Computer Networks - Midterm

November 2, 2018

Duration: 2h15m

- This is a closed-book exam.
- Please write your answers on these sheets in a readable way, in English or in French.
- You can use extra sheets if necessary (don't forget to put your name on them).
- The total number of points is 100.
- This document contains 16 pages.
- Good luck!

Full Name (Nom et Prénom):

SCIPER No:

Division: ☐ Communication Systems ☐ Computer Science
☐ Other (mention it): . . . .

Year: ☐ Bachelor Year 2 ☐ Bachelor Year 3
☐ Other (mention it): . . . .

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

1 Short questions (10 points)

For each question, please circle a single best answer.

1. Which of the following identifies a process?
   (a) http://www.epfl.ch
   (b) http://www.epfl.ch/index.html
   (c) www.epfl.ch:80 *(Correct)*
   (d) 192.168.76.3

2. Which mechanism do proxy web servers use to check for stale cached data?
   (a) Timestamps.
   (b) Cookies.
   (c) Conditional GET requests. *(Correct)*
   (d) All of the above.

3. Which of the following DNS servers is guaranteed to know the IP address of www.epfl.ch?
   (a) An authoritative DNS server that is responsible for epfl.ch. *(Correct)*
   (b) A top-level-domain DNS server that is responsible for .ch.
   (c) The root DNS server that is physically closest to EPFL.
   (d) All of the above.

4. If a root DNS server stops caching name-to-IP mappings,
   (a) it will start receiving more DNS requests.
   (b) it will start sending more DNS requests. *(Correct)*
   (c) it will start sending incorrect DNS responses.
   (d) nothing will change in its behavior. *(Correct)*

5. According to the analysis we did in class, peer-to-peer file distribution scales better than client-server file distribution, because
   (a) peer processes typically run on more powerful machines.
   (b) peer processes typically consume fewer resources than client or server processes.
   (c) as more peers join the system, there are more requests for download, but also more sources to download from. *(Correct)*
   (d) most peers tend to request the same few pieces of content.

6. Which is the minimum number of UDP sockets that a process must create in order to communicate with 1000 remote processes?
   (a) 1. *(Correct)*
   (b) 2 (one for sending, and one for receiving traffic).
   (c) 1000.
   (d) I don't have enough information to answer.

7. Why do we use pipelining in reliable data delivery protocols?

   (a) To recover from packet corruption and loss.
   (b) To reduce the transmission delay between sender and receiver.
   (c) To reduce the RTT (round-trip time) between sender and receiver.
   (d) To improve the throughput between sender and receiver. *(Correct)*

8. If we want to significantly decrease the RTT between a sender and a receiver, we should

   (a) reduce the size of the packets they exchange.
   (b) reduce the length of the links that interconnect them. *(Correct)*
   (c) increase the transmission rates of the links that interconnect them.
   (d) do all of the above.

9. If we increase the size of a packet switch's forwarding table,
   the packets that traverse the switch may experience higher

   (a) transmission delay.
   (b) propagation delay.
   (c) queuing delay. *(Correct)*
   (d) None of the above.

10. A sender is connected to a receiver over three consecutive links.
    What will happen if we double the transmission rate of one of these links?

    (a) We will double the throughput from sender to receiver.
    (b) We will halve the RTT between sender and receiver.
    (c) We will impact neither the throughput nor the delay between sender and receiver.
    (d) I don't have enough information to answer. *(Correct)*

2 Lab-related questions (15 points)

Question 1 (10 points):

Suppose you ping a given destination (e.g., `www.epfl.ch`) multiple times and with different packet sizes, and you obtain the following information:

Packet size	Average delay	Minimum delay
50	15.160	15.010
250	15.480	15.230
500	15.960	15.460
750	16.480	16.080
1000	16.960	16.560
1250	17.480	17.180
1500	17.960	17.510

What is a good approximation of the propagation+processing delay of the path from your machine to the target and back? What is a good approximation of the throughput of the bottleneck link on this path? Explain your answers.

The best approximation to propagation+processing delay is the minimum delay among all minimum delays, for this question it is: 15.010 ms. The reason why we may make this approximation is that the packet that faced the minimum delay of all other packets, is the one that has been affected the least from other types of delay like queuing or transmission delay. For that packet, the propagation and the processing delay took the most significant share of the total delay. Thus, we can consider the other delays (queuing and transmission) to be insignificant, and approximate the propagation+processing delay with the minimum overall delay of all. Usually, the overall minimum delay is the minimum delay for the smallest packet, because this is the one that faces the smallest transmission delay of all other packets.

The link rate is: Packet size/Transmission delay. We can estimate the transmission delay by looking at the minimum delays of the smallest and the largest packets, and get a good approximation of the transmission delay that would be faced by a packet whose size is equal to the difference of these two: i.e. 1500 - 50 = 1450 bytes. The transmission delay of such a packet is the difference between the minimum delays of the largest packet and the one of the smallest packet; for this question this is: 17.510 - 15.010 = 2.500 [ms]. Then, the link throughput is:

$$\begin{aligned}
\textit{Transmission delay} &= 2.5\ \textit{ms} = 0.0025\ \textit{seconds}. \\
\textit{Packet size} &= (1500 - 50) = 1450\ \textit{bytes} = 11600\ \textit{bits}. \\
\textit{Link throughput} &= 11600/0.0025 = 4640000\ \textit{bps} = 4640\ \textit{Mbps}.
\end{aligned}$$

The reason why this approximation holds is two-fold: (a) the minimum delays are the ones that have been affected the least from the queuing delay, and (b), when taking the difference of these minimum delays, the propagation and processing delays (that are almost equal for every packet size) cancel out. Thus, the increase of the total delay when considering these two packets (the smallest and the largest), is mainly because of the increase of the transmission delay that depends on the packet size. Of course, we may reason in the same way for any pair of packets with different packet size, not only the smallest and the largest ones. And we may provide an estimate of the transmission delay that is faced by a packet whose size is equal to the difference of the sizes of the two packets in the pair. Here we just gave an example computation.

Question 2 (5 points):

Are traceroutes always symmetric? In other words, does the traceroute from Alice to Bob always include the same nodes as the traceroute from Bob to Alice? Explain your answer.

No, they do not. A general rule of thumb is that routes on the Internet do not need to be symmetric. For example, an administrative entity may choose to employ separate routers to handle incoming and outgoing connections, to achieve better load balancing. Nevertheless, even when both the forward and the reverse path cross the same router, it is possible that different IP addresses are observed. The reason behind this is that the names we see in the traceroute output are the names of router interfaces and not of routers. So, both IP addresses can indeed belong to the same router, but have been allocated to different interfaces of it.

3 Web browsing (30 points)

Consider Figure 1.

All EPFL end-systems:

- Use local DNS server `ns.epfl.ch`, whose IP address they already know.
- Access the web through proxy web server `proxy.epfl.ch`, whose IP address they already know.
- Implement the following security measure: if an end-system sends out a DNS request and receives more than one DNS response to this request within a short period of time (all with the same destination port number), it drops all these DNS responses.

All DNS caches are initially empty. The proxy web server's web cache is initially empty.

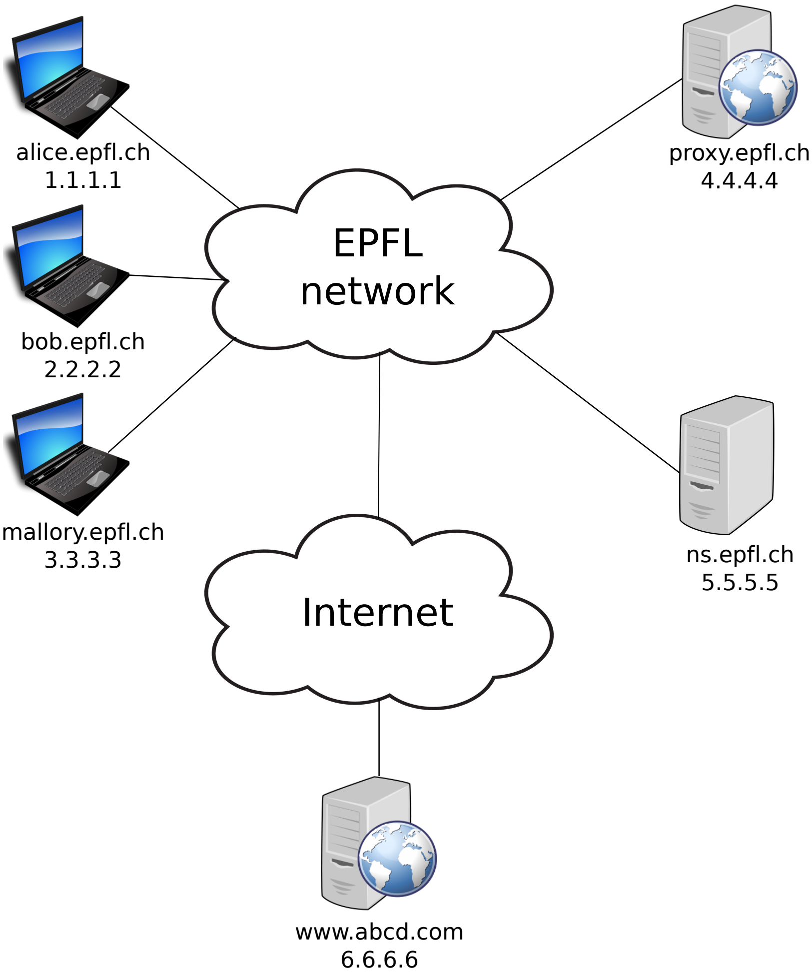

Figure 1: Topology for Problem 3.

Question 1 (12 points):

Mallory asks the local DNS server for the IP address of `www.abcd.com`.

Shortly after, Alice uses her web browser to retrieve page `http://www.abcd.com/index.html`, which includes two images: `http://www.abcd.com/img1.png` and `http://www.abcd.com/img2.png`.

a. Which is the most efficient way for Alice's web browser to use the underlying transport layer protocol? **For the rest of Problem 3, assume that all web browsers use the underlying transport layer protocol as you described.**

   With a persistent connection and pipelining.

b. List all packets that are exchanged in the network **from the moment Alice sends her first packet**, including any connection-setup packets, by filling in Table 1. The number of rows in Table 1 does not imply anything about the size of the solution.

   The local DNS server has cached `www.abcd.com`'s IP address after Mallory's request. The proxy gets all objects from the origin web server, because they are accessed for the first time.

Packet	Source IP	Dest. IP	Transport protocol	Application protocol	Purpose
ex.1	`1.0.0.1`	`1.0.0.2`		HTTP	HTTP reply with `index.html`
1	`1.1.1.1`	`4.4.4.4`	TCP		Connection req.
2	`4.4.4.4`	`1.1.1.1`	TCP		Connection resp.
3	`1.1.1.1`	`4.4.4.4`	TCP	HTTP	GET /index.html, host: www.abcd.com
4	`4.4.4.4`	`5.5.5.5`	UDP	DNS	IP address of www.abcd.com?
5	`5.5.5.5`	`4.4.4.4`	UDP	DNS	6.6.6.6
6	`4.4.4.4`	`6.6.6.6`	TCP		Connection req.
7	`6.6.6.6`	`4.4.4.4`	TCP		Connection resp.
8	`4.4.4.4`	`6.6.6.6`		HTTP	GET /index.html; host: www.abcd.com
9	`6.6.6.6`	`4.4.4.4`		HTTP	GET response; OK with index.html
10	`4.4.4.4`	`1.1.1.1`		HTTP	GET response; OK with index.html
11	`1.1.1.1`	`4.4.4.4`		HTTP	GET /img1.png, host: www.abcd.com
12	`1.1.1.1`	`4.4.4.4`		HTTP	GET /img2.png, host: www.abcd.com
13	`4.4.4.4`	`6.6.6.6`		HTTP	GET /img1.png; host: www.abcd.com
14	`4.4.4.4`	`6.6.6.6`		HTTP	GET /img2.png; host: www.abcd.com
15	`6.6.6.6`	`4.4.4.4`		HTTP	GET response; OK with img1.png
16	`6.6.6.6`	`4.4.4.4`		HTTP	GET response; OK with img2.png
17	`4.4.4.4`	`1.1.1.1`		HTTP	GET response; OK with img1.png
18	`4.4.4.4`	`1.1.1.1`		HTTP	GET response; OK with img2.png

Table 1: Packets transmitted on the Internet, in Problem 3, Question 1

Question 2 (8 points):

Shortly after, Bob uses his web browser to retrieve page `http://www.abcd.com/news.html`, which references image `http://www.abcd.com/img1.png` (which was also referenced in Question 1).

List all packets that are exchanged in the network, including any connection-setup packets, by filling in Table 2. The number of rows in Table 2 does not imply anything about the size of the solution.

The proxy has already cached the image, so it sends a conditional GET.

Packet	Source IP	Dest. IP	Transport protocol	Application protocol	Purpose
ex.1	`1.0.0.1`	`1.0.0.2`		HTTP	HTTP reply with `index.html`
1	`2.2.2.2`	`4.4.4.4`	TCP		Connection request
2	`4.4.4.4`	`2.2.2.2`	TCP		Connection response
3	`2.2.2.2`	`4.4.4.4`	TCP	HTTP	GET /news.html; host: www.abcd.com
4	`4.4.4.4`	`6.6.6.6`	TCP		Connection request
5	`6.6.6.6`	`4.4.4.4`	TCP		Connection response
6	`4.4.4.4`	`6.6.6.6`	TCP	HTTP	GET /news.html; host: www.abcd.com
7	`6.6.6.6`	`4.4.4.4`	TCP	HTTP	GET response; OK with news.html
8	`4.4.4.4`	`2.2.2.2`	TCP	HTTP	GET response; OK with news.html
9	`2.2.2.2`	`4.4.4.4`	TCP	HTTP	GET /img1.png; host: www.abcd.com
10	`4.4.4.4`	`6.6.6.6`	TCP	HTTP	GET /img1.png; host: www.abcd.com; **If-modified-since:...**
11	`6.6.6.6`	`4.4.4.4`	TCP	HTTP	GET response; Not modified (header of img1.png)
12	`4.4.4.4`	`2.2.2.2`	TCP	HTTP	GET response; OK with img1.png

Table 2: Packets transmitted on the Internet, in Problem 3, Question 2

Question 3 (3 points):

What is the minimum number of sockets that `proxy.epfl.ch` uses in Question 1? What is each socket's type and purpose?

1 UDP socket for communication with the DNS server. 1 TCP socket to listen for incoming connections. 1 TCP socket for communication with Alice. 1 TCP socket for communication with www.abcd.com.

Question 4 (10 points):

Mallory knows that Bob is about to access a new page, `http://www.efgh.com/index.html`, and wants to make him access a fake page from her machine instead.

Can she do this? If no, explain why. If yes, explain in which scenario she can do it and how.

Recall that EPFL machines implement a special security measure.

She can, in the scenario where nobody has accessed `www.efgh.com` before the attack, such that the proxy web server has not cached `www.efgh.com`'s correct IP address.

Mallory can guess when the proxy web server sends a DNS request for `www.efgh.com`'s IP address and send a fake DNS response, i.e., impersonate the local DNS server.

To circumvent the security measure, she needs to prevent the local DNS server from responding to the proxy web server's DNS request. She can do this by launching a denial-of-service attack against the local DNS server, e.g., flooding the link to the DNS server with enough traffic to cause congestion, such that the proxy web server's DNS request is either dropped or significantly delayed.

4 Delay and transport layer (45 points)

Figure 2 shows the topology of the EPFL network from Figure 1, but in more detail.

- Each link has length ℓ and propagation speed c.
- Each link is annotated with its transmission rate, which is the same in both directions of the link.
- Packet switches $S1$ and $S2$ perform store-and-forward packet switching and introduce insignificant processing delays. Each switch has a separate, infinite queue for each outgoing link. E.g., switch S_2 has three queues: one for packets going to the proxy web server, one for packets going to Deborah, and one for packets going to the other end-systems. As soon as S_2 receives a packet, it immediately places the packet in the corresponding queue, depending on the packet's destination.
- The end-systems communicate using a Go-Back-N transport layer protocol with sender window size $N = 10P$. There is no packet loss or corruption unless explicitly stated.

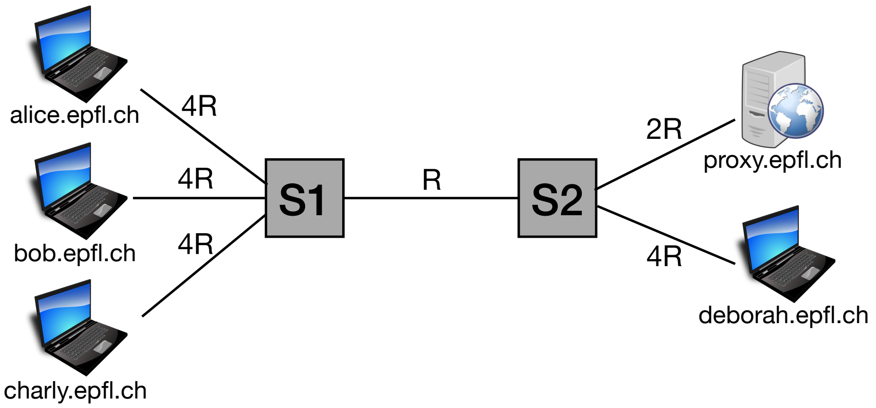

Figure 2: The topology of the EPFL network from Figure 1, in more detail

Question 1 (10 points):

a. Alice sends a request to the proxy web server, which fits in a single packet of length Q.

How long does it take for this request to be fully delivered to the proxy web server? Explain your answer.

The timing diagram is shown in Figure 3. Due to the fact that the switches use store-and-forward packet switching, they wait until receiving the full packet, before transmitting it. Also, there is no processing delay. The transfer time is

$$\frac{Q}{4R} + \frac{Q}{R} + \frac{Q}{2R} + 3\frac{l}{c}.$$

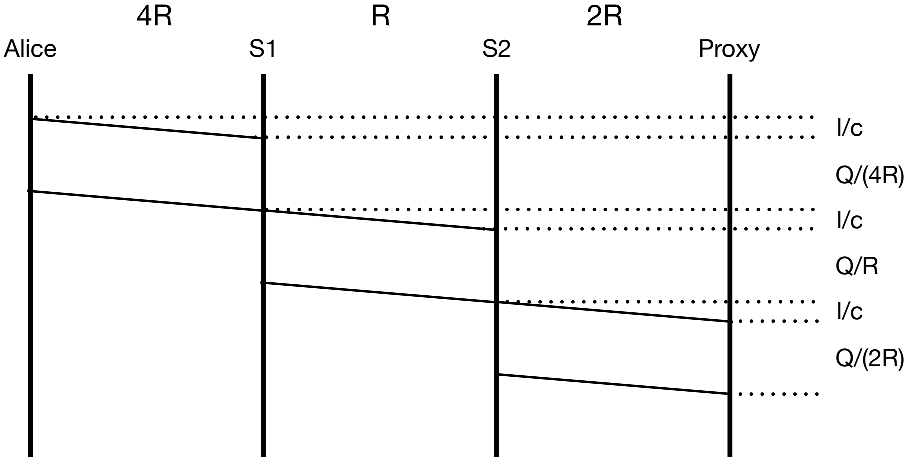

Figure 3: Timing diagram for transferring one packet in sequence over three links.

b. The proxy web server splits a file into P packets of equal length L and sends them back-to-back to Alice.

How long does it take for this file to be fully delivered to Alice? Explain your answer.

We use the timing diagram in Figure 4. The transfer time is

$$\frac{L}{2R} + \frac{PL}{R} + \frac{L}{4R} + 3\frac{l}{c}.$$

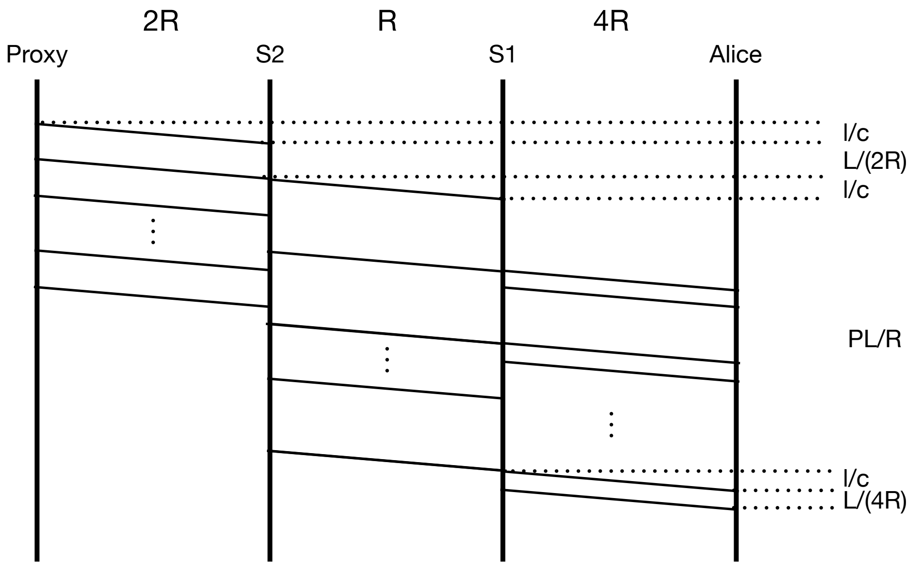

Figure 4: Timing diagram for transferring P equal-sized packets in sequence over three links.

Question 2 (10 points):

Consider again Question 1b and suppose that switch $S1$ is compromised and corrupts the payload of a single packet.

Which packet should that be so as to cause the maximum number of retransmissions?

Draw a diagram that shows all the packets transmitted by Alice and the proxy web server, including their sequence and acknowledgment numbers, and all the events that cause one or more retransmissions. Pick any number you want as the first sequence number.

The first packet, such that Alice has to retransmit everything when she times out.

Let the first packet have sequence number 1*. The diagram is shown in Figure 5.*

The above solution assumes that the timeout is larger or equal to the time it takes for all P *packets to be transmitted. If this is not the case,* $S1$ *can corrupt any packet from the first one up to the* i^{th}*-to-last, where* i *is the timeout expressed in number of packets transmitted, so as to cause* $i < P$ *retransmissions.*

Both answers were accepted as correct.

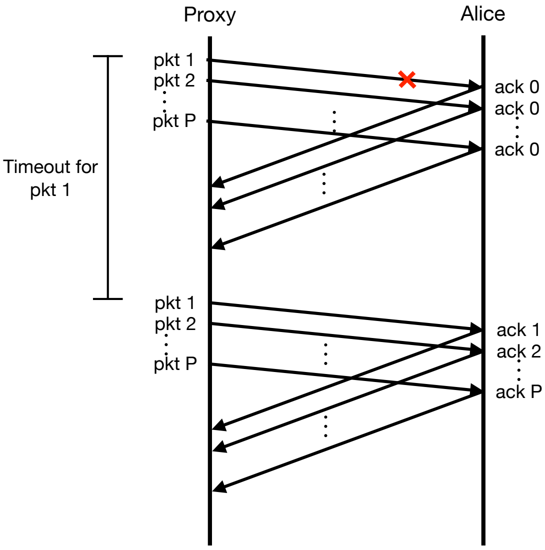

Figure 5

Question 3 (10 points):

Alice has requested a file from the proxy web server, and the proxy web server has retrieved it. The proxy web server splits the file into 2 packets, the first of length L and the second of length $0.1L$, and sends them back-to-back to Alice.

a. What is the queuing delay of each packet? Explain your answer.

First packet: 0*, since there is no other traffic in the network before the first packet, so it always begins transmission immediately.*

Second packet (see Figure 6):

$$d_1 + d_2 = \left(\frac{L}{R} - \frac{0.1L}{2R}\right) + \left(\frac{L}{4R} - \frac{0.1L}{R}\right).$$

Answers that additionally considered the time the second packet waits at the proxy for the first packet to be transmitted over the first link $\left(\frac{L}{2R}\right)$ *also received full points.*

b. How long does it take for the file to be fully delivered to Alice? Explain your answer.

We use again the timing diagram shown in Figure 6. The transfer time is

$$\frac{L}{2R} + \frac{L}{R} + \frac{1.1L}{4R} + 3\frac{l}{c}.$$

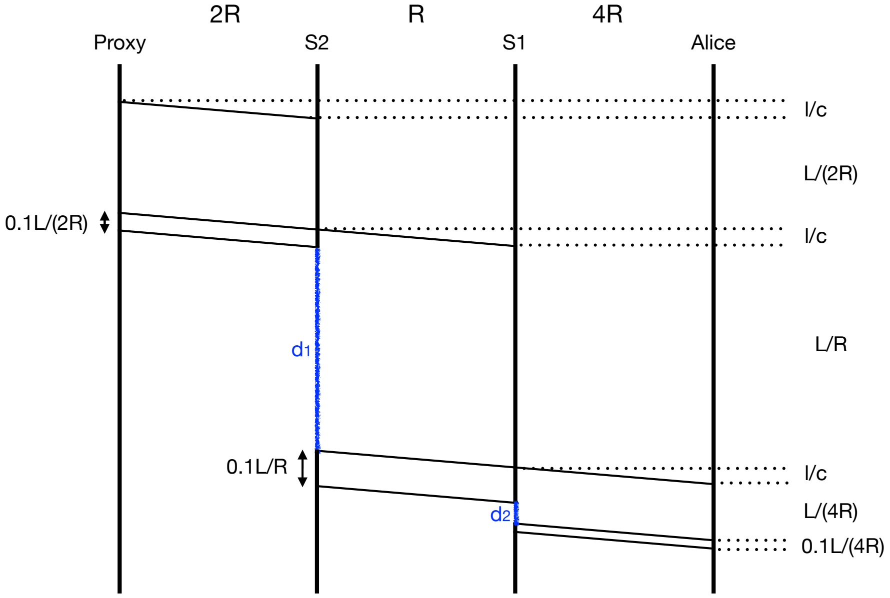

Figure 6

Question 4 (10 points):

Consider the same setup as in Question 3. Additionally, Deborah sends 3 packets of equal length L back-to-back to Charly. Deborah starts transmitting her first packet at exactly the same time that the proxy web server starts transmitting its first packet.

How long does it take for the proxy web server's file to be fully delivered to Alice? Explain your answer.

Let D_1, D_2, D_3 denote the first, second and third packet of Deborah and P_1, P_2 the first and second packet of the proxy.

Only two out of the three packets of Deborah interfere with the communication between the proxy and Alice: the first packet to arrive at $S2$ is D_1 (at time $t_1 = \frac{L}{4R} + \frac{l}{c}$), followed by the simultaneous arrival of D_2 and P_1 ($t_2 = \frac{L}{2R} + \frac{l}{c}$). The next packet to reach $S2$ is P_2 ($t_3 = \frac{1.1L}{2R} + \frac{l}{c}$), and finally, D_3 ($t_4 = \frac{3L}{4R} + \frac{l}{c}$).

Depending on whether $S2$ prioritizes the transmission of D_2 over P_1 on the link $S2 \to S1$ or not, we distinguish two cases (correct answers to any of these cases received full points):

Case 1: D_2 goes ahead of P_1

See Figure 7. In this scenario, P_2 experiences the same queuing delay at $S1$ as in Question 3a, hence the transfer time is

$$\frac{L}{4R} + \frac{3L}{R} + \frac{L}{4R} + \frac{0.1L}{4R} + 3\frac{l}{c}.$$

Case 2: D_2 goes after P_1

As shown in Figure 8, P_2 is not queuing up at $S1$. The transfer time is

$$\frac{L}{4R} + \frac{3.1L}{R} + \frac{0.1L}{4R} + 3\frac{l}{c}.$$

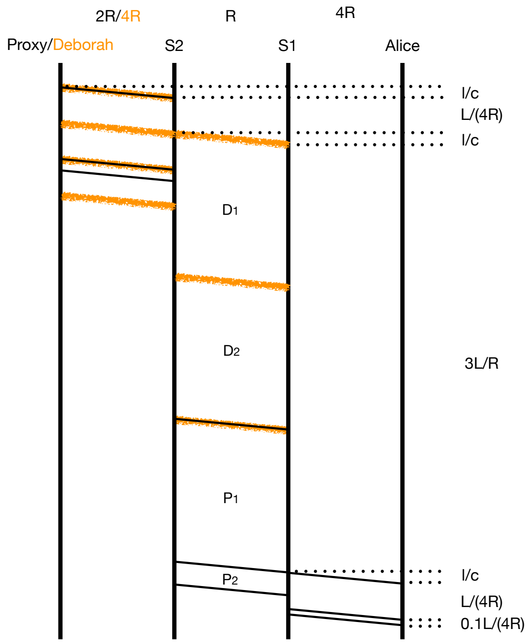

Figure 7: Case 1.

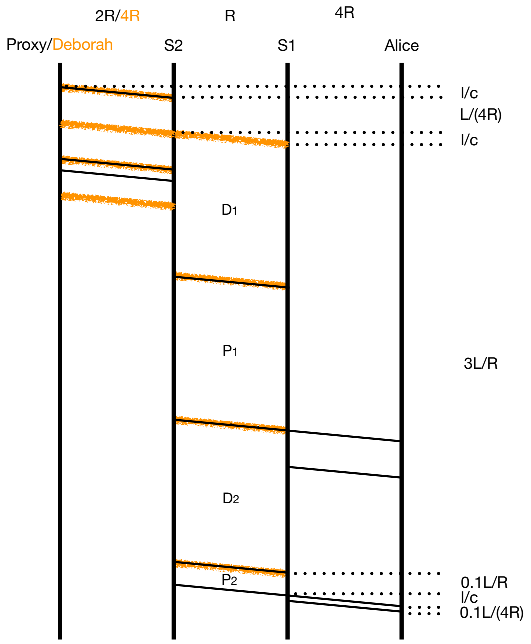

Figure 8: Case 2.

Question 5 (5 points):

In Question 4, could Alice retrieve the file faster by requesting it directly from the origin web server? What does the answer depend on?

Yes, it is possible.

The answer depends on the topology of the network, i.e., whether the path from Alice to the origin web server crosses the congested link.

It also depends on the conditions on the rest of the Internet, e.g., the transmission rate, length and propagation speed of the traversed links and queuing delays due to other traffic crossing the same links.