Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**A Proposal for ARQ Procedures Based on the IEEE 802.2 Standard**
Date Submitted	**2001-05-11**
Source(s)	Huanchun Ye BeamReach Networks 755 North Mathilda Avenue Sunnyvale, CA 94086 Voice: 408-869-8748 Fax: 408-869-8701 mailto:hye@beamreachnetworks.com
Re:	IEEE 802.16.3-01/12 (TG3 MAC Draft TOC) IEEE P802.16/D3d2-2001 (Draft Standard Airlink Interface for Fixed Broadband Wireless Access Systems)
Abstract	
Purpose	To propose a set of operational procedures for the ARQ protocol based on the IEEE 802.2 LLC standard.
Notice	
Release	
Patent Policy and Procedures	

A Proposal for ARQ Procedures Based on the IEEE 802.2 Standard

Huanchun Ye

BeamReach Networks

Introduction

IEEE 802.2 LLC Standard [1] defines a logical link control protocol for reliable datagram delivery using ARQ. This protocol belongs to a family of protocols derived from the High-Level Data Link Control (HDLC) standard, whose other members include LAPB for X.25, LAPD for ISDN, LAPM for V.42 modem, Frame Relay, SDLC for SNA, and many PPP variants. It is therefore a very familiar and mature protocol, with many commercial and public domain implementations available. The purpose of this contribution is to discuss how to adapt this standard for the purpose of TG3 MAC.

There are two main aspects to an ARQ protocol. One is the definition of retransmission unit. Previous contributions to this forum have discussed this topic extensively. The other aspect is the operational procedures, namely the initialization and updating of communicating ARQ state machines, and how to recover from errors. This contribution will focus on the second topic.

A Summary of ARQ Procedures

This section will provide a brief description of the main elements of ARQ procedures as defined in the IEEE 802.2 LLC standard. Please refer to the official specification, ISO/IEC 8802-2:1998, for more details.

The LLC standard defines a set of messages between communicating ARQ state machines. The messages are classified as commands or responses. A command contains a “poll” bit, and a response contains a “final” bit. When a receiver receives a command with the poll bit set, it must send a response with the final bit set. This provides the ARQ state machines with a logical control channel with minimal overhead, and significantly reduces the amount of state information that must be maintained. Specifically, only one timer is required to keep track of all frames and their possible retransmissions.

The following fields are required in the ARQ header:

C/R	P/F	Control Codes	Rcv Sequence No (optional)	Send Sequence No (optional)
1b	1b	6b	8b (TBC)	8b (TBC)

Here the C/R bit indicates a command or a response. P/F is the poll and final bit mentioned earlier. Control Codes indicate the specific command or responses. Some commands and responses require no sequence numbers, some require a receive sequence number field, and some require both receive and send sequence numbers. We will discuss them shortly.

The sizes of the sequence number fields are determined by the bandwidth-delay products of the physical channel. The TG3 FRD specifies a maximum 10 Mbps channel. The delay is not finalized at this point. As an example, let us assume the round-trip delay is 10 ms, and that on average a frame contains 100 bytes of data. Then the number of outstanding frames in the channel will be

$$(10 \text{ Mbps} * 10 \text{ ms}) / (8 \text{ bits/byte} * 100 \text{ bytes/frame}) = 125 \text{ frames}.$$

The sequence numbers should be at least twice as large as the above number to prevent wrap-around ambiguity. Therefore, in this example, 8 bits are sufficient for sequence numbers.

The LLC standard define the following commands and responses, which are encoded in the Control Codes:

1. SABME (Set Asynchronous Balanced Mode Extended): This is a command used to initialize ARQ state machines, or reset them if errors occur.
   It requires no sequence numbers.

2. UA (Unnumbered Acknowledgement): This is a response used as the reply to SABME. The ARQ state machines can be initialized with a single exchange of SABME and UA.
It requires no sequence numbers.
3. DISC (Disconnect): This is a command used to terminate the logical link.
It requires no sequence numbers.
4. DM (Disconnected Mode): This is a response used as the reply to DISC. The logical link is terminated with a single exchange of DISC and DM.
It requires no sequence numbers.
5. RR (Receive Ready): This can be a command or a response. It is used for inclusive acknowledgement of all received frames. The use of C/R and P/F bits are described in detail in the LLC specification.
It requires the receiver sequence number.
6. REJ (Reject): This can be a command or a response. It is used to request retransmission using go-back-N algorithm.
It requires the receiver sequence number.
7. RNR (Receive Not Ready): This can be a command or a response. It is used to throttle the sender if a local busy condition is detected. This is more important for end stations than for internetworking devices that we are concerned. Thus we may not need this feature.
It requires the receiver sequence number.
8. FRMR (Frame Reject): This is a response used to report unrecoverable errors. The receipt of this message will cause the logical link either to re-initialize or terminate. We may not need this feature if we are only interested in maintaining the logical link and not in detailed error reporting.
9. I (Information): This is actually a user data frame with piggybacked control fields. It can be a command or a response, and requires both sequence numbers. Frames sent with this header require ARQ. (It is associated with DL-Data.XXX primitives.)
10. UI (Unnumbered Information): This is also a user frame. It is a command and requires no sequence numbers. In contrast to the I commands, user data sent with UI commands do not need retransmission. (It is associated with DL-UnitData.XXX primitives.) Therefore, the LLC standard provides a mechanism to bypass the ARQ machines. This is important when different applications are carried over the same connection, where some applications require low data loss and others require low delay.

Note that the LLC standard specifies only the go-back-N algorithm. Its parent standard, HDLC, does provide the selective reject capability. Therefore, it is straightforward to put it back into LLC. We will discuss this topic in the next section.

Selective Reject-Reject

Depending on the frame error rate, selective reject (SREJ) can provide significant higher net throughput than go-back-N alone. With SREJ, the first error (i.e., an out of sequence frame) is always replied with a SREJ. If an implementation supports multiple SREJ, then if another error occurs before the first SREJ condition is cleared, another SREJ can be sent. However, the complexity of the ARQ state machine increases significantly with the number of simultaneous SREJ. Therefore, a common solution is to allow a single SREJ. In this case, if another error occurs before the SREJ condition is cleared, a REJ will be sent and all frames after the second error will need to be retransmitted. This algorithm bears a special name “selective reject-reject.” It is a good choice because the first SREJ provides the biggest improvement [2].

To support selective reject-reject, we simply add another code to the Control Codes list in the previous section:

11. SREJ (Selective Reject): This can be a command or a response. It is used to request retransmission using the selective reject algorithm.
It requires the receiver sequence number.

Conclusion

In conclusion, we propose a set of operating procedures for the ARQ protocol based on the IEEE 802.2 LLC standard. This standard is very mature, and many implementations are already available, so people are familiar with it. The overhead is very low, with many commands requiring only one byte. We considered some minor modifications aimed at simplifying the implementation, such as removing the busy conditions and error reporting. We also propose to add selective reject-reject capability in order to improve net throughput of the airlink.

References

[1] ISO/IEC 8802-2 (1998), available from IEEE.

[2] Uyless Black, Data Link Protocols, Prentice Hall, New Jersey (1993).