Homework3: Transport Layer

COM-208: Computer Networks

In most of the problems, you will be asked to complete a diagram, stating all exchanges, events and actions taken by the transport layers of a sender and a receiver. For each exchanged segment, indicate whether data or acknowledgment (ACK) segment, its sequence (SEQ) or ACK number, and whether the segment was lost. Also, indicate any timeout events and changes to the sender or receiver window.

We define the "duration of a file transfer" as the amount of time that elapses from the moment the sender transmits the first bit of the file until the moment the receiver receives the last bit of the file. Whenever computing the duration of a file transfer, assume that segment headers and ACK segments have insignificant size (i.e., their transmission delay is insignificant).

Go-Back-N in the presence of data-segment loss

End-system A is communicating with end-system B using a Go-Back-N (GBN) protocol with window size $N = 4$ and valid sequence numbers ranging from 0 to 10. The network between the two end-systems may drop segments, but it never reorders or corrupts segments. Suppose B has received and acknowledged data segments with SEQs 0 and 1, and it is expecting data segment with SEQ 2. Instead, it receives data segments with SEQs 4, 5, 2, and 3 (in this order). What has happened? Which (data or ACK) segments have been lost? How does B respond upon receiving each data segment?

Complete the diagram in Figure 1. We have already filled in some of the information: segment 2 was lost and A timed out waiting for an ACK to segment 2.

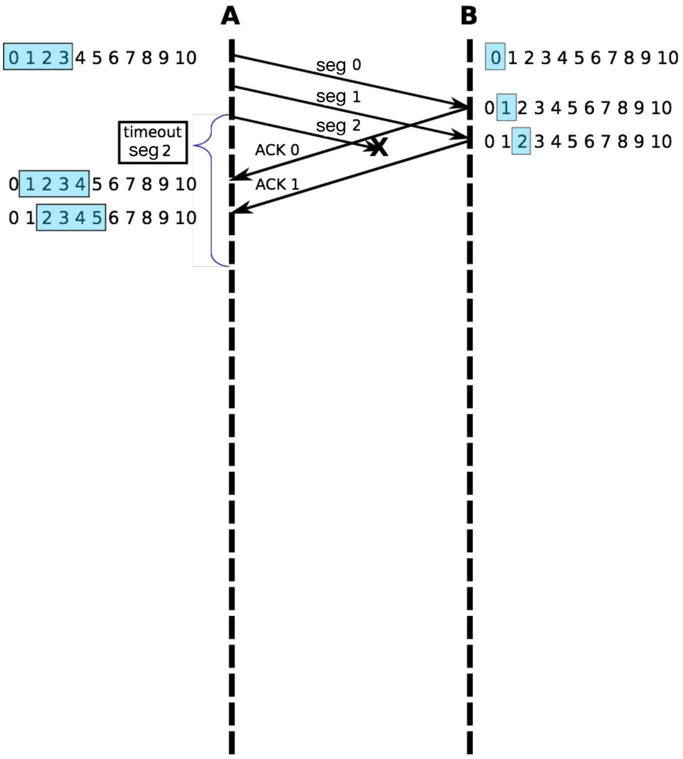

Figure 1

Now also with ACK-segment loss

End-system A communicates with end-system B using a GBN transport-layer protocol with sender window size $N = 4$ and valid sequence numbers ranging from 0 to 10. The network between them may drop segments, but it never reorders or corrupts segments. End-system B receives data segments with the following SEQ numbers (in this order): 0, 1, 3, 4, 5, 2, 3, 4, 5. End-system A receives ACK segments with the following ACK numbers (in this order): 1, 1, 1, 2, 3.

Complete the diagram in Figure 2. We have already filled in some of the information to help you get started: end-system A sent data segments with SEQs 0 and 1, which reached end-system B; ACK 0 was lost.

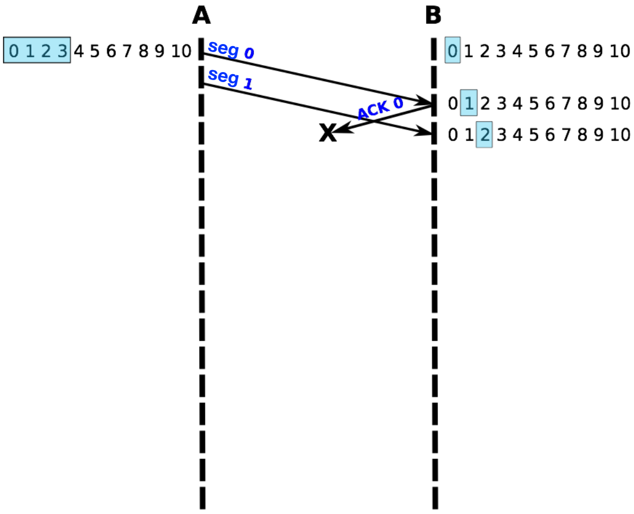

Figure 2

A more challenging scenario

End-system A communicates with end-system B using a GBN transport-layer protocol with sender window size $N = 3$. The network between them may drop segments, but it never reorders or corrupts segments. A wants to send a file to B. It splits the file into 8 data segments with SEQs from 0 to 7. In the entire duration of the file transfer, A receives ACK segments with the following ACK numbers (in this order): 1, 1, 4, 6, 7.

Complete the diagram in Figure 3. We have already completed some of the information to help you get started: A sends data segments with SEQs 0 and 1, which reach B. B sends ACKs 0 and 1, but only ACK 1 reaches A.

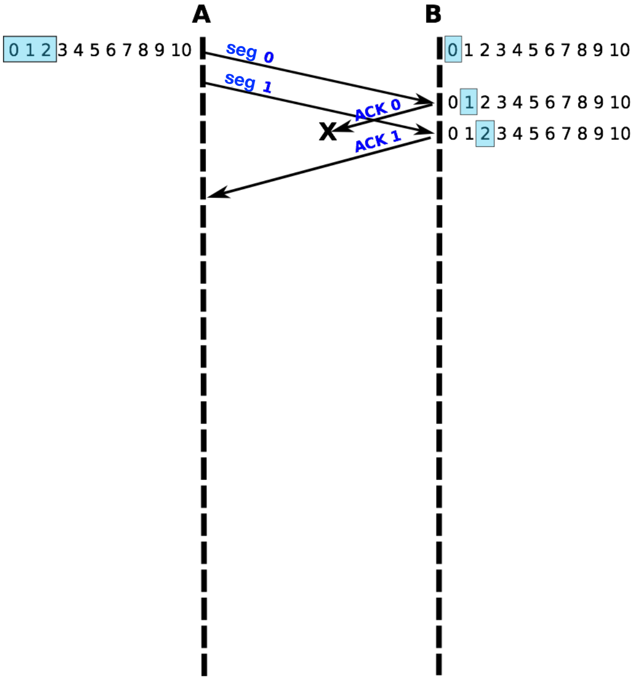

Figure 3

Thinking creatively about Go-Back-N

An end-system wants to send 4 data segments over a network that may drop segments, but it never reorders or corrupts segments.

- In the GBN protocol, what mechanism is used by the sender to detect the loss of data segments?
- Are there situations where the sender has sufficient information that could be used to detect a segment loss sooner and, thus, recover faster? How would you improve the sender algorithm? What is the minimum amount of time needed to detect the loss of a data segment? Think about the pattern of acknowledgments received by the sender when a data segment is lost.
- Can you identify any problem with the modified sender algorithm if the network starts reordering segments?
- Propose a solution that makes the modified algorithm work better in the case where the network can reorder segments.

Thinking about different transport-layer protocols

End-system A is connected to end-system B over a link that has propagation delay d_{prop} and transmission rate R (in each direction). The link is reliable and does not drop, reorder or corrupt segments. A wants to send a file of size F to B.

- What is the minimum duration of the file transfer (regardless of the transport-layer protocol used)? Assume that the entire file fits in a single segment.
- Suppose that the file-transfer application uses a Stop-and-Wait transport-layer protocol What is the duration of the file transfer? Assume data segments of size P and that P perfectly divides F (i.e. F/P is an integer).
- Suppose that the file-transfer application uses a GBN protocol with window size N. What is the duration of the file transfer? How does your answer compare to the result you found for the Stop-and-Wait protocol? Assume data segments of size P and that P perfectly divides F (i.e. F/P is an integer). Assume that the window size N perfectly divides the number of segments that A transmits during the file transfer.
- Would you prefer a GBN or a Selective Repeat (SR) transport-layer protocol to perform this file transfer over this specific link? Justify your answer.

Now with segment loss

An end-system wants to send 4 data segments over a network that may drop segments, but it never reorders or corrupts segments. Compute the time it takes for the sender to send all data segments and receive all acknowledgments in each of the following scenarios. If there are many possible answers, identify all of them. Assume that the round trip time between the sender and the receiver is $RTT = 100$ ms, the transmission delay of any single segment is negligible, and the retransmission timer duration is $TO = 400$ ms.

- The sender uses a GBN transport-layer protocol with window size $N = 4$. One data segment (from the sender to the receiver) is lost the first time it gets transmitted; no other segment is lost.
- The sender uses a GBN transport-layer protocol with window size $N = 4$. An ACK segment is lost the first time it gets transmitted; no other segment is lost.
- The sender uses an SR transport-layer protocol with window size $N = 4$. A data segment is lost the first time it gets transmitted; no other segment is lost.
- The sender uses an SR transport-layer protocol with window size $N = 4$. An ACK segment is lost the first time it gets transmitted; no other segment is lost.

A more challenging scenario

End-system A wants to send a file to end-system B using a transport-layer protocol that provides reliable data delivery with pipelining and sender window $N = 5$. A's transport layer splits the file in 10 data segments with SEQs from 0 to 9.

- The first data segment from A to B is lost the first time it is transmitted, while all other segments arrive at their destination uncorrupted and in order. Complete the two diagrams in Figure 5 in each of the following two cases: (i) A and B use GBN, and (ii) A and B use SR.

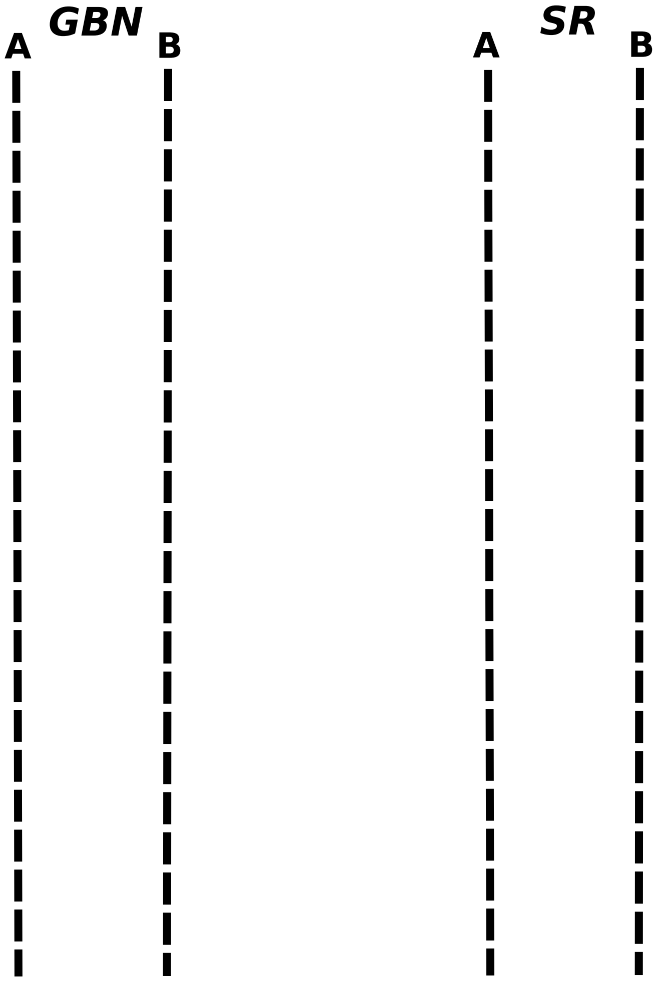

Figure 5

- The first 4 ACKs from B to A are lost the first time they are transmitted, while all other segments arrive at their destination uncorrupted and in order. Complete the two diagrams in Figure 6 in each of the following two cases: (i) *A* and *B* use GBN, and (ii) *A* and *B* use SR.

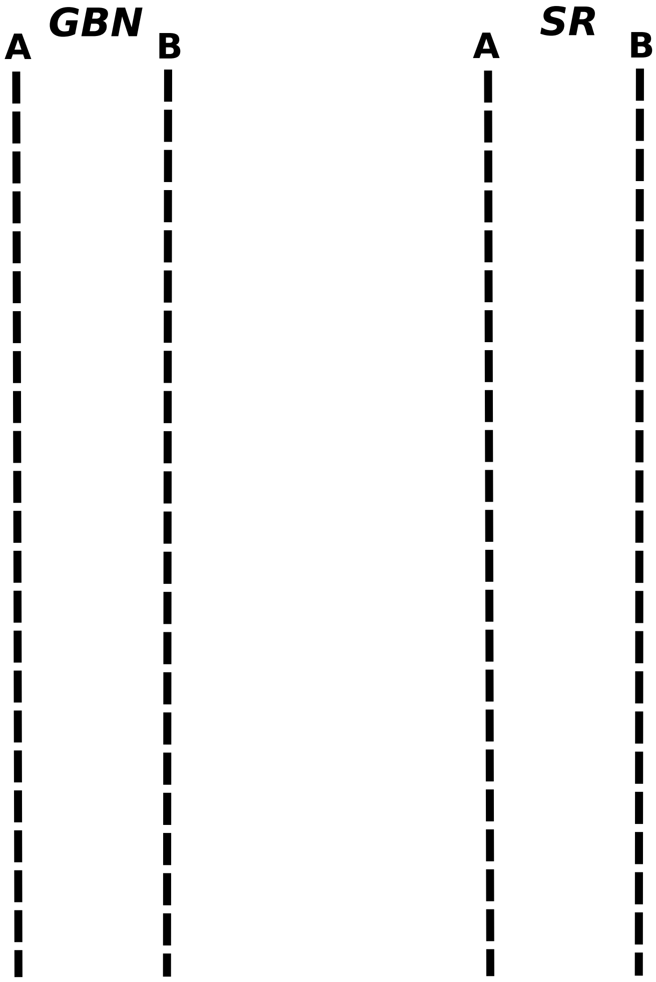

Figure 6

- Based on your answers to the previous two questions, can you draw a general conclusion about when one should prefer GBN over SR and vice versa? Justify your answer, e.g., if you state that one should prefer GBN in situation X, please say what metric GBN improves relative to SR in situation X.

Reliable data delivery over unreliable networks

Table 1 describes different network channels between a sender A and a receiver B. E.g., in channel (3), there may be segment corruption only on the path from A to B, and there needs to be support for pipelining. There is no segment reordering in any of the channels.

channel ID	errors in $A \to B$	errors in $B \to A$	support for pipelining?
(1)	none	none	yes
(2)	corruption	none	no
(3)	corruption	none	yes
(4)	corruption &loss	none	yes
(5)	corruption & loss	corruption & loss	yes

Table 1: Description of network channel types.

For each channel, specify which of the mechanisms in Table 2 are necessary for providing reliable data delivery (no data loss and no data duplication).

You have the following constraints:

- Specify a mechanism only if it is necessary. E.g., if one can provide reliable data delivery over a given channel without checksums, you should not specify checksums for that channel.
- Do not specify timeout-based retransmissions if NACK-based retransmissions are sufficient.

Fill in Table 2 by writing "X" in a corresponding cell to select each necessary mechanism. Then justify your choice for each channel by explaining why the mechanisms you picked are necessary and sufficient.

channel ID	sequence numbers	checksums	NACK-based retransmissions	timeout-based retransmissions
(1)				
(2)				
(3)				
(4)				
(5)				

Table 2: Reliability mechanisms.