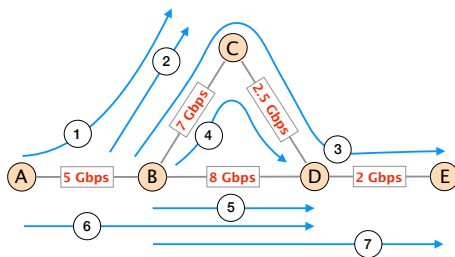


## Communication Networks

Prof. Laurent Vanbever

### Solution: Exercise 4 – Transport Concepts II

#### 4.1 Fairness



A network with shared links and 7 flows.

Consider the network on the left consisting of 5 nodes (A to E). Each link has a maximal bandwidth indicated in red. 7 flows (1 to 7) are using the network at the same time. You can assume that they have to send a lot of traffic and will use whatever bandwidth they will get. Apply the max-min fair allocation algorithm discussed in the lecture to find a fair bandwidth allocation for each flow. You can use the table below. In the top row, indicate which link is the current bottleneck. The other rows contain the corresponding bandwidth distribution for each flow.

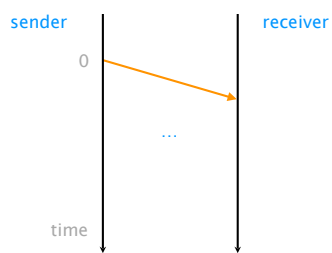
#### Solution:

Bottleneck link	D-E	C-D	B-C	A-B	B-D
Flow 1 A - B - C	1	1.5	2.25		
Flow 2 B - C	1	1.5	2.25		
Flow 3 B - C - D - E	1				
Flow 4 B - C - D	1	1.5			
Flow 5 B - D	1	1.5	2.25	2.75	4.25
Flow 6 A - B - D	1	1.5	2.25	2.75	
Flow 7 B - D - E	1				

## 4.2 Reliable Transport (Exam Style Question)

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.

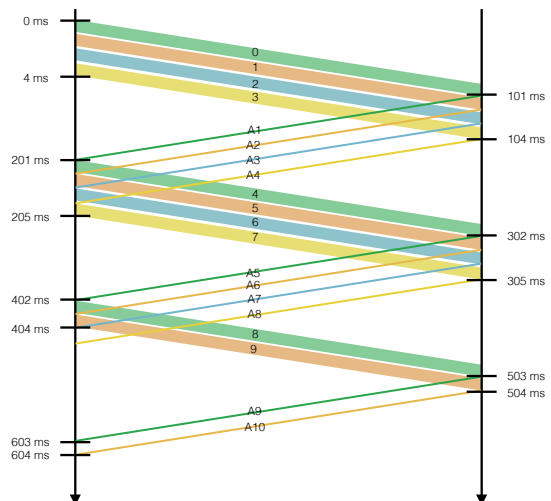
Draw a time-sequence diagram (see left) showing the transmission of 10 segments (each segment contains 10 000 bits). An ACK is transmitted as soon as the last bit of the corresponding data segment is received. The size of an ACK is very small, that means they have a negligible transmission delay.



How long would a transfer take?

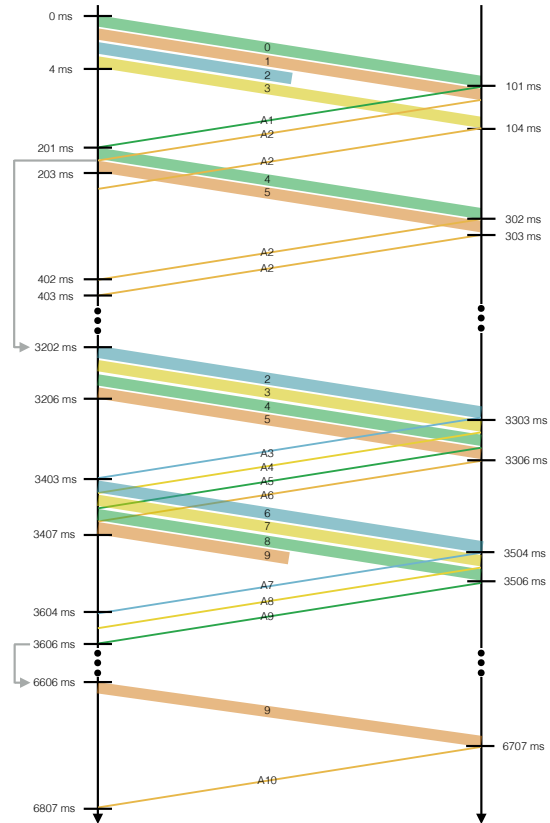
- a) Draw the time-sequence diagram for the case where there are no losses.

**Solution:** The acknowledgments always point to the next expected sequence number and not to the sequence number of the last received segment. This means that, for example, the segment with sequence number 5 is acknowledged with A6.



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

**Solution:**



### 4.3 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.

**Solution:**

- Best case: 11 segments, the last segment is dropped.
- Worst case: 14 segments, e.g., the second segment is dropped. GBN will retransmit all packets in the current window.

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 has the sender to transmit in the best (resp. worst) case and which ACK was lost?

**Solution:**

- Best case: 10 segments, e.g., the ACK for segment 5 is lost. Since GBN uses cumulative ACKs, the ACK for segment 6 implicitly also acknowledges segment 5.
- Worst case: 11 segments, the ACK for the very last segment is 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)?

**Solution:** To send three ACKs with number 4, the receiver also has to receive three data segments. That is for example possible if it received segments 5, 6, 7, but not segment 4 (otherwise, not all ACKs would have number 4). As out-of-order segments are saved in a buffer, retransmitting segment 4 is enough. However, if data segments or ACKs are duplicated in the network, we can no longer be sure that data segment 4 was the only missing segment.