ECE 333: Introduction to Communication Networks
Fall 2000

Lecture 9: Data Link Layer V

- **Selective Repeat ARQ**

In this lecture, we continue our discussion of sliding window ARQ protocols. Previously we considered the go back N protocol. This protocol improves on the efficiency of stop-and-wait, by allowing us to "fill the pipe" and not have any idle time at the transmitter. Ignoring any transmission errors, by choosing a large enough window size, go back N can achieve near 100% efficiencies (assuming the header sizes are negligible).

Now let us consider the effect of errors. To recover from errors a timer is once again used. When a packet is not acknowledged before a time-out, the transmitter will go back and begin retransmitting. If the round trip time is known this timer should be set to be approximately equal to the round trip time. When the transmitter times out due to an error in a transmitted packet, all of the packets sent during the time-out time will have to be retransmitted. Even if these other packets had arrived correctly, by the operation of the protocol they would not be saved at the receiver. Such effects can limit the efficiency of go back N. How significant these effects are depends in part on the probability that a frame is in error and the number of frames sent in a round trip time. When needed, an alternative strategy called ***selective repeat*** can be used to improve the efficiency of go back N.

Selective Repeat

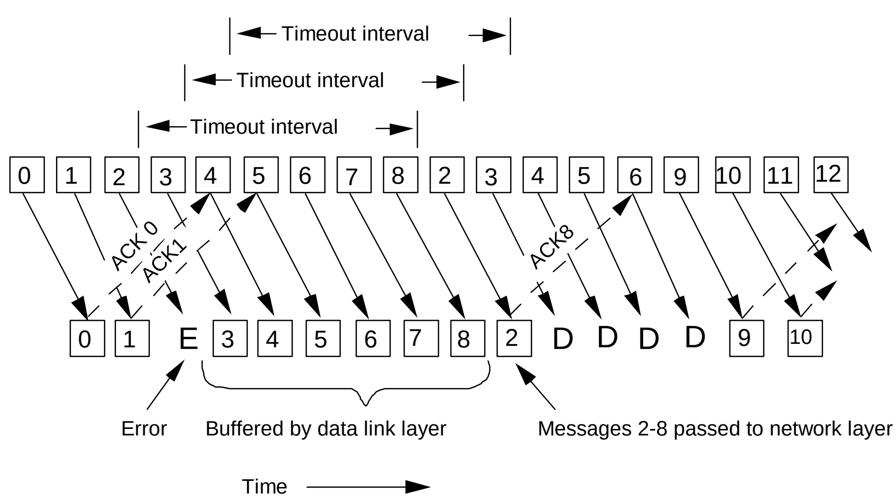
Basic idea: let receiver accept and buffer out-of-order frames that arrive correctly. Then the sender can back up when an error occurs, but will only need to resend the frames in error.

If the sender only retransmits frames in error, then the receiver is going to have to hold onto frames that were received after the one with an error, and save them until the corrupted frame is repeated. This is needed so that the receiver can pass the information to the upper layer in the correct order.

This means that in addition to the sender having a window of frames that it can transmit, the receiver now has a window of frames that it will accept. Specifically we define the ***receiver window*** at each time to start with the first unacknowledged packet and include the next N-1 packets, where N is still denotes the maximum window size at the transmitter. (Note even if one of these packets has been received, we still include it in the receiver window.)

There are several variations of selective repeat. For now assume that the transmitter still retransmits on the basis of a timeout; we will make some improvements on this assumption later. Assume that the receiver returns acknowledgements containing the sequence number of the first unacknowledged packet. In this way an acknowledgement serves as a ***cumulative acknowledgement,*** indicating all previous packets have been correctly received.

Selective Repeat



Not all frames sent in a timeout interval need to be retransmitted.

How Many Sequence Number Do We Need?

With Go Back N, we saw that if n bits are used for sequence numbers then we must have $N \le 2^n - 1$. If instead selective repeat is used, what is the maximum window size?

Note with selective repeat, the window size refers to both the transmitter's window size as well as the receiver's window size. We should not expect to use any larger window size than with Go Back N. Suppose we use the same window size. Specifically, lets again assume that we are using 3 bit sequence numbers. That gives 8 different sequence numbers. Suppose we allow a maximum window size of 7 for the sender and the receiver. The next example shows that this is not enough sequence numbers.

Example with Too Few Sequence Numbers

Consider the following scenario:

Sender: Transmits frames 0, 1, 2, 3, 4, 5, 6
Receiver: Receives frames 0, 1, 2, 3, 4, 5, 6 (all correct)
It advances its windows to allow frames 7, 0, 1, 2, 3, 4, 5.

Assume all these acknowledgments are lost!

Sender: Times out and resends, starting with frame 0
Receiver: Puts it in buffer (since it is in the window). ACK returned will be for frame 6, since 0-6 were received, and new 0 has a hole behind it in the buffer where 7 should go.
Sender: Hearing ACK for 6, knows 0-6 arrived
It then sends 7, 0, 1, 2 . . . 5
Receiver: Receives frame 7 and passes it to network. Since frame 0 is valid, passes it also and is waiting for 1

Wrong frame 0 was given to layer above !!

The Correct Modulus for the Sequence Numbers in Selective Repeat

Solution is to have the maximally advanced receiver window never overlap sequence numbers with the minimally advanced sender's window. If the maximum window size is N, this means $2N$ sequence numbers are required. In other words with n bits available for sequence numbers, we must have $N \leq 2^n / 2 = 2^{n-1}$ for correct operation of selective repeat.

NAKs

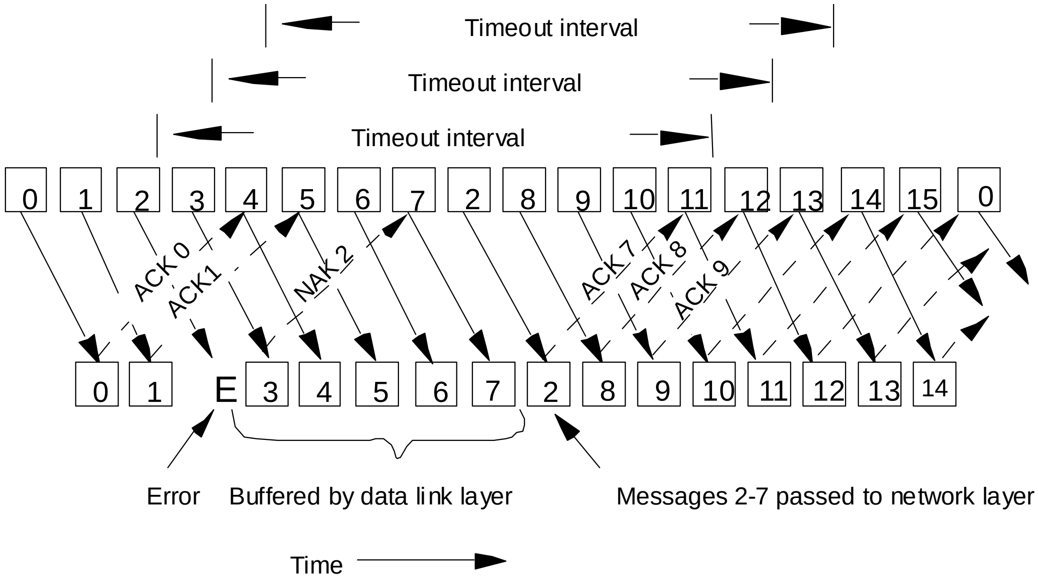
Most implementations of selective repeat is to allow the receiver to send a NAKs (Negative Acknowledgment) when an error is detected (this may be due to a CRC failing or due to an out of sequence arrival.).

The effect of a NAK can also come from a receiving a ***duplicate acknowledgement***, i.e. two acknowledgements in a row for the same packet. Another related idea is to allow the receiver to send a selective acknowledgement, which only acknowledges a given packet, not any prior packets.

When the Transmitter receives an NAK it will go back and repeat only that packet then continue sending new packets in its window. Assuming a large window size and long time-out time, then with the above approach, the sender will end up resending only frames that are in error and not duplicating any correct frames.

An example is shown next:

Selective repeat with NAKs



Go Back N vs. Selective Repeat

Conceptually, the main difference between the "Go Back N" and "Selective Repeat" protocols is the size of the receiver window. (Go back N can be thought of as having a receiver window of size 1).

Implementation-wise, selective repeat requires more sequence numbers to "fill the pipe," more memory at the receiver and is more complex to implement. Selective repeat does result in higher efficiencies than go back N; how significant these gains are depend on the error probability, packet sizes, and the round-trip time.