Exercise Session 7: TCP (Transmission Control Protocol)

COM-208: Computer Networks

File transfer time calculation

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 and propagation delay $d_{prop} > \frac{4 \cdot MSS}{R}$ secs.
A and B use a modified version of TCP were:

i Congestion control and flow control are disabled.

ii Sender window size is fixed to $4MSS$ bytes.

iii Retransmission timeout is $T \gg RTT$ secs.

Assume that B stores (does not discard) all out-of-order segments.

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.

- No segments are lost.
- The $5th$ segment is lost and Fast-Retransmit is disabled.
- The $5th$ segment is lost and Fast-Retransmit is enabled.

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.

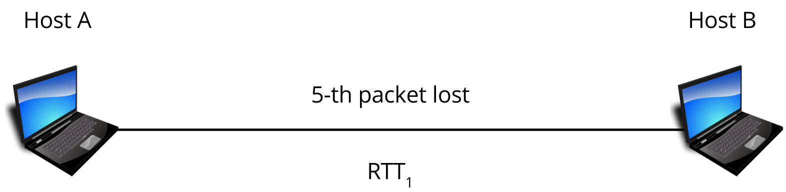

Figure 1: Network topology.

For the following questions, make the following assumptions:

i The transmission delay for packets is negligible.

ii The round-trip time between A and B is RTT_1.

iii The retransmission timer of host A has a fixed duration equal to $2 * RTT$.

iv TCP has Fast Retransmit disabled.

v A TCP receiver sends an ACK for each packet it receives.

vi The first segment that A transmits will have a sequence number of 1.

vii B stores (does not discard) all out-of-order packets.

- 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)*.

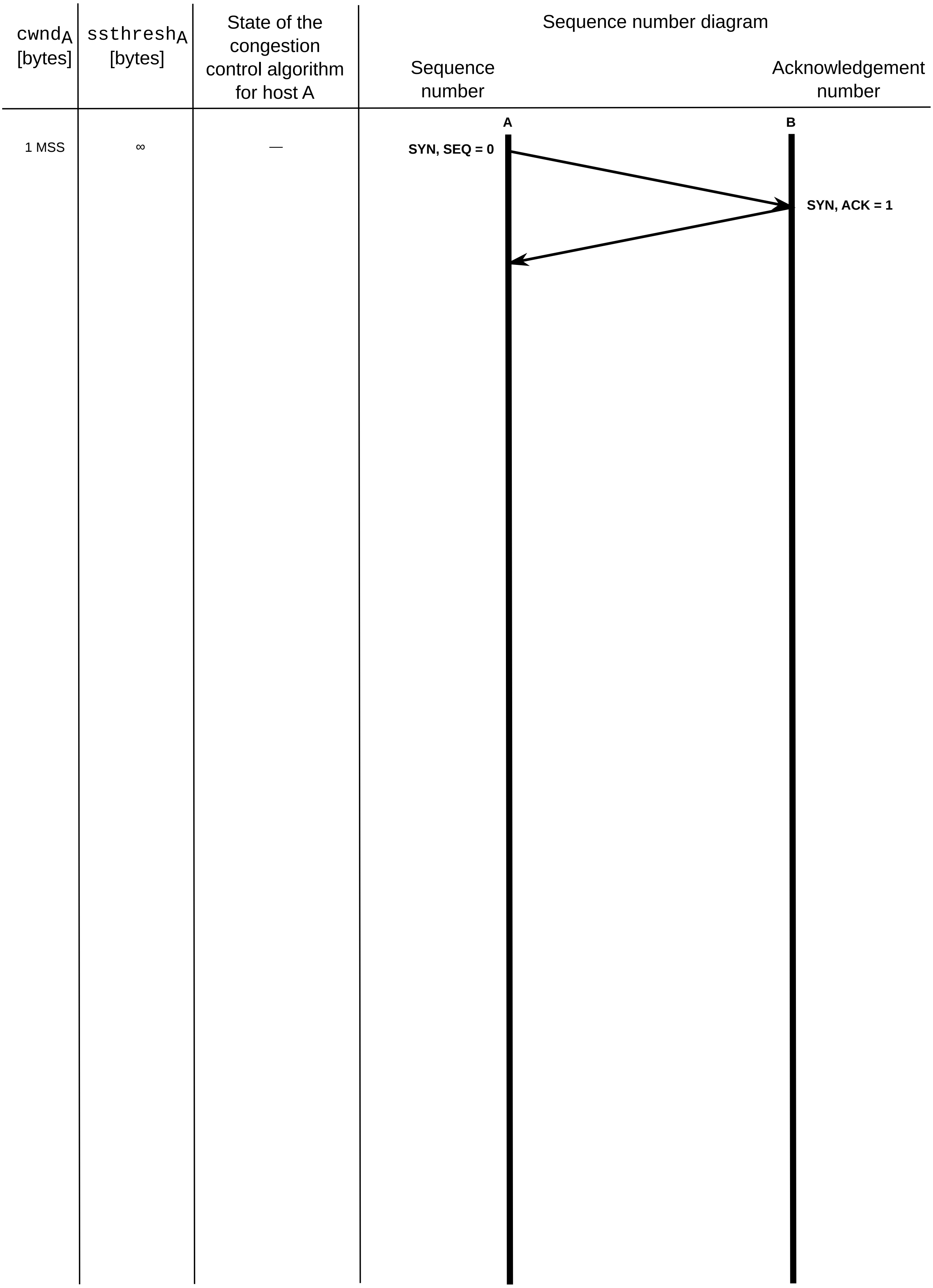

Figure 2: Sequence diagram of packets exchanged.

- Calculate how much time it takes for B to finish receiving the file.
  (Note: The one-way propagation delay from A to B is $\frac{RTT_1}{2}$)

Now, assume A uses node P, which runs an application-layer proxy to transmit the file to B, as shown in Figure 4.

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 $RTT_2 = \frac{RTT_1}{2}$.

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.

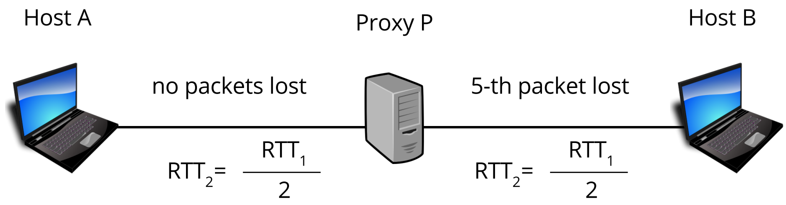

Figure 4: Network topology after adding a proxy.

- 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 $2 \times RTT_2 = RTT_1$)*
- 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?

Now with routing

Consider the topology shown in Figure 5. End-host A sends end-host B a large file. This 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.

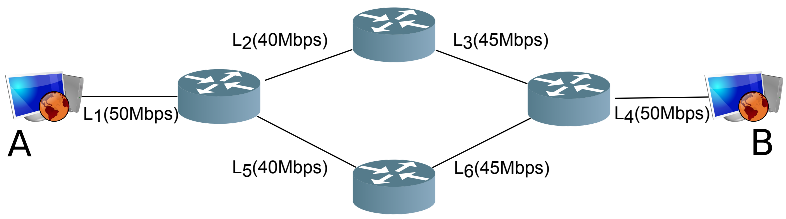

Figure 5: Network Topology with Links' Bandwidth Capacity.

- What is the max 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 $RTT_{new} = 5 \cdot RTT_{orig}$.

- Do A and B need to establish a new TCP connection? If yes, describe the message exchange.
- Will the path change affect the TCP flow control?
- What effect will the path change have on the TCP traffic between A and B?

TCP fairness

Consider the topology shown in Figure 6. Links A-R, B-R and C-R have identical characteristics, and have a higher transmission rate than link R-D. Transmitting hosts A, B and C all have to share the same link to send data to host D.
(Note: Assume that TCP is fair, in the sense that all TCP connections sharing the same bottleneck link will equally share the link's bandwidth.)

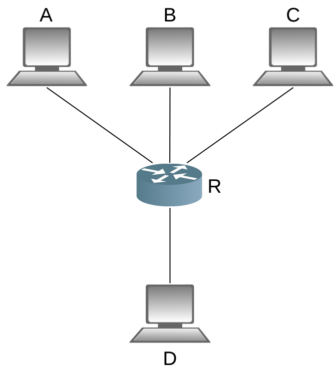

Figure 6: Network Topology

Calculate the share of the link capacity of R-D that each host gets (a long time after the connections have been established) in each of the following cases.
You should justify your answers.

- A, B and C are transmitting data. Each host uses a single TCP flow.
- Only A and B are transmitting data. A uses only one TCP flow, and B uses two parallel TCP flows.
- A, B and C are transmitting data. A and B each use one TCP flow. C uses a UDP-based application which transmits data at a constant rate of 40% of the transmission rate of link R-D.

Multiple TCP flows

Two flows, $flow_1$ (from A to C) and $flow_2$ (from B to C) traverse the same bottleneck link, R-C, as shown in Figure 7. The end-hosts use a variant of TCP where the congestion window is fixed to 60 packets (i.e., the window does not adapt to network conditions). You know that the capacity of link R-C is 3100 packets/sec.

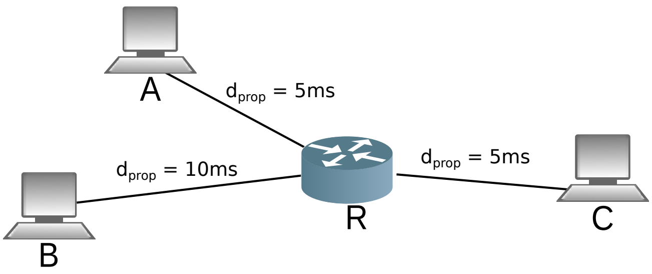

Figure 7: Network Topology.

- Suppose that the transmission delay is negligible. Compute the throughput that $flow_1$ ($Tput_1$) and $flow_2$ ($Tput_2$) would achieve if only one flow is active at a time, and no packet is ever dropped. Compute ratio $\frac{Tput_1}{Tput_2}$.
- You run an `ns-2` simulation where both flows are active at the same time, and you get that $\frac{Tput_1}{Tput_2} = 1.28$. In your simulation, you have set the buffer size of router R to 1000 packets, so that no packet is ever dropped.

  Explain why the ratio between the throughput of the two flows is different from what you computed in the previous sub-question.

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 were 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.

Reading congestion window plot

Consider the graph shown in Figure 8, which plots the window size of a TCP sender as a function of time.

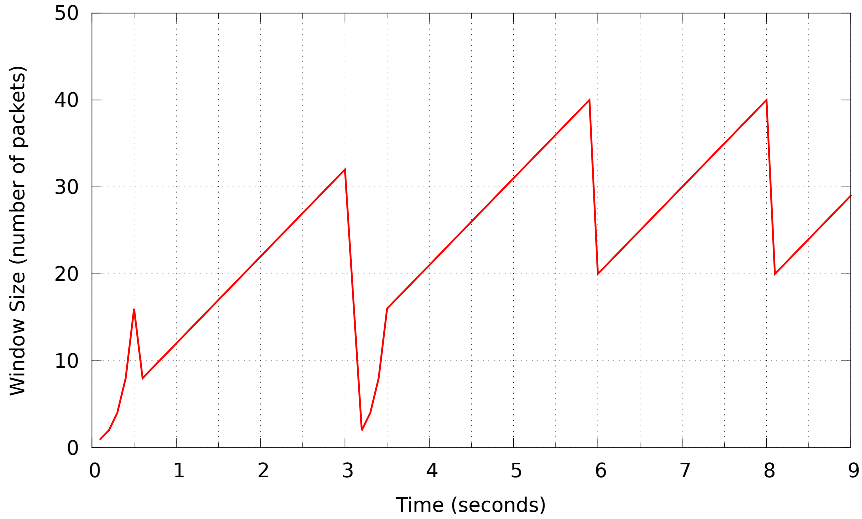

Figure 8: Congestion window size over time.

- 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: 1. describe the state transition (previous state and next state), 2. identify the event that caused it, and 3. 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.*

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

- Calculate the RTT of the TCP flow.

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.

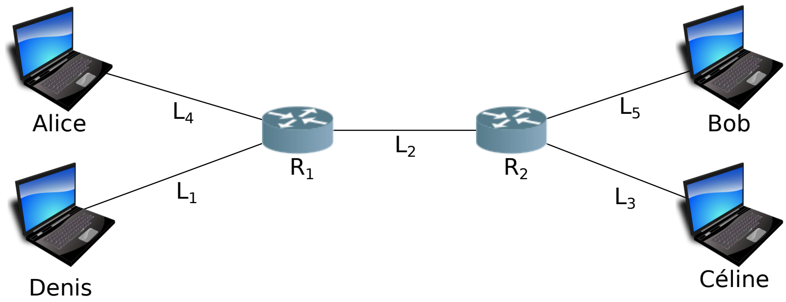

Figure 9: Network topology.

- 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.

A recap of everything

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 11 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.

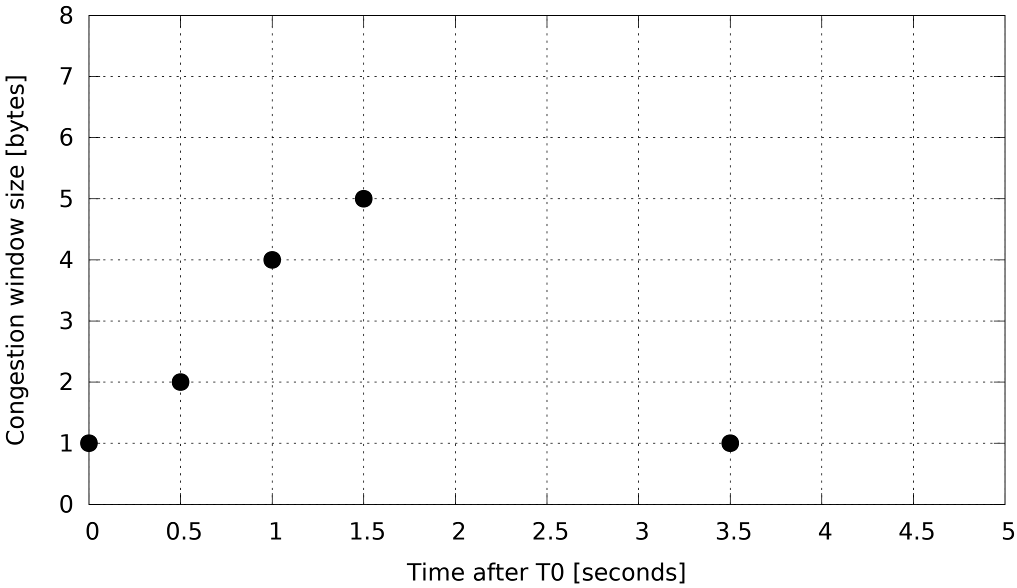

Figure 11: Congestion window of Alice over time

- What is the RTT between Alice and Bob?
- What is the retransmission timeout used by Alice?

- What was the size of Alice's congestion window, *cwnd*, the last time a packet was lost before T_0?
- Complete the diagram in Figure 12 that shows what happens after T_0 and until the file transfer completes:
  - 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.

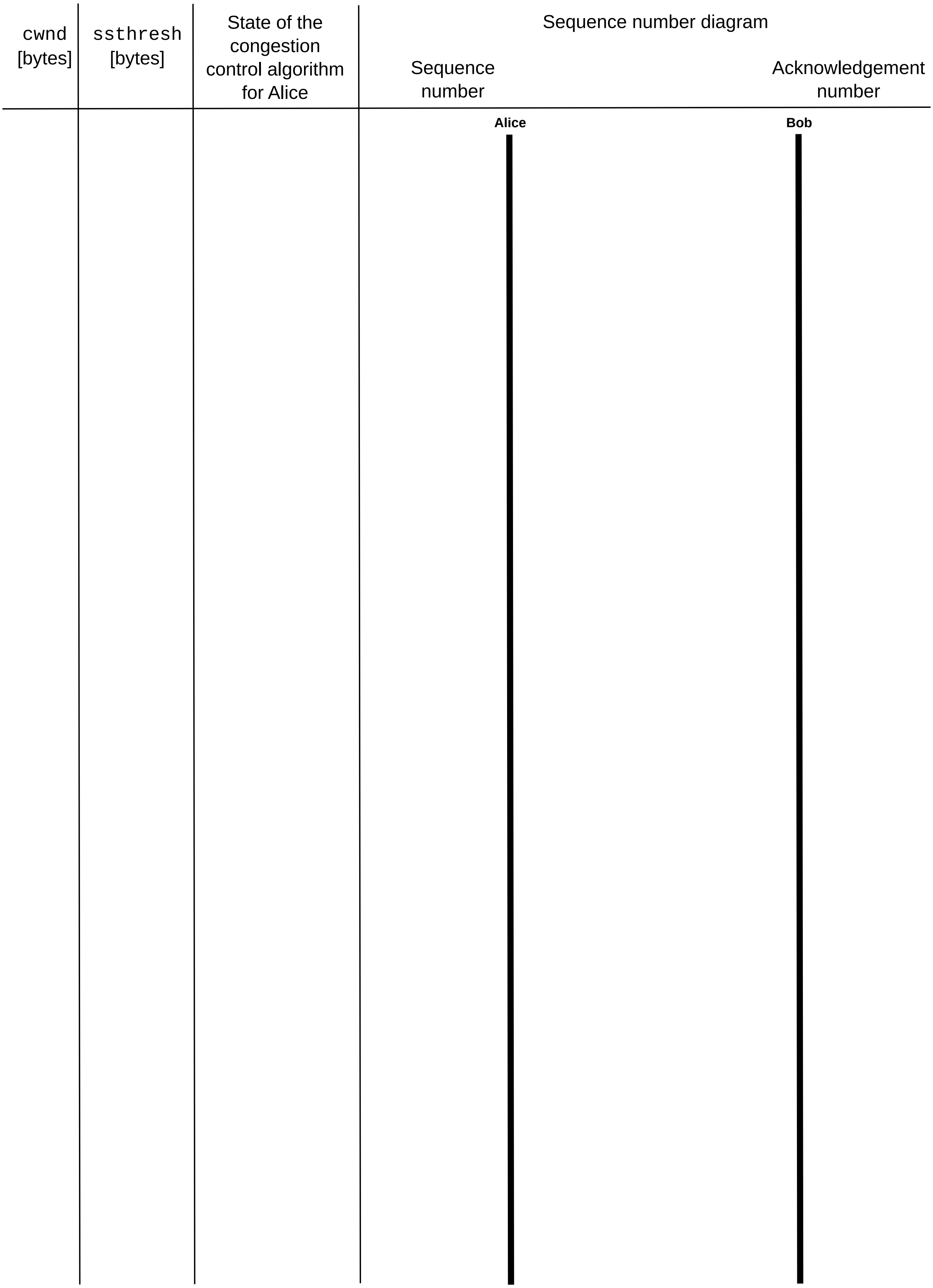

Figure 12: Sequence diagram.

- How long does the file transfer take? Assume that the file transfer completes once Alice has received the final ACK for file data.
- Now assume (just for this part) that fast-retransmit is enabled. Does this change the duration of the file transfer and how/why?