Computer Networks - Midterm

November 6, 2020

Duration: 2h15m

- This is an open-notes exam.
- Write your answers clearly, in English or in French, using extra sheets if necessary. When you are done, scan or take a picture of your entire exam and upload it to Moodle.
- The total number of points is 50.
- This document contains 14 pages.
- Please work on your own. 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. DSL and cable are both:
   (a) Technologies that provide reliable data delivery.
   (b) Technologies used to connect households to the Internet. *(Correct)*
   (c) Transport-layer technologies.
   (d) Shared broadcast technologies.

2. An Internet Service Provider (ISP) and an Internet eXchange Point (IXP) may be related in the following way:
   (a) Both of them are customers of the same Tier-1 ISP.
   (b) The IXP is a customer of the ISP.
   (c) The ISP is physically connected to the IXP. *(Correct)*
   (d) They cannot be related in any way.

3. When a process running on your computer sends a message to a remote process (running on a different computer), which layers on your computer get involved (i.e., to process/handle the message)?
   (a) The application layer.
   (b) The transport layer.
   (c) The application and transport layers.
   (d) All the layers. *(Correct)*

4. End-systems Alice and Bob are connected over a sequence of N links, each of propagation delay D. Alice sends a packet of size L bits to Bob. The propagation delay experienced by the packet is:
   (a) D · N · L (D multiplied by N multiplied by L).
   (b) D.
   (c) D · N (D multiplied by N). *(Correct)*
   (d) I don't have enough information to answer.

5. End-systems Alice and Bob are connected over a sequence of N links, each of transmission rate R and propagation delay D. All packet switches are store-and-forward. Alice sends a file of size F to Bob, as a sequence of back-to-back packets of size L. The transfer time for the file is F/R + 9L/R + 10msec. Assume 0 queuing and processing delays. Which of the following is true?
   (a) N=1 link, D=10msec.
   (b) N=9 links, D=1msec.
   (c) N=10 links, D=1msec. *(Correct)*
   (d) N=10 links, D=10msec.

6. In which of the following scenarios could ”connection switching” be a better option than packet switching?
   (a) The network layer is perfectly reliable (it never corrupts, loses, or unpredictably delay packets).
   (b) Network links have very high transmission rates.
   (c) The network layer must provide performance guarantees. *(Correct)*
   (d) ”Connection switching” is never a better option than packet switching.
7. What does it mean that the network layer of the Internet provides ”best-effort service”?
   (a) Packets may be lost.
   (b) Packets may be unpredictably delayed.
   (c) Packets may be corrupted.
   (d) All of the above. *(Correct)*
8. Cookies enable a web server to link subsequent HTTP requests to the same:
   (a) Human user.
   (b) Web browser. *(Correct)*
   (c) Source IP address.
   (d) Computer.
9. Web clients and servers communicate over persistent TCP connections (as opposed to non-persistent ones) because that enables them to:
   (a) Achieve reliable data delivery.
   (b) Make better use of caching.
   (c) Keep state (information) about each other.
   (d) Amortize the cost of TCP connection setup over multiple HTTP messages. *(Correct)*
10. What might happen if TCP did a fast retransmit upon receiving only one duplicate ACK (as opposed to three duplicate AKCs):
   (a) Faster recovery of lost segments.
   (b) More unnecessary retransmissions.
   (c) Fewer timeouts.
   (d) All of the above. *(Correct)*

2 Web browsing + delays (30 points)

Consider the network in Figure 1:

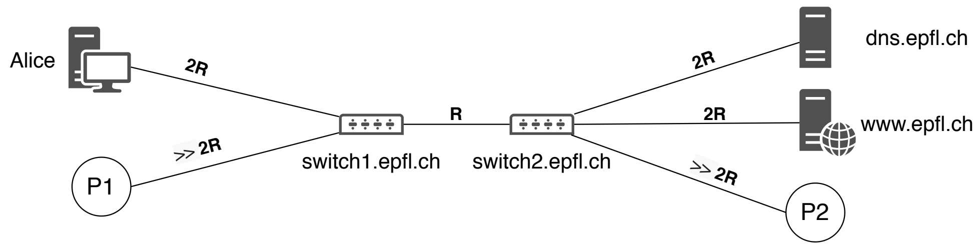

Figure 1: Network Topology for Problem 2

- Each link has propagation delay D_p.
- For each link, the transmission rate has the value shown in the figure in each direction. The symbol "$\gg$" means "significantly larger than."
- `switch1.epfl.ch` and `switch2.epfl.ch` are store-and-forward packet switches. When a packet arrives at a packet switch, it is stored in a queue that has a finite size and fits Q packets of size MSS.
- Assume 0 processing delays. When computing delays, assume packet header size is insignificant and do not assume anything else about delays (e.g., do not assume that propagation delays are insignificant relative to transmission delays).
- The maximum segment size (MSS) in this network is 100 bytes.
- If you don't know the size of a message, assume it is equal to MSS, i.e., 100 bytes.
- Alice's computer and Bob's computer (Bob's is not shown in the picture) both use `dns.epfl.ch` as their local DNS server.
- Web browsers and servers communicate over persistent TCP connections.
- Web browsers and servers use a simplified version of TCP that does no flow or congestion control.
- The sender window is stable and set to $100,000$ bytes throughout the connection.
- No packet loss occurs throughout this problem. Q is large enough to absorb all the packets sent by all the end-systems.

Bob (not shown in the picture) accesses web page `http://www.epfl.ch/index.html`, which contains no embedded objects. The base file for this web page is $10,000$ bytes long.

A short time later, Alice accesses the same web page. This is the first time ever that Alice's computer needs to contact `www.epfl.ch`.

Question 1 (4 **points):**

List the first 8 packets generated as a result of Alice's action, including any TCP connection setup packets. In particular, for each packet, state:

- Which computer sends it.
- Which computer receives it.
- The source port number.
- The destination port number.
- A short description (a few words) of what message the packet is carrying, e.g., "request for …"

If the value of a port number could be anything, you can write "x" instead of an actual number.

Source	*Destination*	*Src port*	*Dst port*	*Role*
Alice	*DNS server*	*x*	*53*	*DNS request for web server's IP address*
DNS server	*Alice*	*53*	*x*	*DNS response*
Alice	*web server*	*y*	*80*	*TCP connection setup request*
web server	*Alice*	*80*	*y*	*TCP connection setup response*
Alice	*web server*	*y*	*80*	*HTTP GET request for base file*
web server	*Alice*	*80*	*y*	*HTTP GET response (first 100 bytes of the file)*
web server	*Alice*	*80*	*y*	*HTTP GET response (next 100 bytes)*
web server	*Alice*	*80*	*y*	*HTTP GET response (next 100 bytes)*

Question 2 (3 **points**):

How many calls to the transport layer does Alice's application layer make? What is the role of each call? (You don't need to give the exact names of the calls. Only the role of each call.)

By "calls" we mean calls to functions, like "send" and "receive," used by the application layer to communicate with the transport layer.

From 10 *to* 12 *calls:*

- *(Optional) Create UDP socket.*
- *(Optional) Bind it to a local IP address and port number.*
- *Send DNS request.*
- *Read DNS response.*
- *(Optional) Close the UDP socket.*
- *Create TCP socket.*
- *Bind it to a local IP address and port number.*
- *Connect to the web server's listening socket.*
- *Send HTTP GET request.*
- *Read the beginning of the HTTP response (including the headers that mention the size of the response).*
- *Read the rest of the HTTP response.*
- *(Optional) Close the TCP socket.*

Question 3 (3 **points**):

Assume there is no traffic in the network outside the one caused by Alice.

How long does it take from the moment Alice's computer starts transmitting the first packet until it is ready to transmit the first data-carrying segment to `www.epfl.ch`? Justify your answer.

By "data-carrying segment" we mean a segment that carries an application-layer message (or part of an application-layer message).

DNS request:

- *Transmission delay, one term per link:* $MSS\left(\frac{1}{2R}+\frac{1}{R}+\frac{1}{2R}\right)=\frac{2MSS}{R}$.
- *Propagation delay,* d_p *per link:* $3d_p$.
- *Total:* $\frac{2MSS}{R}+3d_p$.

DNS response: same as DNS request.

TCP connection-setup and response: $6d_p$.

Total: $\frac{4MSS}{R} + 12d_p$.

Question 4 (6 **points):**

Assume there is no traffic in the network outside the one caused by Alice.

How long does it take from the moment Alice's computer starts transmitting the first data-carrying segment to `www.epfl.ch` until it receives the last bit of the requested web page? Justify your answer.

HTTP GET request:

- *Transmission delay, one term per link:* $MSS\left(\frac{1}{2R}+\frac{1}{R}+\frac{1}{2R}\right)=\frac{2MSS}{R}$.
- *Propagation delay, one term per link:* $3D_p$.
- *Total:* $\frac{2MSS}{R}+3D_p$.

HTTP GET response:

- *Transmission delay of the first packet on the first link:* $MSS\frac{1}{2R}$.
- *Transmission delay of the file on the second (bottleneck) link:* $100MSS\frac{1}{R}$.
- *Transmission delay of the last packet on the last link:* $MSS\frac{1}{2R}$.
- *Propagation delay, one term per link:* $3D_p$.
- *Total:* $\frac{101MSS}{R}+3D_p$.

Total: $\frac{103MSS}{R}+6D_p$.

Question 5 (3 **points):**

Denis is a malicious user who wants to slow down Alice's communication with the web server. Which scenario is better for Denis, i.e., enables Denis to delay Alice's receiving the web page more?

a. His computer is located in position $P1$ (see Figure 1). A few moments before Alice starts communicating with the web server, Denis starts sending traffic to the web server at the highest possible rate. He continues doing this throughout Alice's communication with the web server.

b. His computer is located in position $P2$ (see Figure 1). A few moments before Alice starts communicating with the web server, Denis starts sending traffic to Alice at the highest possible rate. He continues doing this throughout Alice's communication with the web server.

Justify your answer.

Scenario (b) enables Denis to slow down Alice's communication more.

If Denis is in Position $P1$, he can potentially slow down every packet from Alice to the web server. If Denis is in Position $P2$, he can potentially slow down every packet from the web server to Alice. The web server sends more packets to Alice than the other way round, so being in Position P2 enables Denis to slow down Alice's communication more.

Question 6 (6 **points):**

Assume the scenario you chose in your answer to Question 5. Also, assume that no packets are lost (the packet-switch queues are large enough to absorb Alice's, Denis's, and the web server's traffic).

Compute an upper bound for the amount of time that it takes from the moment Alice's computer starts transmitting the first data-carrying segment to `www.epfl.ch` until it receives the last bit of the web page. Justify your answer.

- *Since we assume no packet loss and* 0 *processing delay, the worst that Denis can do is introduce queuing delay.*
- *The max queuing delay that a packet from the web server to Alice may encounter is:*

$$(Q-1)\frac{MSS}{R}.$$

- *The web server sends* 101 *packets to Alice: a TCP-connection setup response that fits in* 1 *packet, and an HTTP response split into* 100 *packets.*
- *If Denis causes each of these packets to encounter the max queuing delay, the total queuing delay he introduces is:*

$$101(Q-1)\frac{MSS}{R}.$$

- *If we add this to the answer to Question 4, we get:*

$$(103+101(Q-1))\frac{MSS}{R}+6D_p.$$

(Lab related) Question 7 (5 **points):**

Alice runs `traceroute www.epfl.ch`, while there is no other traffic in the network. What output will she get?

If you can think of multiple possible outputs, pick the one that provides the most information that traceroute could ever give.

If you don't have enough information to determine the value of a field, e.g., an IP address, make it up. Justify your answer.

```
switch1.epfl.ch (1.1.1.1) x1 x2 x3
switch2.epfl.ch (1.1.1.2) y1 y2 y3
www.epfl.ch (1.1.1.3) z1 z2 z3
```

where $x1, x2, x3$ *are a bit above* $2D_p$*,* $y1, y2, y3$ *are a bit above* $4D_p$*,*
and $z1, z2, z3$ *are a bit above* $6D_p$*.*

Since there is no other traffic in the network (hence no queuing delays) and we assumed 0 *processing delays, the RTT to each device will be approximately the sum of the propagation delays of all the links encountered from Alice to the device and back.*

3 Transport layer (15 points)

Consider two end-systems A and B. A process running on A has a certain amount of data ready to send to a process running on B. The process running on B does not have any data to send to the process running on A.

The two processes start communicating over a Go-back-N transport-layer protocol, like the one discussed in class. A's transport layer transmits a segment for the first time at $t = 0$sec, and for the last time at $t = 3.4$sec. The sequence number on A's first data segment is 0.

Figure 2 shows A's transmissions over time.

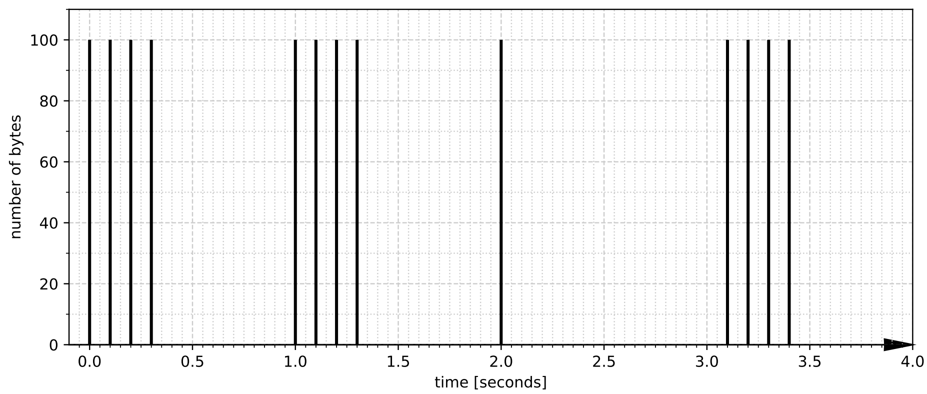

Figure 2: A's transmissions over time

For example, at time $t = 0$sec, A's transport layer transmits a segment of size 100 bytes; at time $t = 0.1$sec, it transmits a segment of size 100 bytes; etc. Each segment shown in the figure might be a new data segment or a retransmission (you will need to guess).

No packet loss occurs until t=1sec (there may or may not be loss after that).

The round-trip time (RTT) between A and B and A's timeout remain stable throughout the communication.

Question 1 (0.5 **point):**

What is the maximum segment size (MSS)? Justify your answer.

100 bytes. A transmits 100 bytes at a time.

Question 2 (0.5 **point):**

What is A's sender window size? Justify your answer.

4. A transmits up to 4 segments one after the other (presumably without waiting for an ACK).

Question 3 (4 **points):**

Assuming no ACK segments from B to A were lost, which data segment(s) from A to B may have been lost? If it's more than one, list all of them. Justify your answer.

5 was definitely lost: A transmits segment 8, which means that it received an ACK for segment 4 (otherwise its window would not allow to send segment 8). Right after, A does not transmit anything, which means that it is stuck waiting for an ACK to segment 5.

Any of 6, 7, and 8 may also have been lost: B discards them and A retransmits them anyway since they are received out of order, so whether they are lost or not does not change the picture.

Question 4 (4 **points):**

Assuming no data segments from A to B were lost, which ACK segment(s) from B to A may have been lost? If it's more than one, list all of them. Justify your answer.

ACKs 5, 6, 7, and 8 were definitely lost, otherwise A would not be stuck waiting for ACK 5.

Also, the ACKs for the retransmitted 5, 6, and 7 may have been lost: Go-back-N ACKs are cumulative. So, if A receives ACK 8 before the timeout for the retransmitted segment 5 expires, that is equivalent to it receiving all the previous ACKs.

Question 5 (1 **point**):

What is the round-trip time (RTT) between A and B? Justify your answer.

Approximately 1*sec. A receives the first ACK approximately* 1*sec afer it transmits the first segment.*

Question 6 (1 **point**):

What is the timeout value used by A's transport layer? Justify your answer.

Approximately 2*sec. A retransmits segment* 5 *approximately* 2*sec afer it transmits it for the first time.*

Question 7 (4 **points**):

What is the average throughput between the two processes (averaged across their entire communication)? Do you have enough information to compute it? If yes, compute it. If not, state what information you are missing and compute the best approximation you can. Justify your answer.

Be careful: in this context, the average throughput is the average rate at which the process running on B receives data sent by the process running on A.

A transmits a total of 9 *segments of size MSS, hence* 900 *bytes.*

We know that the communication ends sometime after $t = 3.4$sec, but we are not sure exactly when, so we cannot precisely compute the average throughput. In particular, we don't know how long it takes for the last retransmitted segment to reach B.

However, based on the picture, when A transmits segment 2 *(the* 3*rd back-to-back segment after* 0 *and* 1*), it takes approximately* 1*sec for A to receive an ACK for* 2*. We could use this as an upper bound for how long it takes for the last retransmitted segment (which is the* 3*rd back-to-back retransmitted segment) to reach B. In that case, B receives the last retransmitted segment at time $t = 3.4 + 1 = 4.4$, and the average throughput is* $\frac{900}{4.4}$.

Another possibility is to say that A must have received an ACK for the last retransmitted segment within 2*sec, otherwise it would have timed out (and we know it did not). We could use this as a (more conservative) upper bound on how long it takes for the last retransmitted segment to reach B. In that case, B receives the last retransmitted segment at time $t = 3.4 + 2 = 5.4$, and the average throughput is* $\frac{900}{5.4}$.