Computer Networks - Midterm

November 18, 2022

Duration: 2h15m

- This is a closed-book exam. You can use two A4 “cheat-sheets.”
- Write your answers clearly, in English or in French, using extra sheets if necessary.
- It consists of 3 problems. The total number of points is 50.
- This document contains 18 pages.

Good luck!

Last Name:
First Name:

SCIPER No:

(answers to the questions are shown in italic and blue) *(grades in red)*

1 Short questions (5 points)

For each question, circle a single best answer.

1. If you connect to the Internet through a 1 Gbps (downstream) DSL Internet connection, this means that:
   (a) You will always download data from the Internet at a rate (throughput) of 1 Gbps.
   (b) You will share physical links with other households from your neighbourhood.
   (c) Your Internet traffic will travel over your phone line. *(Correct)*
   (d) All of the above.

2. You open up a packet traversing a network link somewhere on the Internet. What might you find inside?
   (a) An HTTP header and message, encapsulated in a TCP header, encapsulated in an IP header, encapsulated in a link-layer header.
   (b) A DNS header and message, encapsulated in a UDP header, encapsulated in an IP header, encapsulated in a link-layer header.
   (c) A BitTorrent header and message, encapsulated in a TCP header, encapsulated in an IP header, encapsulated in a link-layer header.
   (d) Any of the above. *(Correct)*

3. Alice and Bob are connected over a network link of transmission rate R and propagation delay D. If you add a second link of the same type between Alice and Bob (such that they can exchange traffic in parallel over both links), you change:
   (a) The round-trip time (RTT) of a packet between Alice and Bob.
   (b) The maximum throughput achievable from Alice to Bob (and vice versa). *(Correct)*
   (c) Both of the above.
   (d) None of the above.

4. Consider a queue that feeds a network link. If you change the size of the queue, you may affect:
   (a) The queuing delay experienced by packets in the queue. *(Correct)*
   (b) The processing delay experienced by packets after they are read from the queue.
   (c) Both of the above.
   (d) None of the above.

5. Two packets, one of them 100 times the size of the other, traverse the same network path at about the same time, and they experience approximately the same end-to-end delay. You may conclude that the dominant delay factor for both packets was:
   (a) Propagation and/or queuing delay. *(Correct)*
   (b) Transmission delay.
   (c) Any of the above.
   (d) This scenario cannot happen.

6. A web browser downloads a file from a web server. The dominant delay factor is propagation delay. Could the web browser download the same file significantly faster through a proxy web server?

   (a) Yes, because downloading through a proxy web server is always faster.
   (b) Yes, because the dominant delay factor is propagation delay, which can always be avoided by downloading through a proxy.
   (c) No, because the dominant delay factor is propagation delay, which can never be avoided by downloading through a proxy.
   (d) It is possible, depending on the properties of the links between the web browser and the proxy web server, and between the proxy web server and the origin web server. *(Correct)*

7. Alice distributes a very large file to a small number of friends using the client/server approach. The resulting file distribution time (time for all friends to get the file) is X. Then, Alice distributes another file, of the same size, to the same friends, using the peer-to-peer approach we saw in class. The resulting file distribution time is approximately X. Assume network conditions never change. Which of the following would be a plausible explanation for the two approaches achieving approximately the same distribution time?

   (a) Alice's friends are located very far from her.
   (b) Alice's friends are located very close to her.
   (c) In both approaches, the bottleneck is Alice's Internet connection (in particular, her upload capacity).
   (d) In both approaches, the bottleneck is the Internet connection of one of Alice's friends (in particular, the friend's download capacity). *(Correct)*

8. Suppose a network guarantees that packets are never lost and are always delivered within reasonable time and in order (but may still be corrupted). Which of the following elements of reliable data delivery are unnecessary for this network?

   (a) Timeouts. *(Correct)*
   (b) Retransmissions.
   (c) Checksums.
   (d) All of the above.

9. How do we prevent SYN flooding attacks that target a TCP server's incomplete-connection buffer?

   (a) With web cookies.
   (b) With random sequence numbers.
   (c) By pushing the relevant state to the TCP client, hence removing the need for an incomplete-connection buffer. *(Correct)*
   (d) We cannot prevent it.

10. Suppose the network layer of the Internet guaranteed that packets were never reordered or duplicated. Which of the following changes to TCP would make sense?

   (a) A receiver would send selective, not cumulative ACKs.
   (b) A sender would retransmit multiple packets upon a timeout.
   (c) A sender would never do a fast-retransmit.
   (d) A sender would do a fast-retransmit after a single duplicate ACK. *(Correct)*

2 Web browsing + delays (28 points)

Consider the network in figure 1.

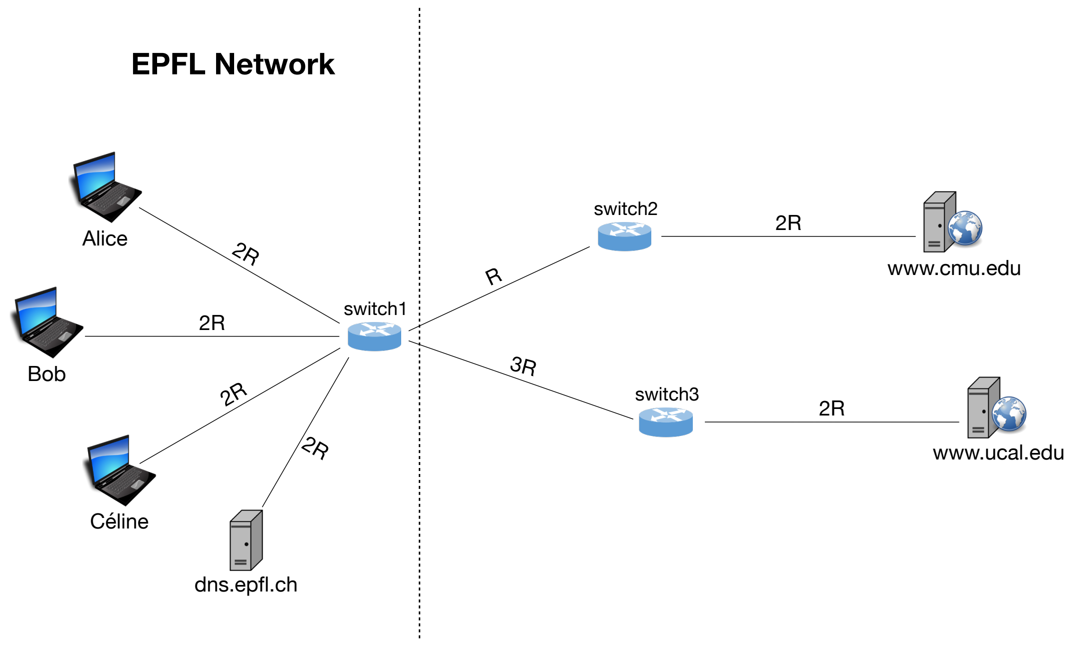


Figure 1: Network topology for Problem 2

- The maximum segment size is MSS.
- Each link has propagation delay D.
- For each link, the transmission rate has (in each direction) the value shown in the figure.
- **switch1**, **switch2**, and **switch3** are store-and-forward packet switches with 0 processing delay.
- All EPFL computers use **dns.epfl.ch** as their local DNS server. It performs recursive DNS requests.
- Web browsers and web servers communicate through persistent TCP connections (i.e., they reuse each established TCP connection to exchange as much traffic as possible).
- At the beginning of this problem, all caches (of all kinds) are empty, and there are no established TCP connections.
- All DNS records have time-to-live (TTL) 24 hours.
- For Questions 1 to 4, there is no other traffic on the Internet other than the traffic caused by Alice's and Bob's actions.
- There is no packet loss or corruption in this problem.

Packet sizes:

- DNS requests and responses, HTTP requests, and TCP connection-setup packets experience 0 transmission delay.
- All headers (including HTTP headers) are insignificant, i.e., you can assume they have size 0.
- Size of `www.cmu.edu/index.html`: MSS
- Size of `www.cmu.edu/logo.png`: $2MSS$
- Size of `www.ucal.edu/logo.png`: MSS

Question 1 (2 **points):**

Alice types in her web browser `http://www.cmu.edu/index.html` and presses enter. The requested web page references two embedded images, in this order: `http://www.cmu.edu/logo.png`, then `http://www.ucal.edu/logo.png`.

List all the sockets, on Alice's computer and on any server that she directly contacts, that send or receive packets as a result of Alice's action. For each socket, state on which computer this socket exists (e.g., "Alice's" or "www.cmu.edu") and the socket's type (UDP socket, TCP listening socket, or TCP connection socket). Assign a number to each socket, such that you can refer to it in subsequent questions.

Socket #	Computer	Type
1	Alice's computer	UDP socket
2	dns.epfl.ch	UDP socket
3	Alice's computer	TCP connection socket
4	www.cmu.edu	TCP listening socket
5	www.cmu.edu	TCP connection socket
6	Alice's computer	TCP connection socket
7	www.ucal.edu	TCP listening socket
8	www.ucal.edu	TCP connection socket

Question 2 (8 **points):**

List the packets that are sent or received by Alice's computer as a result of Alice's action. Include any TCP connection-setup packets, but ignore any packets that carry only TCP ACKs. For each packet, list which computer sends it (e.g., "Alice's") and which computer receives it (e.g., "www.cmu.edu"), which socket sends it and which socket receives it, and briefly its purpose (e.g., request for ...).

Be careful: In past midterms, students were asked to list the source and destination port number of each packet. This time I am asking you to list the socket that sends and receives each packet. You can list a socket using the number you assigned to it in Question 1. If you have no idea what I am talking about, just leave those columns blank and complete the rest of the table.

Packet #	Source	Destination	Source socket	Dest. socket	Purpose
1	alice	dns.epfl.ch	1	2	DNS request for www.cmu.edu's IP address
2	dns.epfl.ch	alice	2	1	DNS response
3	alice	www.cmu.edu	3	4	connection-setup request
4	www.cmu.edu	alice	5	3	connection-setup response
5	alice	www.cmu.edu	3	5	HTTP GET request for base file
6	www.cmu.edu	alice	5	3	HTTP GET response for base file
7	alice	www.cmu.edu	3	5	HTTP GET request for first logo
8	alice	dns.epfl.ch	1	2	DNS request for www.ucal.edu's IP address
9	dns.epfl.edu	alice	2	1	DNS response
10	alice	www.ucal.edu	6	7	connection-setup request
11	www.cmu.edu	alice	5	3	HTTP GET response for first logo, first packet
12	www.cmu.edu	alice	5	3	HTTP GET response for first logo, second packet
13	www.ucal.edu	alice	8	6	connection-setup response
14	alice	www.ucal.edu	6	8	HTTP GET request for second logo
15	www.ucal.edu	alice	8	6	HTTP GET response for second logo

Question 3 (3 **points):**

Soon after Alice views `http://www.cmu.edu/index.html`, Bob also types `http://www.cmu.edu/index.html` in his web browser and presses enter. How long does it take for Bob to obtain the base file for this web page? Justify your answer.

- DNS request + DNS response: $4D$
- TCP connection-setup request + response: $6D$
- HTTP GET request for base file: $3D$
- HTTP GET response: $3D + \frac{MSS}{2R} + \frac{MSS}{R} + \frac{MSS}{2R}$
- Total: sum of all

Question 4 (6 **points):**

How long does it take for Bob to obtain the two images?

Compute from the moment Bob's web browser has downloaded the base file. Assume that Bob's web browser operates efficiently, i.e., it sends out all HTTP GET requests for embedded objects the one after the other.

The correct answer involves many possible scenarios. Choose the one that simplifies your calculation the most and describe it clearly (i.e., state under which assumptions this scenario holds).

Let us first compute, for each image, the time it takes from the moment switch 1 transfers the GET request till the time it receives the last bit of the image.

For the HTTP GET request of `www.cmu.edu/logo.png`, it takes:

- HTTP GET request for first image: $2D$
- Last bit of HTTP GET response arrives at switch1: $2D + \frac{MSS}{2R} + \frac{2MSS}{R}$

As for the HTTP GET request of `www.ucal.edu/logo.png`, it takes:

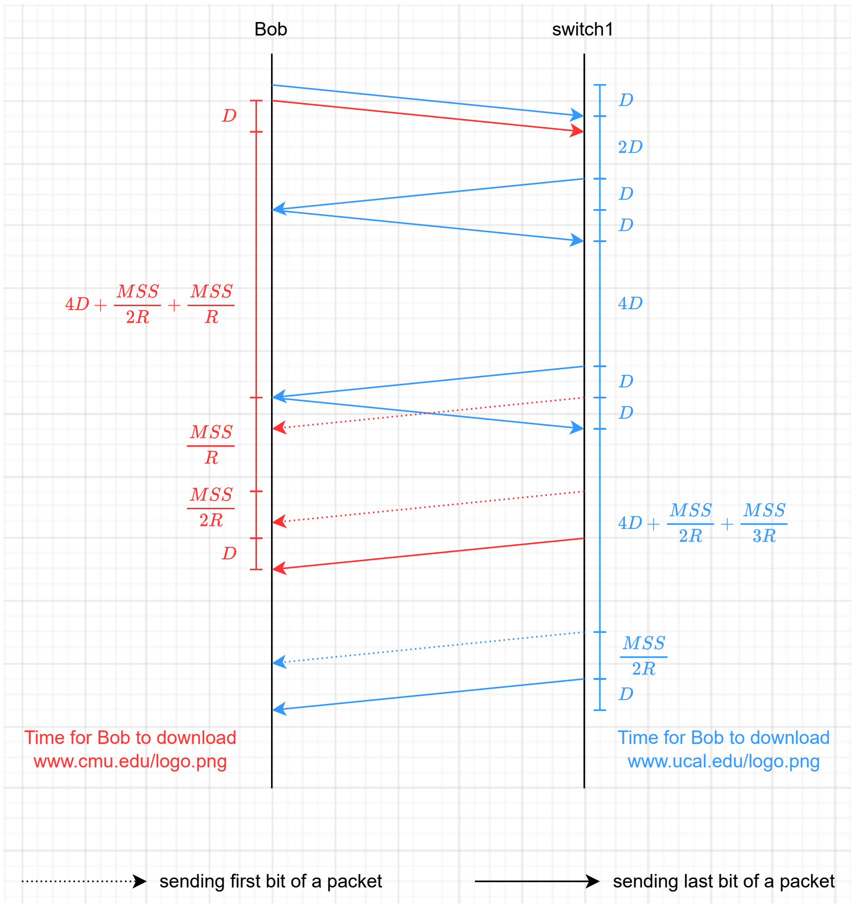
- HTTP GET request for second image: $2D$
- First bit of HTTP GET response arrives at switch1: $2D + \frac{MSS}{2R} + \frac{MSS}{3R}$

There are two possible cases for this question:

- **Case2**: Bob sends the GET request of `www.cmu.edu/logo.png` and the DNS for `www.ucal.edu` back to back (or as soon as possible).
- **Case1**: Bob send DNS + TCP connection for ucal first. And then, send both GET requests in parallel.

Case1: The simplest calculation, in this case, corresponds to the scenario where the last bit of the first image reaches switch1 (and gets sent) before the first bit of the second image.
In this scenario, the delay to get the second image masks the delay to get the first one. So, the total delay to get the two images is the delay to get the second image:

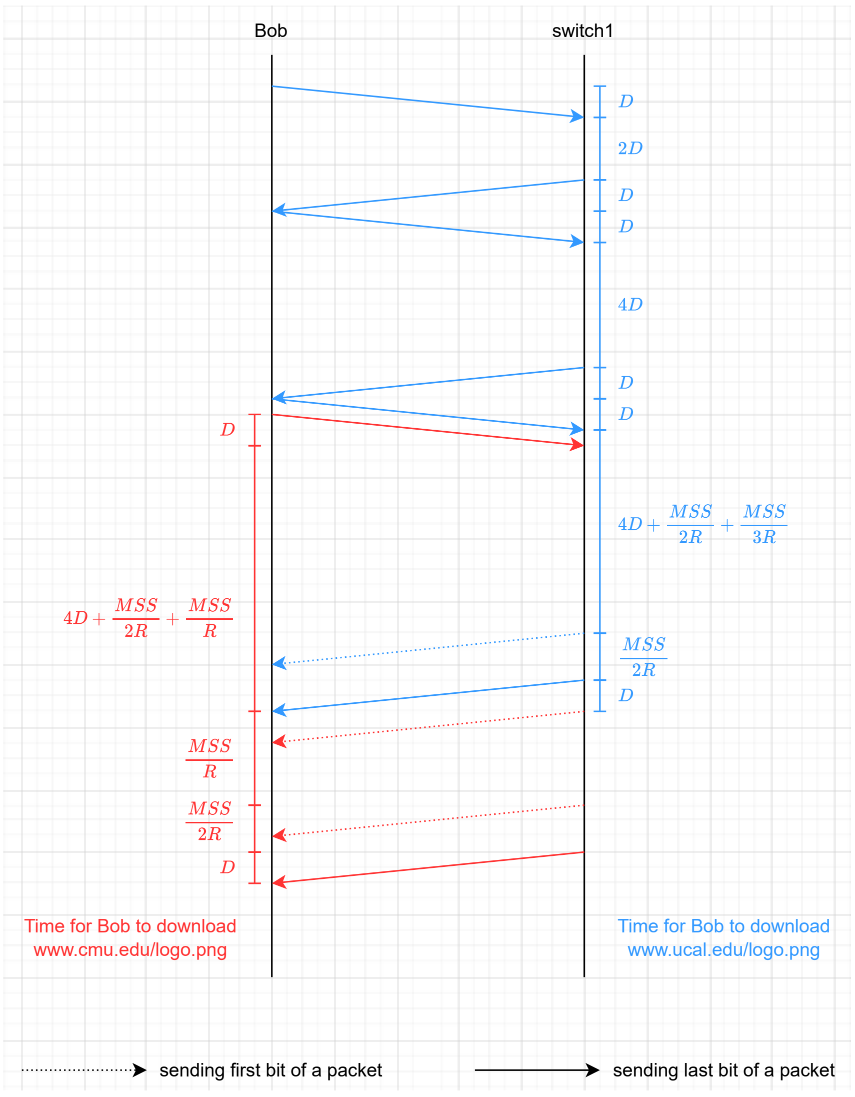
- DNS request + response: $4D$
- TCP connection-setup request + response: $6D$
- HTTP GET request for second image: $3D$
- HTTP GET response: $3D + \frac{MSS}{2R} + \frac{MSS}{3R} + \frac{MSS}{2R}$
- Total: sum of all.


Bob
switch1
D
$2D$
D
D
$4D$
D
D
$4D + \frac{MSS}{2R} + \frac{MSS}{3R}$
$\frac{MSS}{2R}$
D
D
$4D + \frac{MSS}{2R} + \frac{MSS}{R}$
$\frac{MSS}{R}$
$\frac{MSS}{2R}$
D
Time for Bob to download www.cmu.edu/logo.png
Time for Bob to download www.ucal.edu/logo.png
sending first bit of a packet
sending last bit of a packet

Problem 2, Question 4, Solution Case 1

Case2: In this case we need to check for queuing delay; we can not state any assumptions. After Bob sends both GET request, switch 1 finishes transferring the last bit of logo.png for ucal after: $2D + \frac{MSS}{5R} + \frac{MSS}{3R} + \frac{MSS}{2R} = 5D + 1.33\frac{MSS}{R}$. At the same time, switch 1 receives the first bit of logo.png for cmu after: $5D + \frac{MSS}{2R} + \frac{MSS}{R} = 5D + 1.5\frac{MSS}{R}$; which is greater than $5D + 1.33\frac{MSS}{R}$. Hence, there is no queuing delay.
So, the total delay to get the two images:

- DNS request + response: $4D$
- TCP connection-setup request + response: $6D$
- HTTP GET request for first image: $3D$
- HTTP GET response: $3D + \frac{MSS}{2R} + \frac{2MSS}{R} + \frac{MSS}{2R}$
- Total: sum of all.


Bob
switch1
D
$2D$
D
D
$4D$
D
D
$4D + \frac{MSS}{2R} + \frac{MSS}{3R}$
$\frac{MSS}{2R}$
D
D
$4D + \frac{MSS}{2R} + \frac{MSS}{R}$
$\frac{MSS}{R}$
$\frac{MSS}{2R}$
D
Time for Bob to download www.cmu.edu/logo.png
Time for Bob to download www.ucal.edu/logo.png
sending first bit of a packet
sending last bit of a packet

Problem 2, Question 4, Solution Case 2

Question 5 (6 **points):**

This question has nothing to do with Alice, Bob, or their communication.

Céline wants to know whether anyone from EPFL has accessed web page `www.berkeley.edu` within the last 24 hours. Is there a way for her to make an informed guess?

If you think there is, clearly describe the steps she will have to take.

If you think there isn't, try to explain why.

First, Céline sends to her local DNS server DNS requests for names that are unlikely to have been accessed by anyone at EPFL (*). She measures how long it takes for the local DNS server to respond to each request. Based on these measurements, she estimates the minimum time it takes, say t_1, for the local DNS server to respond to a request when it does not have the answer cached.

Then, she sends to her local DNS server DNS requests for the same names. She measures again how long it takes for the local DNS server to respond to each request. Based on these measurements, she estimates the time it takes, say t_2, for the local DNS server to respond to a request when it does have the answer cached.

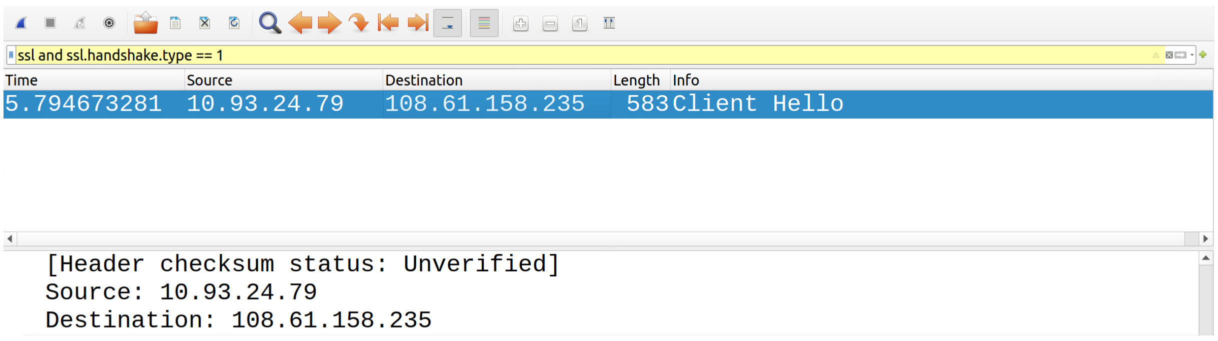
If t_1 is significantly larger than t_2, then Céline can make an informed guess as follows:

She sends to her local DNS server a DNS request for `www.berkeley.edu` and measures how long it takes for the server to respond. If the measurement is close to t_1, then she guesses that, very likely, nobody from EPFL accessed `www.berkeley.edu` within the last 24 hours.

(*) Ideally, the DNS server that acts as an authoritative DNS server for these names is the same as the DNS server that acts as an authoritative DNS server for `www.berkeley.edu`. So, ideally, these names are obscure (unlikely to have been accessed) DNS names from the `berkeley.edu` domain.

Question 6 (3 **points):**

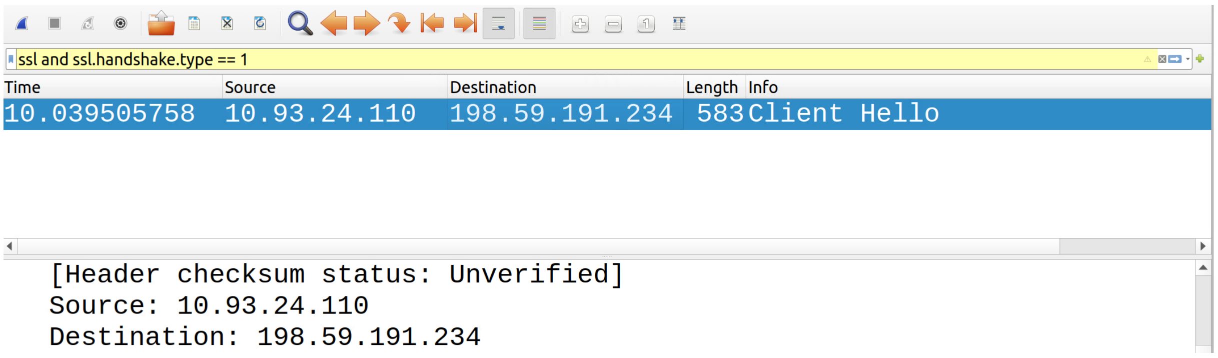
Alice and Bob are sitting in the same room, connected to the same network. They both type in their respective web browsers `https://nocss.club/` and press enter. Figure 2 and Figure 3 show snapshots of the traffic captured by Wireshark on Alice's and Bob's computers, respectively.



```
ssl and ssl.handshake.type == 1
Time          Source        Destination      Length  Info
5.794673281   10.93.24.79   108.61.158.235   583     Client Hello

    [Header checksum status: Unverified]
    Source: 10.93.24.79
    Destination: 108.61.158.235
```

Figure 2: Traffic captured by Wireshark (Alice's machine).



```
ssl and ssl.handshake.type == 1
Time           Source         Destination      Length  Info
10.039505758   10.93.24.110   198.59.191.234   583     Client Hello

    [Header checksum status: Unverified]
    Source: 10.93.24.110
    Destination: 198.59.191.234
```

Figure 3: Traffic captured by Wireshark (Bob's machine).

- What is the difference between the snapshot captured on Alice's computer and the snapshot captured on Bob's computer? What may explain this difference?

  The IP address Bob's computer connects to is different from the one seen in Alice's case. As both access the same website, there are two main reasons for this difference: first, Bob could be connecting to proxy web server; second, the DNS answer for `https://nocss.club/` was different for Alice and Bob, this could happen because of a stale mapping (a change in IP address between the two queries not being immediately reflected due to the TTL) or because of a cache poisoning attack.

3 Transport layer (17 **points**)

Consider the network in Figure 4.

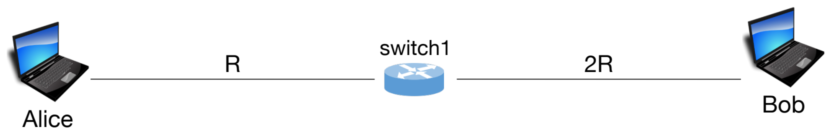


Figure 4: Network topology for Problem 3.

- The maximum segment size is MSS bytes.
- Each of Alice and Bob's send and receive buffers fits $1000MSS$ bytes.
- Each link has propagation delay D.
- For each link, the transmission rate has (in each direction) the value shown in the figure.
- **switch1** is a store-and-forward packet switch with 0 processing delay.

A process running on Alice's machine has established a TCP connection to a process running on Bob's machine and is sending data to it. Alice's process has an infinite amount of data to send to Bob's process, while Bob's process does not have any data to send. There is no traffic on this network other than the traffic exchanged between these two processes.

Question 1 (2 **points):**

If there is never any packet loss or corruption, what is the maximum number of unacknowledged bytes that Alice can send (before she starts receiving Bob's acknowledgments)? Justify your answer.

It is the bandwidth-delay product: $R \times 4D$. This is the maximum number of bytes that Alice can send before she starts receiving Bob's acknowledgments.

Question 2 (1 **point):**

What is the maximum receiver window that Bob may advertize to Alice? Justify your answer.

The size of its receive buffer: $1000 MSS$.

Question 3 (3 **points):**

Alice has been sending data to Bob for a very long time. There has been no packet loss, no corruption, and no unexpected delay for a very long time. What is, approximately, the average throughput from Alice's process to Bob's process after this long time? Justify your answer.

Since there has been no loss or corruption, the congestion window has kept increasing.

Hence, average throughput is limited either by link capacity R, or by Bob's receiver buffer size (flow control). In the latter case, Alice sends 1000MSS every RTT.

Average throughput is approximately $min(R, \frac{1000MSS}{4D})$.

Question 4 (4 **points):**

After this long period of no loss, corruption, or unexpected delay, during a period of $4D$ time units, 999 of the packets exchanged between Alice and Bob get dropped. If you could choose which packets get dropped, so as to minimize the disruption to Alice and Bob's communication, which 999 packets would you choose? Justify your answer.

999 consecutive ACKs from Bob to Alice. Since TCP ACKs are cumulative, the 1000th ACK will subsume the 999 lost ACKs that were sent before it.

Question 5 (4 **points):**

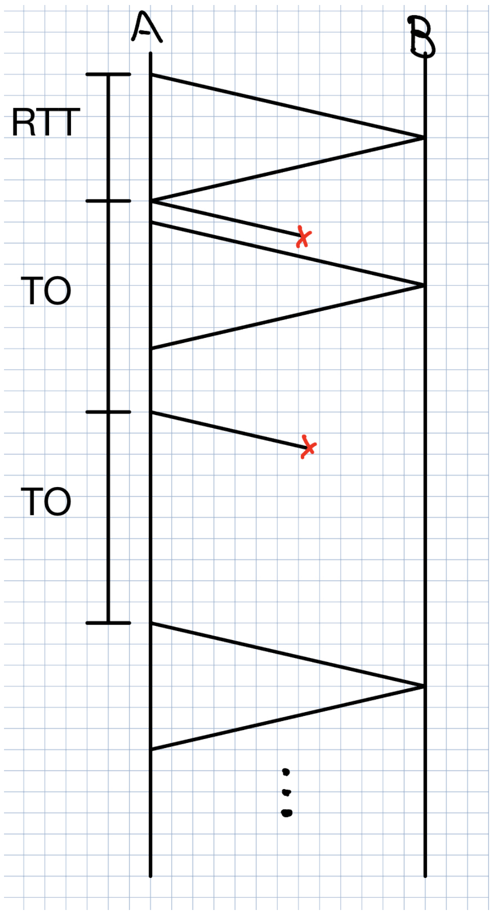
Ignore the events of Question 4. Assume you are at the end of Question 3.

After this long period of no loss, corruption, or unexpected delay, during a period of $4D$ time units, all packets from Alice to Bob get dropped. This big loss event ends at time t. From that point on, every second packet from Alice to Bob gets dropped (but all packets from Bob to Alice reach Alice). I.e., the first packet from Alice to Bob reaches Bob, the next one is dropped, the next one again reaches Bob, the next one is dropped, and so on. This continues for a very long time.

What is, approximately, the average throughput from Alice's process to Bob's process after this long time? Justify your answer. For this question, assume time begins at t (ignore what happened before time t).

The first time a packet is lost, Alice will time out, reset her window to $1MSS$, and retransmit one packet. If this packet reaches Bob, Bob will acknowledge it, Alice will increase her window to $2MSS$, and transmit two packets. Only the second one will reach Bob, so Alice will time out, reset her window to $1MSS$, and retransmit one packet. And so on.

Hence, average throughput is $\frac{2MSS}{4D+2Timeout}$.



Problem 3, Question 5, Solution

Question 6 (3 **points):**

Alice creates a client/server application that connects workers of different professions to candidate customers. E.g., if someone needs a carpenter, they use a client process to send a relevant request to a server process, which returns to them a list of available carpenters in their geographical area.

How do you think that the server process determines which profession each client is interested in and which is the client's geographical area?

Each client includes the "profession" and "geographical area" they are interested in as fields in the header of this application.

Scratch Paper

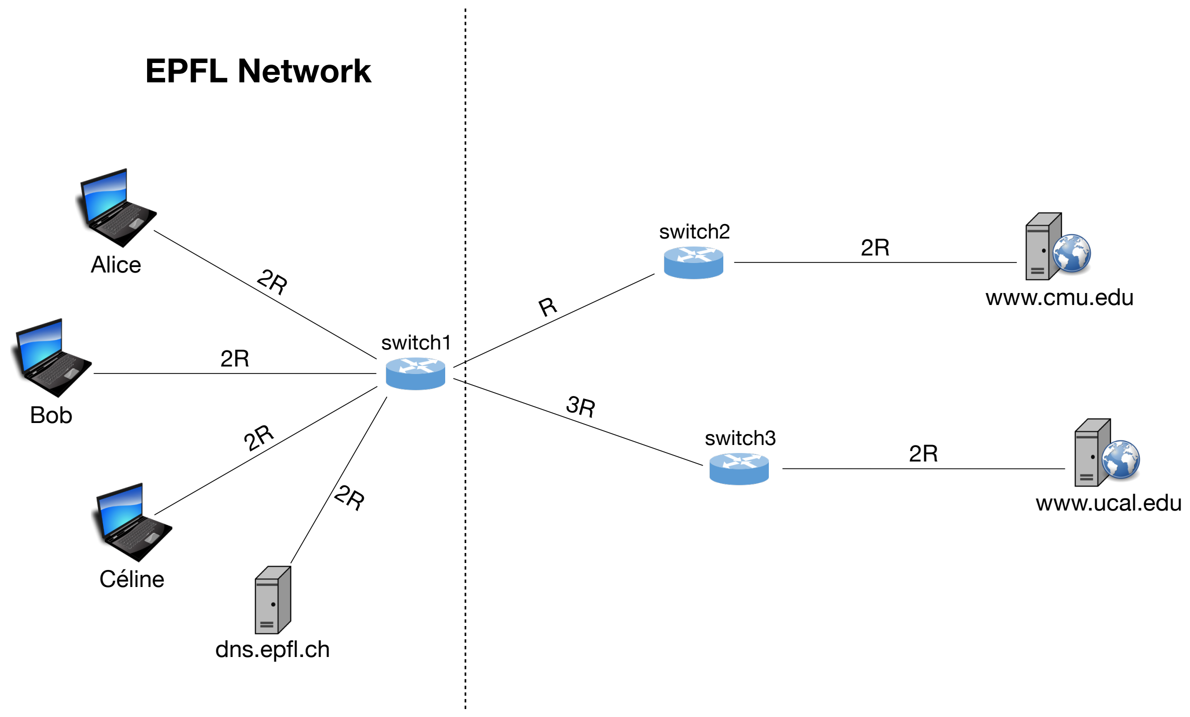


Figure 5: The network topology used in Problem 2.