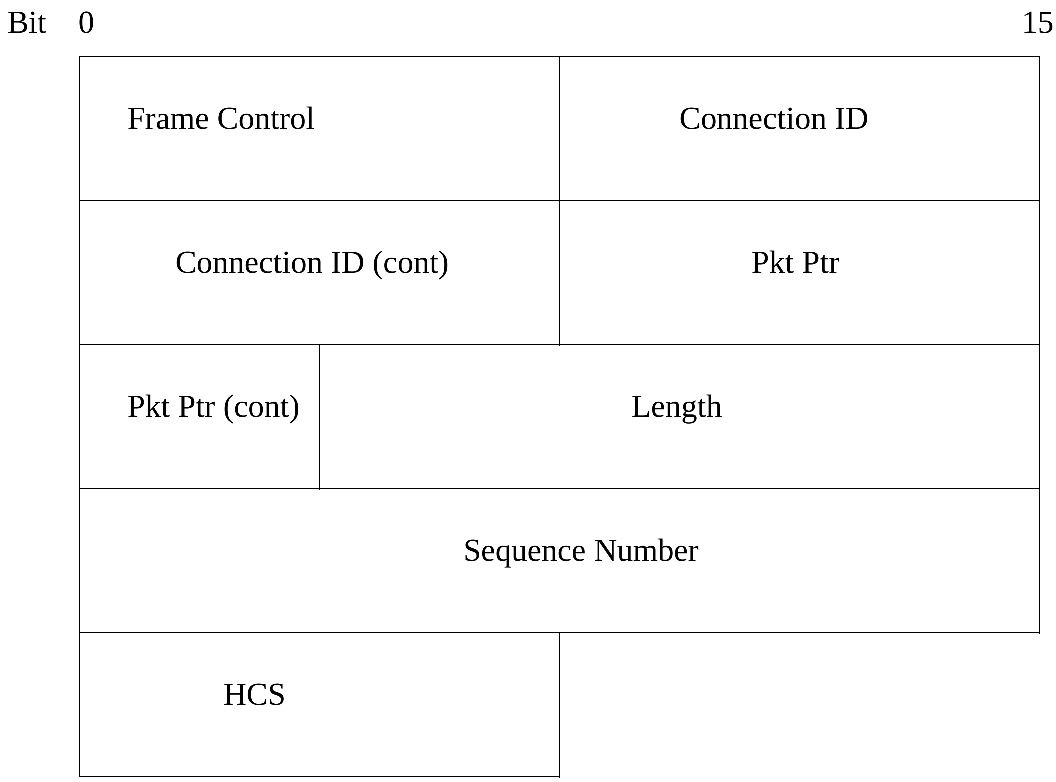


FIGURE 6. Downstream MAC Header

The format shown in Figure 5 shall be used for all PDUs transmitted by the SS to the BS in the uplink direction. For the downlink transmissions, the format shown in Figure 6 shall be used. These two generic header formats are equivalent with the exception of the Grant Management field, which is only present in uplink transmissions.

The Grant Management field is 2 bytes in length and is used to by the SS to convey bandwidth management needs to the BS. This field is encoded differently based upon the type of connection (as given by the Connection ID). The use of this field is defined in Section 2.10.

The format and contents of the Frame Control field is described in Table 1.

Bit 0	15
Frame Control	Connection ID
Connection ID (cont)	Sequence Number
Sequence Number (cont)	BW Request
BW Request (cont)	HCS

FIGURE 7. Bandwidth Request Packet Header Format

The third header is a special format used by a SS to request additional bandwidth. This header shall always be transmitted without a PDU. The format of the Bandwidth Request Header is given in Figure 7.

Bit 0		15
Frame Control Byte	Connection ID	
Connection ID (cont)	Reserved	Length
Length (cont)	HCS	

FIGURE 8. MAC Management Packet Header Format

The fourth header shown in Figure 8, is a special format used for MAC Management messages.

Bit 0	15
Frame Control	Connection ID
Connection ID (cont)	Sequence Number
Sequence Number (cont)	ACK/NACK Status
Link Status	
HCS	

FIGURE 9. Upstream ACK Packet Header Format

The fifth and final header format, shown in Figure 9, is used for upstream ACK packets.

TABLE 1. MAC Frame Control Field Usage

Name	Length (bits)	Description
FC_TYPE	2	MAC Frame Control Type Field 00: Data Packet 01: MAC Control Packet
FC_PARM	4	Parameter bits, use dependent on FC_TYPE Data Packet x x x 0/1: Encryption Key Sequence x x 0/1 x : Encryption Not Used/Encryption Used x 0/1 x x : CRC not appended/CRC appended 0/1 x x x : Convergence Sublayer Indication MAC Control Packet 0000 : Channel Descriptor Packet 0001 : Ranging Packet 0010 : Bandwidth Request Packet 0011 : Upstream ACK Packet
RSVD	2	

TABLE 2.

Name	Length (bits)	Description
Frame Control	8	See Table 1
Connec-tion ID	16	Connection Identifier
Pkt Ptr	12	Points to the first encapsulated HL_PDU, which beginning falls within this MAC packet.
Length	12	Length in bytes of the MAC payload, excluding the MAC header and CRC-32 fields
Sequence Number	16	The value of the curWinOff counter is inserted here for data packets. The value of the ackWinOff field is inserted here for ACK packets.
Piggy-back Request	16	The value of the reqWinOff counter is inserted in this field

4.2 Specific Comments on Section 2.5.2.2

Add the following row to Table 11.

TABLE 3.

Burst Type	DIUC	Comments
ACK	6	Used to ACK upstream data

4.3 Specific Comments on Section 2.5.3

Add the following two figures and text to Section 2.5.3

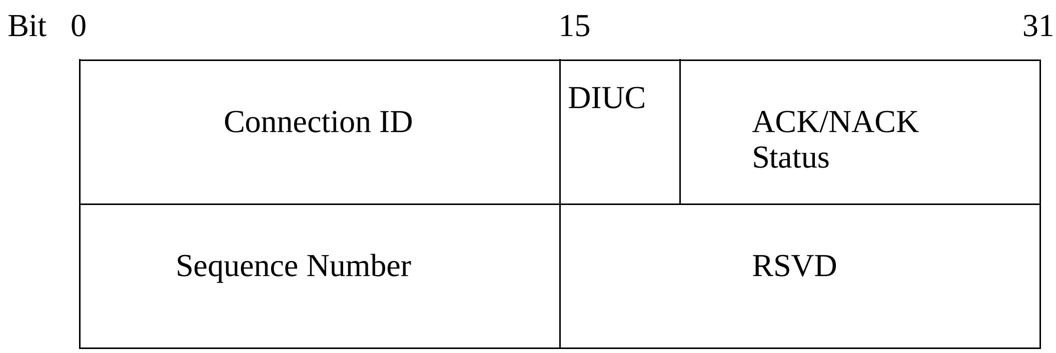


FIGURE 10. DL-MAP Information Element for ACKs for Upstream Data (DIUC = 6)

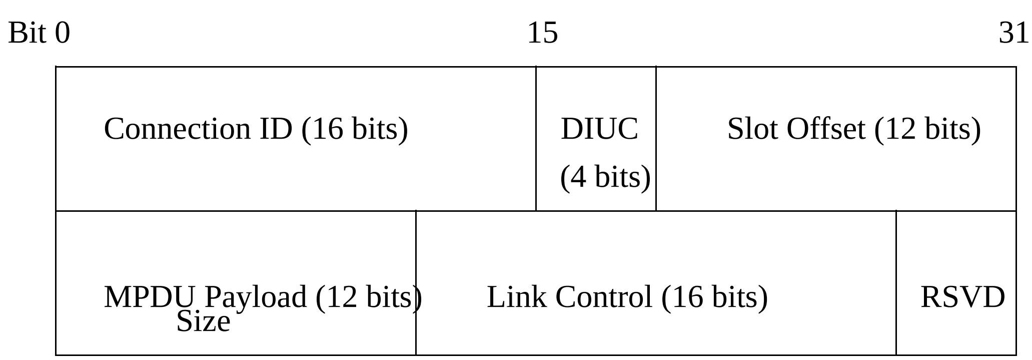


FIGURE 11. TDMA MAP Message Element Format for Downstream Data Grants

If DIUC is either 8, 9 10, 11, 12, or 13, then the MAP IE should be as shown in Figure 11.

4.4 Specific Comments on Section 2.5.4

Add Figure 10 and the following paragraph to this section

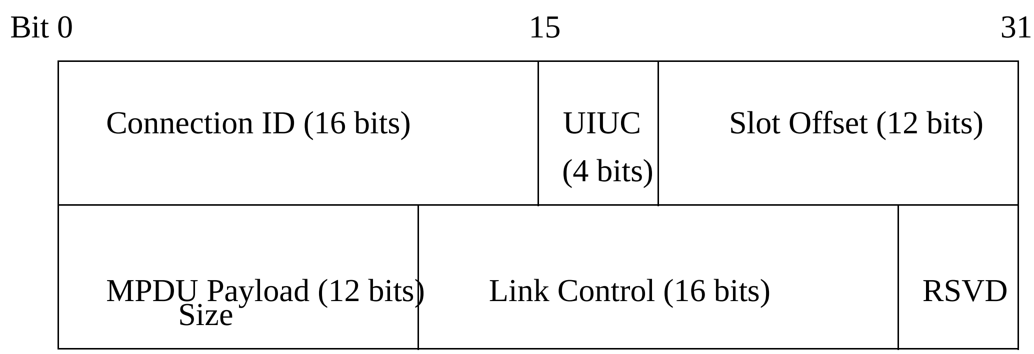


FIGURE 12. TDMA MAP Message Element Format for Upstream Data Grants

If UIUC is either 4, 5, 6, 7, 8, or 9, then the MAP IE should be as shown in Figure 12.

4.5 Specific Comments on Section 2.5.5

Add the following row to Table 13

TABLE 4.

IE Name	UIUC	Connection ID	Mini-slot Offset
ACK	12	unicast	Starting Offset of ACK region