ETH zürich

Networked Systems
ETH Zürich — seit 2015

Communication Networks

Prof. Laurent Vanbever

Solution: Exercise 10 – Go-Back-N

10.1 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?

Solution:

a) The sender can advance its window when an in-order acknowledgment arrives. The receiver can advance its window with each in-order segment.

b) The sender needs to buffer all sent, but unacknowledged, segments. When the segment with the lowest sequence number is acknowledged, it can be removed.

c) An out-of-order buffer helps to reduce unnecessary retransmission of already received segments.

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?

Solution:

d) Cumulative ACKs improve the behaviour for lost ACKs, as each ACK covers all previous ones, which can help to avoid retransmitting already received segments due to lost ACKs.

When *Selective Repeat* is used, the sender retransmits a segment after duplicate ACKs.

e) How does Selective Repeat improve performance?

f) Compare Selective Repeat in the case of mild congestion (some segment losses) and heavy congestion (nearly all segments lost).

Solution:

e) With duplicate ACKs, the sender can detect isolated segment loss and can quickly resend segments, without waiting for timeouts, thus improving performance.

f) Selective Repeat works best for mild congestion, as explained above. In the case of heavy congestion, not enough segments may arrive to even receive a significant number of duplicate ACKs. In this case, the sender needs to wait for the timeout.

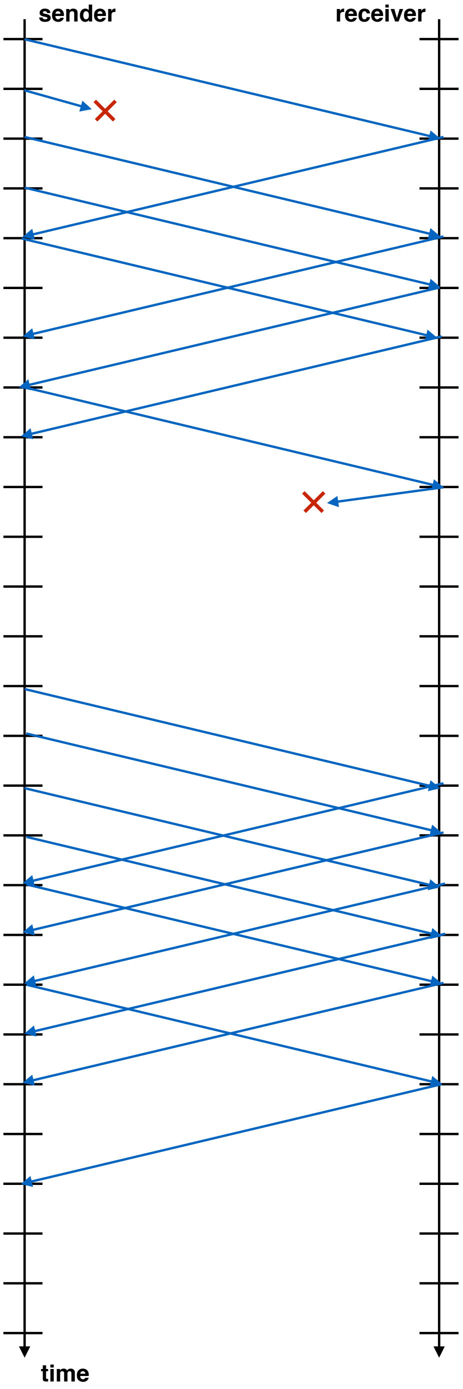
10.2 Go-Back-N (Exam Question 2017)

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 Selective Repeat. After three duplicate ACKs, the sender immediately retransmits the corresponding data segment. For instance, if the sender gets the following ACKs [A1, A1, A1], it will immediately retransmit the data segment D1;
- 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 (D0 to D5) if the **first data segment** (D1, 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.

Solution:



sender buffer	data segment	ACK segment	out-of-order buffer
D0	D0	-	-
D0, D1	D1	-	-
D0, D1, D2	D2	A1	-
D0, D1, D2, D3	D3	-	-
D1, D2, D3, D4	D4	A1	D2
D1, D2, D3, D4	-	A1	D2, D3
D1, D2, D3, D4	-	A1	D2, D3, D4
D1, D2, D3, D4	D1	-	D2, D3, D4
D1, D2, D3, D4	-	-	D2, D3, D4
D1, D2, D3, D4	-	A5	-
D1, D2, D3, D4	-	-	-
D1, D2, D3, D4	-	-	-
D1, D2, D3, D4	-	-	-
D1, D2, D3, D4	D1	-	-
D1, D2, D3, D4	D2	-	-
D1, D2, D3, D4	D3	A5	-
D1, D2, D3, D4	D4	A5	-
D5	D5	A5	-
D5	-	A5	-
D5	D5	A6	-
D5	-	-	-
-	-	A6	-
-	-	-	-
-	-	-	-

10.3 Reliable Transport[2] (Exam Style Question)

On the next page you see the beginning of a communication between two end-points using the Go-Back-N protocol with Selective Repeat. Consider that the sender has infinitively many data segments to send and they are immediately available.

We ask you to fill in the missing values in the two tables. Stop if you either reach the bottom of the tables or the sender is no longer able to send new data segments because its buffer is full. Start with the blue row indicated on the left.

Note: Please read the entire question carefully!

Set-up:

- Every table row corresponds to one time-slot. The sender and receiver can send one data segment respectively ACK segment in every time-slot;
- Consider that the *Sender buffer* contains all the sent but not yet acknowledged segments, while the *Out-of-order buffer* contains all the messages which has been received. . . out-of-order;
- If the sender receives an ACK in one time-slot, it first processes the ACK (e.g. removes segments from the sender buffer) and then sends the data segment for this time-slot. Similarly, the receiver will first analyse the received data segment and then send a corresponding ACK;
- The link between the sender and receiver is not reliable. The first data segment with a sequence number of 3 and all data segments with a sequence number of 5 are dropped and do not reach the receiver.

Sender behavior:

- The sender uses Selective Repeat after receiving 3 duplicate ACKs. That means as soon as the sender receives an ACK with the same sequence number for the third time, it will retransmit the missing segment in the same time-slot (instead of a new data segment);
- The sender can store at most 5 unacknowledged segments in its sender buffer.

Assumptions:

- You will not reach the maximal sequence number. No overflow;
- The timeout value is very long and will not occur;
- The receiver out-of-order buffer can store an unlimited number of segments.

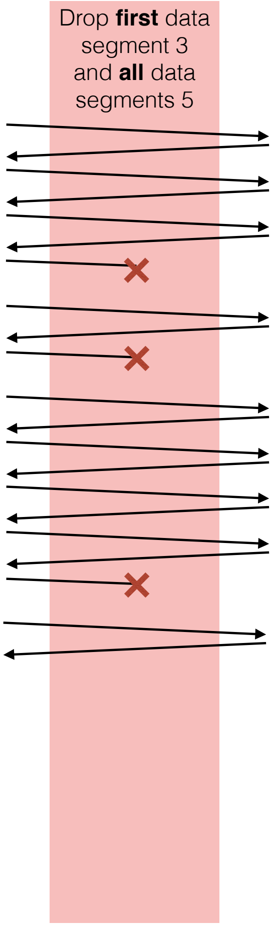
Solution: The acknowledgments always point to the next expected sequence number and not to one of the received segment. This means that, for example, the segment with sequence number 5 is acknowledged with 6.

Fill the following table starting from the blue row on the left.

GBN - Sender
maximal sender buffer size = 5

Sender buffer	Data segment to send	Received ACK number
0	0	–
1	1	1
2	2	2
3	3	3
3, 4	4	–
3, 4, 5	5	3
3, 4, 5, 6	6	–
3, 4, 5, 6	3	3
5, 6, 7	7	5
5, 6, 7, 8	8	5
5, 6, 7, 8	5	5
5, 6, 7, 8, 9	9	–
5, 6, 7, 8, 9	–	5

Solution:



GBN - Receiver

Received data segment	ACK number to send	Out-of-order buffer
0	1	–
1	2	–
2	3	–
–	–	–
4	3	4
–	–	4
6	3	4, 6
3	5	6
7	5	6, 7
8	5	6, 7, 8
–	–	6, 7, 8
9	5	6, 7, 8, 9