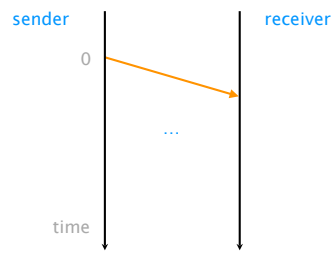# Communication Networks

Exercises week 4 – Reliable Transport

Reliable versus Unreliable Transport

In the lecture, you have learned how a reliable transport protocol can be built on top of a best-effort delivery network. However, some applications still use an unreliable transport protocol.

a) What are the characteristics of best-effort and of reliable transport?

b) What could be advantages of using an unreliable transport protocol?

c) What type of applications are suitable to use unreliable transport protocols?

d) As we will later see, the User Datagram Protocol (UDP) only provides unreliable transport. Assume you are forced to use a network which only supports UDP as a transport protocol. You must transmit an important document which eventually should be correctly transmitted. Do you see a way to implement some of the reliable transport mechanisms despite using UDP?

Reliable Transport (Exam Style Question)

sender   receiver

0

...

time

How long would a transfer take?

Consider a Go-Back-N sender and receiver directly connected by a 10 Mbps link with a propagation delay of 100 milliseconds. The retransmission timer is set to 3 seconds and the window has a length of 4 segments.

Draw a time-sequence diagram (see left) showing the transmission of 10 segments (each segment contains 10000 bits). An ACK is transmitted as soon as the last bit of the corresponding data segment is received and the size of the ACK is very small.

a) Draw the time-sequence diagram for the case where there are no losses.

b) Draw the time-sequence diagram for the case where the 3rd and last segments are lost once.

Reliable Transport[2] (Exam Style Question)

On the next page you see the beginning of a communication between two end-points using 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 unacknowledged 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.

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

Drop **first** data segment 3 and **all** data segments 5

GBN - Receiver

Received data segment	ACK number to send	Out-of-order buffer
0	1	[-]
1	2	[-]