Go-Back-N (GBN)

9.1 Go-Back-N Warm-Up Questions

Sender and receiver keep separate windows and buffers for sent and received segments.

a) Compare how the sender and the receiver advance their respective windows.

b) Which segments does the sender buffer? When can segments be removed from the buffer?

c) A receiver typically buffers out-of-order segments. What is the advantage of such a buffer?

Cumulative ACKs acknowledge that all segments up to the acknowledged segment have been received.

d) Why are cumulative ACKs used? Do they help with lost data segments, lost ACKs, or both?

When Fast Retransmit is used, the sender retransmits a segment after duplicate ACKs.

e) How does Fast Retransmit improve performance?

f) Compare Fast Retransmit in the case of mild congestion (some segment losses) and heavy congestion (nearly all segments lost).
9.2 Understanding Go-Back-N’s Behavior (Exam Style Question)

Assume you have a Go-Back-N (GBN) sender and receiver. The receiver acknowledges each data segment with a cumulative ACK which indicates the next expected data segment. Furthermore, it saves out-of-order segments in a buffer. The sender and receiver buffer can contain four segments each. The time-out period is much larger than the time required for the sender to transmit four segments in a row.

a) The sender wants to transmit 10 data segments (0, . . . , 9) to the receiver. Assume that exactly one segment is lost. How many segments has the sender to transmit in the best (resp. worst) case? For each case, indicate which segment was lost.

b) Once again, the sender wants to transmit 10 data segments (0, . . . , 9) to the receiver. This time, assume that exactly one ACK is lost. How many segments does the sender have to transmit in the best (resp. worst) case and which ACK was lost?

c) Assume the sender just transmitted segments 4, 5, 6 and 7 and is now waiting for ACKs from the receiver. It receives three times an ACK with number 4. Therefore, it cannot remove segments from its buffer and eventually the timeout is reached. Following the GBN protocol, the sender will retransmit all four segments.

A friend of yours explains that she improved her GBN algorithm so that, in the case above, the sender would just retransmit data segment 4 (instead of all four segments). She tells you that, quite often, she would then get an ACK with number 8 back (all four packets were successfully transmitted).

Can you explain why your friend believes that only data segment 4 was missing? Under which network conditions would the proposed improvement not work (assuming you still get three times an ACK with number 4)?
Consider a Go-Back-N (GBN) protocol with the following implementation choices for sender and receiver.

- The sender and receiver window have a size of 4;
- The receiver saves out-of-order segments in an (infinite) buffer and removes them as soon as the missing segment(s) arrive;
- The receiver uses cumulative ACKs which acknowledge all previous segments and point to the next expected data segment;
- The sender uses Fast Retransmit. After three duplicate ACKs, the sender immediately retransmits the corresponding data segment. For instance, if the sender gets the following ACKs \([A_1, A_1, A_1]\), it will immediately retransmit the data segment \(D_1\);
- For each tick in the diagram below, the sender can send one data segment and the receiver can send one ACK. Sender and receiver will first analyze the incoming packet and then send a data segment/ACK;
- The sender uses a retransmission timer of 5 ticks. Each time it sends a data segment or receives an ACK, the timer is reset. After a timeout, the sender retransmits all current segments in its sender buffer (in order, one segment per tick);
- A data segment or ACK needs two ticks to travel to the other end of the connection. See the given start in the diagram.

**Your task:** Use the diagram on the next page to draw the successful transmission of 6 data segments \((D_0 \text{ to } D_5)\) if the **first data segment** \((D_1, \text{already indicated})\) is lost as well as **ACK A5 is lost the first time it is sent**. For each tick, indicate which data segment or ACK is transmitted (if any) as well as the content of the sender and out-of-order buffer.
Consider the situation in which two hosts, A and B, are concurrently using a 1 Mbps link with a Maximum Segment Size (MSS) of 100 kb.

Assuming that B starts with 500 kbps and A with 200 kbps (see left picture). Describe the evolution of the throughput of the two hosts when:

a) A and B rely on Additive Increase Multiplicate Decrease (AIMD).

b) A and B rely on Multiplicative Increase Additive Decrease (MIAD).

Assume now that only A is malicious, and wants to cheat congestion control to get more throughput. Describe two distinct ways A could do so and what would be the net effect on B’s throughput.
Consider the following plot which depicts the evolution of the size of the TCP congestion window of the sender.

What kind of network conditions is this flow seeing?

Describe briefly:

a) What happens at point B?

b) Does the event happening at point B require the network to discard packets? Why or why not?

c) What happens at point E?

d) Does the event happening at point E require the network to discard packets? Why or why not?

Consider that the Maximum Segment Size (MSS) of the connection is 1 kB and the Round-Trip Time (RTT) between the two end points is 100 milliseconds. The sender opens the connection at time $t = 0$. Transmission delay in this network is negligible, so you should only consider the propagation delay in the following.

e) How much time has elapsed at point A?

f) How much time has elapsed between point C and D?

Think about how come point D is higher than point B. Would you expect this to happen often?