Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**ARQ Proposal for TG3/TG4 Systems**
Date Submitted	**2001-03-07**
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/D2-2001, January 2001
Abstract	TG3 Systems operate in a more hostile environment as compared to systems operating in the higher frequency bands. In this contribution we suggest several techniques that can be used to increase the link robustness of TG3 systems
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>.

Increasing Link Robustness in TG3 Systems

Subir Varma

Aperto Networks

1.0 Introduction

Wireless systems operating in the Sub 10 band are subject to a wider variety of link impairments as compared to the higher frequency bands. Link impairments lead to bit errors and lost packets, which have a very detrimental effect on the performance of data traffic, especially that carried by the TCP protocol. In this contribution, we propose techniques that can be used to increase the robustness of the link:

2.0 Objectives

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.
- The ARQ protocol should be simple to implement, and should be able to scale up to hundreds of connections per point to multipoint link
- Since upstream BW is at premium, the ARQ protocol should not consume an excessive amount of upstream BW for ACK slots.

3.0 Header Formats

Bit 0 15

ARQ Sequence Number

FIGURE 1. Format of the ARQ subheader

Bit 0 15

HT	EC	EKS	Type = SR ACK	ARQ SN (MSB)
ARQ SN (LSB)				CID (MSB)
CID (LSB)				ARQ Bit Map (MSB)
ARQ Bit Map (LSB)				HCS

FIGURE 2. Format of the Selective Repeat Upstream ACK

Bit 0 15

<table>
<tr><td>HT</td><td>EC</td><td>EKS</td><td>Type =
GBN ACK</td><td>ARQ SN (MSB)</td></tr>
<tr><td colspan="4">ARQ SN (LSB)</td><td>CID (MSB)</td></tr>
<tr><td colspan="4">CID (LSB)</td><td>HCS</td></tr>
</table>

FIGURE 3. Format of the GBN Upstream ACK

Bit 0 15

<table>
<tr><td>HT</td><td>EC</td><td>EKS</td><td>Type =
GBN BW Req</td><td>BR (MSB)</td></tr>
<tr><td colspan="4">BR (LSB)</td><td>CID (MSB)</td></tr>
<tr><td colspan="4">CID (LSB)</td><td>ARQ SN (MSB)</td></tr>
<tr><td colspan="4">ARQ SN (LSB)</td><td>HCS</td></tr>
</table>

FIGURE 4. Format of the GBN Bandwidth Request Packet

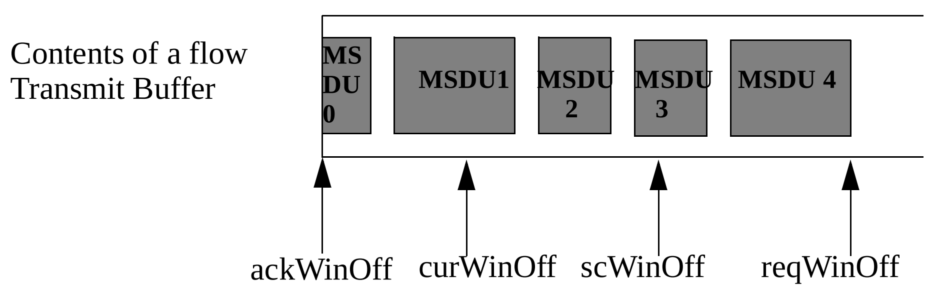
4.0 Comments on the ARQ Scheme

The significant difference between the ARQ scheme in this proposal, and those in competing proposals [3], [4] is that we use a *byte based sequence numbering scheme*. This leads to the following benefits:

- A byte based sequence number fits in very well within an overall MAC protocol structure that also uses byte based upstream REQuests, and a byte based fragmentation/packing scheme (i.e., fragmentation/packing and upstream REQuests do not take MSDU boundaries into account). A byte based sequence number allows the system to treat the contents of transmit buffers purely in terms of bytes occupied, without having to worry about the MSDU boundaries. This considerably simplifies the hardware implementation of the MAC, and allows it to scale to hundreds of active connections (this point is further elaborated in Section 8).
- One of the most widespread ARQ schemes, that used by the TCP protocol, uses a byte based sequence number. This scheme has been well tested, and has been to work well over a period of more than 20 years since it was first introduced.

In the next two sections we give an overview of the algorithm to be followed by the transmitter and receiver, for Go Back N ARQ (Section 5) and Selective Repeat ARQ (Section 6).

5.0 Go Back N ARQ Protocol



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 5. State Variables required to keep track of current ARQ State for GBN

5.1 Downstream GBN-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 MSDU 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 MSDUs gets transmitted, 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 MSDU 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 MSDU, it drops the incomplete MSDUs that it was trying to re-assemble.

5.2 Upstream GBN-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 MSDU, 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 MSDU 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.

6.0 Selective Repeat ARQ

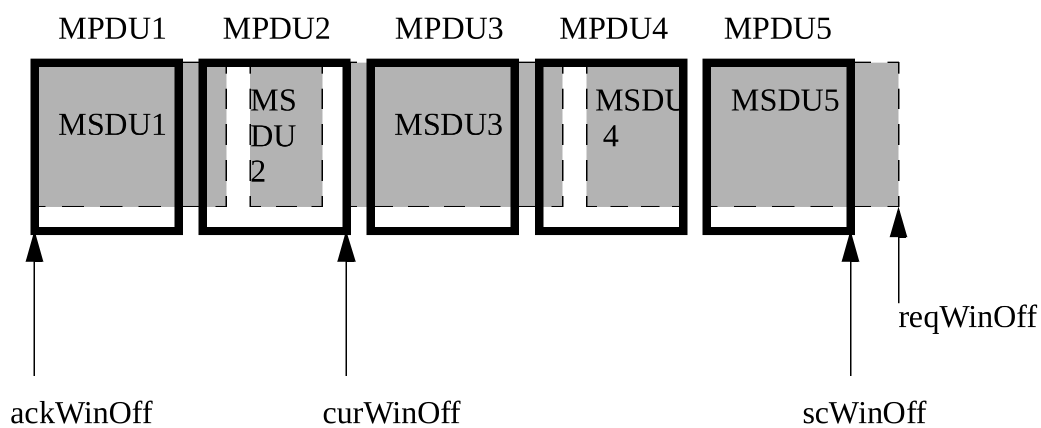


FIGURE 6.

6.1 Downstream SR-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 MSDU 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 MSDUs gets scheduled for transmission, the BS creates the MPDU and inserts the scWinOff field into the MPDU header.

Note: In the GBN ARQ case, the MPDUs are created at the time of transmission, which allows the MPDUs to change in subsequent re-transmissions. In the SR ARQ case, the MPDUs are created at the time of scheduling, and the contents of the MPDU remain fixed in subsequent re-transmissions.

- 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 maintains an ackWinOff counter, on a per flow basis. The value of this counter is set to maximum MPDU sequence number received so far, with the constraint that all MPDUs with lower sequence numbers have been received correctly. If one or more MPDUs are received in error, then the SS returns a NACK to the BS. The NACK has the value of the ackWinOff as well as a 2 byte bitmap containing the status of up to 16

MPDUs that were received after the MPDU with sequence number equal to ackWinOff. The SS buffers out of order MPDUs, until they are re-transmitted correctly.

- When the BS receives a NACK, it re-transmits all the MPDUs in the queue, that have been marked to have been received in error, by the SS. The ackWinOff at the BS is set to the maximum MPDU sequence number that has been acked so for, with the constraint that all MPDUs with lower sequence numbers have been received correctly.

6.2 Upstream SR-ARQ Protocol

The upstream ARQ rules of operation are exactly the same as for the downstream ARQ. Unlike for the GBN ARQ case, the re-transmissions are controlled by the SS, since the BS cannot closely track the state of the SS's transmit queue.

7.0 Interaction with Fragmentation/Packing

The proposed ARQ schemes are independent of the Fragmentation/Packing scheme being employed by the connection. Specifically, they work equally well in all the following cases:

- No Fragmentation, No Packing
- Fragmentation with no Packing
- Packing with no Fragmentation
- Packing and Fragmentation together

8.0 Comparisons with Other ARQ Schemes

Our proposed GBN scheme has the following advantages compared to the other GBN scheme being proposed (in [3]):

- The other GBN Scheme is based on MPDU based sequence numbering, rather than byte based sequence numbering. This does save 1 byte in the sequence number field, but leads to the following problems:

  - The transmitter is forced to keep MPDU based data structures on its transmit queue. In such a situation, it becomes impossible to change the contents of the MPDU between re-transmissions. Such a flexibility in MPDU re-transmission content is crucial in keeping the complexity of the scheduler low, especially in TDD systems. Without such a scheduling flexibility, the following problems arise: (1) The scheduler is forced to find a fit between fixed size MPDUs and empty spaces available in the frame. Thus it takes more time to do scheduling. (2) Lack of scheduling flexibility leads to wastage of bandwidth, for the case when the scheduler is not able to make use of a slot, since none of the available MPDUs would fit into it.

  If the transmitter tries to keep MSDU based data structures in its transmit queue in the GBN Scheme described in [3], then it is forced to keep track of the location of the

MPDUs embedded within the MSDUs. This requires the transmitter to keep track of a state space that increases linearly with the number of queued MSDUs. This makes the scheme very difficult to implement in hardware, and still be scale up to hundreds of connections.

In our proposed GBN scheme on the other hand, there are only four state variables (Figure 5) required to keep track of the ARQ + Packing/Fragmentation state of all the queued MSDUs. The transmitter keeps MSDU based data structures, so that it can easily change the MPDU size between re-transmissions.

In the proposed GBN scheme, even though the transmitter gains the flexibility benefits of maintaining MSDU based data structures, it is not forced to use MSDU as the re-transmission unit (which comes with some disadvantages as explained in the next section).

The proposed GBN scheme allows the BS to explicitly control re-transmissions from the SS. If the channel is experiencing lots of errors, then this feature sigificantly reduces the amount of contention traffic due to re-transmissions in the uplink.

- The proposed SR scheme has similarities to the SR scheme proposed in [3]. The other proposed SR scheme [4] requires that the re-transmission units be MSDUs rather than MPDUs. This can lead to the situation, whereby under high bit error rates, a large MSDU is not able to get through after several re-transmissions. Using MPDU as the re-transmission unit does limit the scheduling flexibility, but that is price that has to be paid for the additional efficiency that SR GBN makes possible. The proposal in [4] causes the ARQ scheme to interact with the fragmentation/packing scheme, so that the two are no longer independent. This leads to complications in specifying the ARQ rules.

Neither of the proposal [3] or [4] allow the upstream ARQ feedback to be sent as independent units. In the case of TDD systems, this can be a major problem, since channel impairments will cause downstream data as well as upstream ACKS to be lost. Our proposal on the other hand allows upstream ACKs to be sent separately, which allows them to be sent with a more robust set of link parameters.

REFERENCES

1. V. Yanover, S. Varma and H. Ye, ‘‘Using the TG1 MAC for TG3 Purposes,’’ *Contribution Number 802.16.3p-00/56*, November 2000.
2. ‘‘Draft Standard for Air Interface for Fixed Broadband Wireless Access Systems’’, *Document Number IEEE 802.16.1/D2-2001*, January 2001.
3. S. Ponnuswamy,‘‘ARQ for TG3 Systems’’, *March 2001*.
4. V. Yanover et. al., ‘‘ARQ proposal for 802.16.3/4 MAC’’, *March 2001*.