CS-202 Exercises on Transport Layer: TCP (L14)

In today's exercise session, we will learn how the TCP protocol at the transport layer operates. More specifically, we will learn how to:

- compute the "file transfer time", that is the 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.
- plot a sequence diagram with all events and actions taken by the transport layers of a sender and a receiver while using TCP.
- deduce connection properties from TCP's congestion window

plots. All problems are paper and pencil exercises.

Exercise 1: file transfer time computation

So far, we have learned how to calculate the time it takes for a sender to transfer N packets to the receiver under different network conditions.

Today, we will practice how to calculate the file transfer time while using the TCP protocol (or a variant of TCP) at the transport layer.

The number of segments needed to transfer a file is not given, and have to be deduced from TCP's *MSS*. Also, the file's segments are not sent all at once but in rounds according to TCP's sender window. Which means, the time spent by the sender waiting for the receiver's ACK before sending the next segments is included in the time computation.
(Note: unless otherwise stated, assume all segment headers and all ACK segments have insignificant size.)

A) File transfer time calculation

Assume we have a simple version of TCP were:

- Congestion control and flow control are disabled.
- Sender window size is fixed to 4 *MSS* bytes.
- Retransmission timeout is $T \gg RTT$ secs.
- The receiver still stores (and does not discard) all out-of-order segments.

End-host *A* sends end-host *B* a file of size $8 \cdot MSS$ bytes.

A and *B* are directly connected over a channel with:

- transmission rate *R* bytes/sec
- propagation delay $d_{prop} > \frac{4.MSS}{R}$ *secs.*

How long will it take until *B* receives the entire file (i.e., B receives the last byte of the file) in each of the following scenarios?
Note: Assume that the packet headers (for all layers) have negligible size and that the connection is already established.

1. No segments are lost.
2. The 5*th* segment is lost and Fast-Retransmit is disabled.
3. The 5*th* segment is lost and Fast-Retransmit is enabled.

B) File transfer over TCP

Consider the topology shown in Figure 1. Host *A* opens a TCP connection to host *B*, and starts sending a file of size $F = 10$ bytes, in segments of size $MSS = 1$byte each. As a result of a faulty link, the 5-th packet (without counting the SYN packet in the TCP handshake) transmitted by *A* is lost.

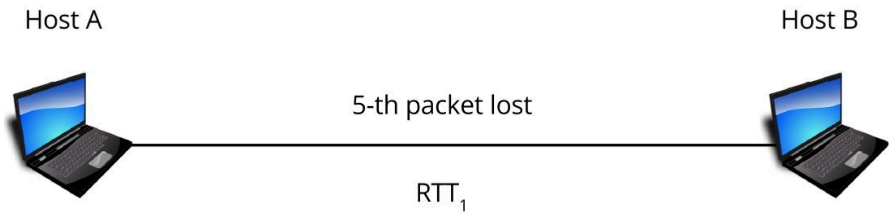

Figure 1: Network topology.

In this problem, make the following assumptions:

- The transmission delay for packets is negligible.
- The round-trip time between *A* and *B* is RTT_1.
- The retransmission timer of host *A* has a fixed duration equal to $2 * RTT$.
- TCP has Fast Retransmit disabled.
- A TCP receiver sends an ACK for each packet it receives.
- The first segment that *A* transmits will have a sequence number of 1.
- B stores (does not discard) all out-of-order packets.

Answer the following questions:

1. Complete the sequence diagram in Figure 2 with all packets exchanged between *A* and *B* *(we have completed part of the diagram to help you get started)*.
2. Calculate how much time it takes for *B* to finish receiving the file. *(Note: The one-way propagation delay from A to B is* $\frac{RTT1}{2}$)

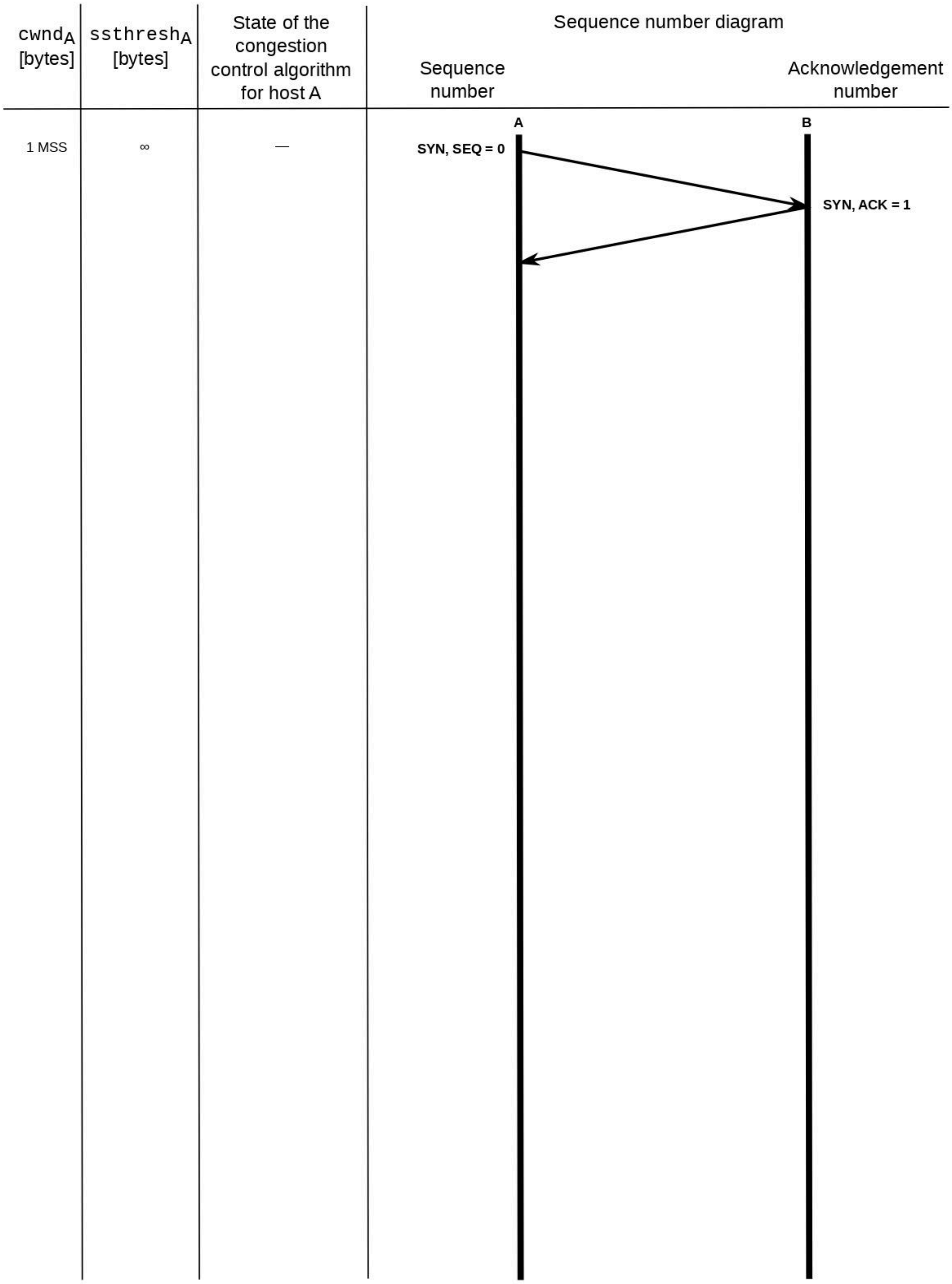

Figure 2: Sequence diagram of packet exchanged.

D) Adding a proxy in-between

Now, assume A uses node P, which runs an application-layer proxy to transmit the file to B, as shown in Figure 3.

When P receives a connection request from A, it connects a TCP socket with B. After that, the proxy application receives data from the TCP socket connected to A (the input socket), and writes data out to the TCP socket connected to B (the output socket). P forwards these packets to the output socket, the moment it can read them from the input socket. The proxy's operations do not incur any processing delay.

P is located exactly in the middle of the path between A and B, such that the round-trip times between A and P, and between P and B are both equal to $RTT_2 = \frac{RTT1}{2}$.

The faulty link described previously is now located on the part of the path between P and B (the second half of the path). As a result, the 5-th packet transmitted on that part of the path is lost. No packet loss occurs on the part of the path between A and P.

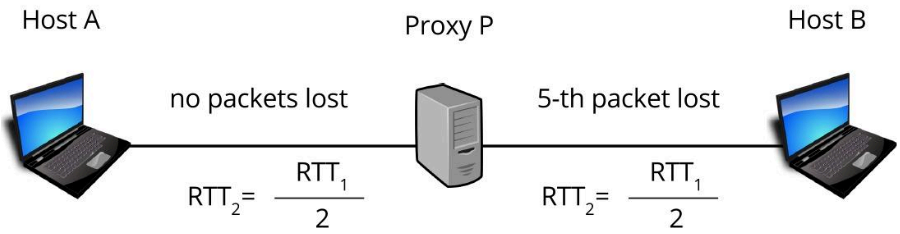

Figure 3: Network topology after a proxy.

3. Calculate the time it takes for the file transfer to be completed in this new setting. *(Note: Do not forget to adjust the timeout interval for the two TCP flows; from A to P, and from P to B. The timeout interval for the two flows is equal to* $2 \times RTT_2 = RTT_1$*)*
4. Does the introduction of the application-layer proxy in the previous part improve or worsen the file transfer? Which features of TCP are responsible for this?

Exercise 2: the effect of Routing on TCP

Consider the topology shown in Figure 4. End-host *A* sends end-host *B* a large file.

A's traffic is the only active flow in the network and all the traffic traverses links L_1, L_2, L_3 and L_4. *A* sends application data into its TCP socket at a rate of 80 Mbps. *B* can read data from its TCP socket at a rate of 30 Mbps.

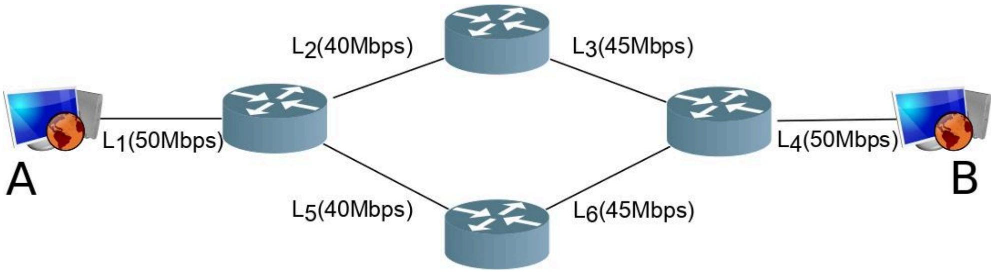

Figure 4: Network Topology with Links' Bandwidth Capacity.

1. What is the maximum transfer rate for the TCP flow in the following scenarios? Which aspect of TCP limits it?
   - The TCP receive buffer at *B* can hold only a small portion of the file.
   - The TCP receive buffer at *B* can hold the entire file.

Now, Assume that the TCP receive buffer at *B* can hold only a small portion of the file. During the file transfer Link L_2 fails and the traffic between *A* and *B* is rerouted via links L_1, L_5, L_6 and L_4. The RTT between *A* and *B* increases from RTT_{orig} to $RTT_{new} = 5 \cdot RTT_{orig}$.

2. Does *A* and *B* need to establish a new TCP connection? If yes, describe the message exchange.

3. Will the path change affect the TCP flow control?

4. What effect will the path change have on the TCP traffic between *A* and *B*?

Exercise 3: reading congestion window plot

Consider the graph shown in Figure 5, which plots the window size of a TCP sender as a function of time.

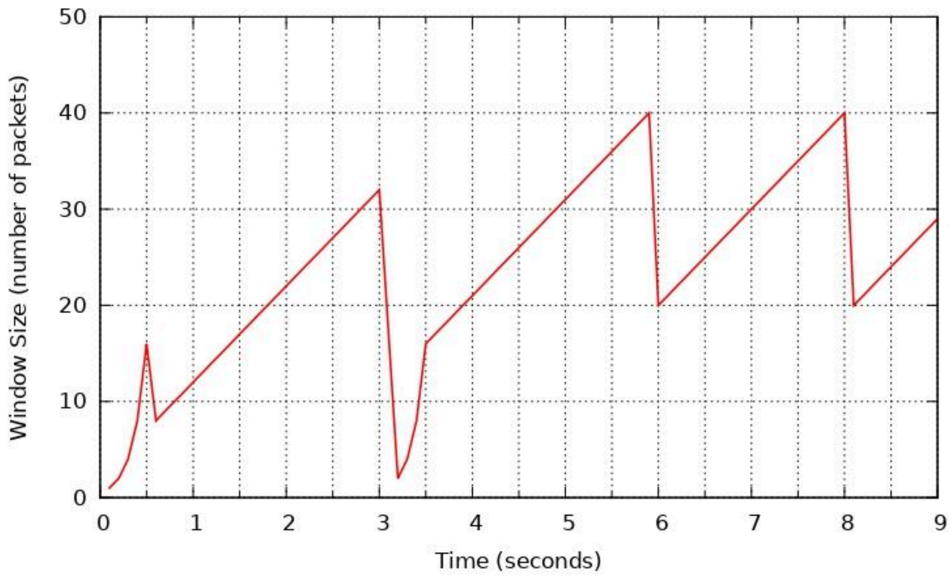

Figure 5: Congestion window size over time.

1. Identify what happens to the congestion window at the following times: (i) t = 0.5 secs, (ii) t = 3 secs, (iii) t = 3.5 secs, and (iv) t = 8 secs.

   For each case, you should: a) describe the state transition (previous state and next state), b) identify the event that caused it, and c) explain how we can conclude that from the graph.

   Example: *at t = 2 secs the sender transitions from state u to state v because event x occurred. We can see that event x has occurred there, because the window size changes from y to z.*

2. Calculate the number of packets that the TCP sender transmits between t = 6 secs and t = 8 secs.

3. Calculate the RTT of the TCP flow.

Exercise 4: a recap

Alice has opened a persistent TCP connection to Bob. At time T_0, Alice starts sending to Bob, over this connection, a file of size 12 bytes in segments of MSS = 1 byte.

Figure 6 shows how the congestion window of Alice, *cwnd*, changes over time after T_0 and until the file transfer completes. Each of the *five* points in the graph shows the time a change in *cwnd* took place and *cwnd*'s value after the change.

Make the following assumptions:

- Transmission delay is negligible.
- Bob sends an ACK for each segment it receives.
- The first segment that Alice transmits after T_0 has sequence number 10.
- Fast-retransmit is disabled.
- Only one segment gets lost after T_0, and it is a segment sent by Alice.
- B stores (does not discard) all out-of-order packets.

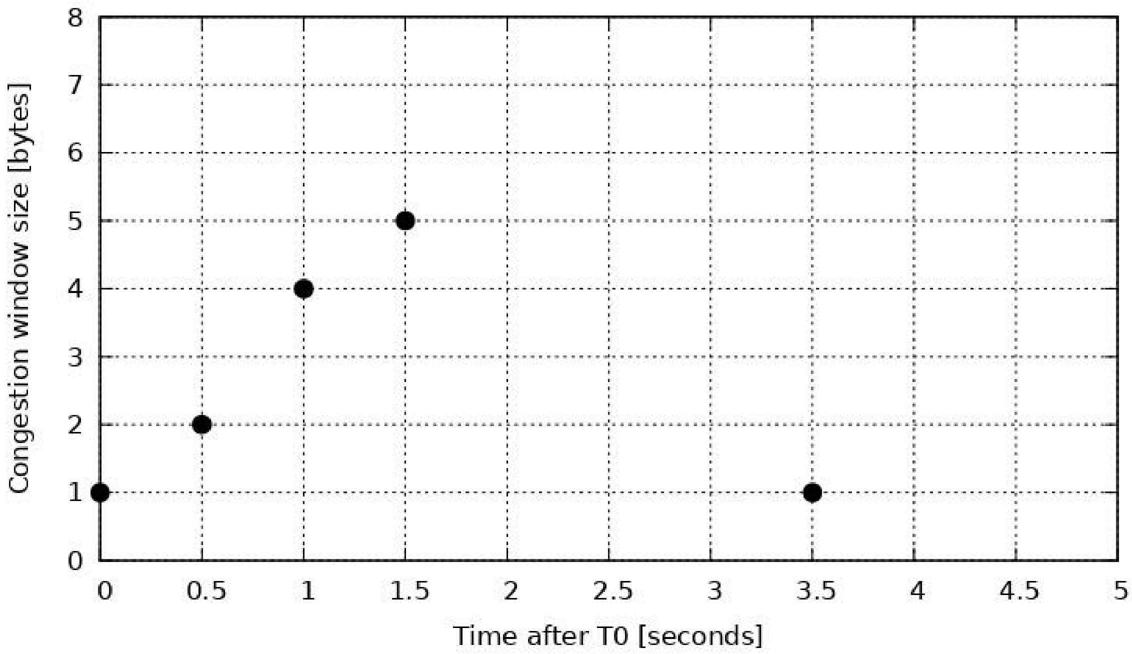

Figure 6: Congestion window of Alice over time

Based on the congestion window plot of Figure 6, answer the following questions:

1. What is the RTT between Alice and Bob?
2. What is the retransmission timeout used by Alice?
3. What was the size of Alice's congestion window, *cwnd*, the last time a packet was lost before T_0?
4. Complete the diagram in Figure 7 that shows what happens after T_0 and until the file transfer completes. In the diagram, show:
   - All segments exchanged between Alice and Bob.
   - The sequence numbers sent by Alice and the acknowledgment numbers sent by Bob.
   - The state of Alice's congestion-control algorithm.
   - The size of Alice's congestion window, *cwnd*, in bytes.
   - The value of Alice's slow-start threshold, *ssthresh*, in bytes.
5. How long does the file transfer take? Assume that the file transfer completes once Alice has received the final ACK for file data.
6. Now assume (just for this part) that fast-retransmit is enabled. Does this change the duration of the file transfer and how/why?

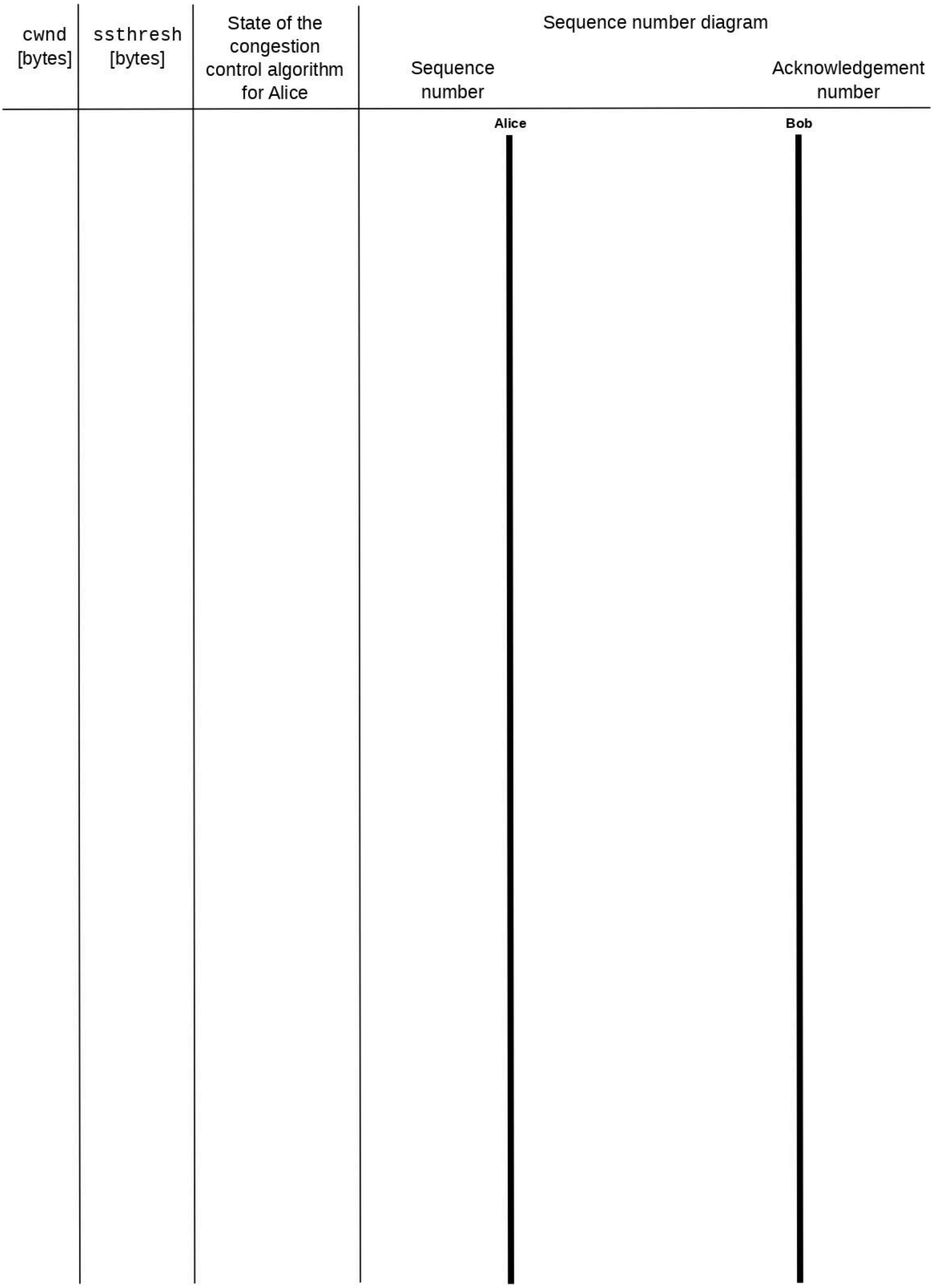

Figure 7: Sequence diagram.

Exercises to challenge yourself

Thinking creatively about TCP

Suppose that each router in the network has infinite buffer space which can hold all packets the router has to forward, so that no packet is ever dropped. Given this setting:

- What happens when a network link becomes congested.
- Describe how individual TCP flows will behave.
- Propose a modification for TCP, which will improve its behavior.

A variant of TCP

Consider a variant of TCP which flow control is disabled. Is the congestion control in this case sufficient to control the transmission rate of a sender if it is overwhelming a receiver (i.e., when the receiver has no more buffer space)? Justify your answers.

Finding a security loophole

In Figure 9, Alice is sending a large file to Bob using TCP. Denis tries to disrupt their communication by sending traffic to Céline. No other hosts send any traffic

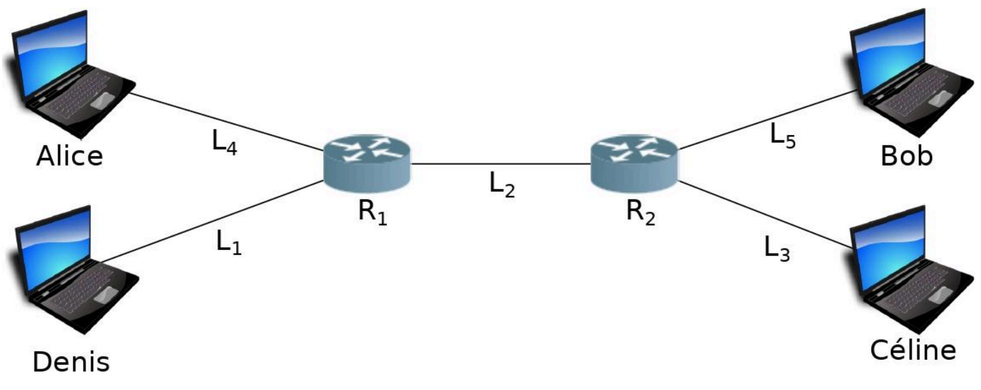

Figure 9: Network topology.

- Describe the simplest attack strategy that achieves Denis's goal. What condition needs to hold for the transfer rates of the links such that this strategy works?

- How will the TCP connection between Alice and Bob be affected by this attack? Draw a simple diagram that shows how Alice's congestion window, *cwnd*, evolves over time during the attack. You do not need to provide specific time values on the *x*-axis, just show the trend (e.g., does *cwnd* increase monotonically?)

- Describe the attack strategy that achieves Denis's goal while minimizing the amount of traffic that Denis sends to Céline.
  Hint: Denis does not need to send traffic at a constant rate.