ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Computer Networks - Final Exam

December 21, 2018

Duration: 2:15 hours, closed book.

- This is a closed-book exam.
- Please write your answers on these sheets in a readable way, in English or in French.
- Please do **not** use a red pen.
- 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 23 pages.
- Good luck!

Last Name (Nom):
First Name (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)

Problem 1 (10 points)

For each question, please circle a single best answer.

1. We use the Address Resolution Protocol (ARP) to map:
   (a) A DNS name to an IP address.
   (b) An IP address to a MAC address. *(Correct)*
   (c) An IP address to an output link.
   (d) A MAC address to an output link.

2. You type in your browser the URL of a web page. What is the minimum number of DNS requests that your computer may send out as a result?
   (a) 0. *(Correct)*
   (b) 1.
   (c) 3.
   (d) 4.

3. DNS name `www.ethz.ch` maps to IP address IP_{old}. This mapping expires today at 22h00. An ETHZ administrator changes the mapping to IP_{new} at 21h00. An EPFL end-system makes a DNS request for `www.ethz.ch`'s IP address at 21h15. What answer will it receive?
   (a) IP_{old}.
   (b) IP_{new}.
   (c) Both IP_{old} and IP_{new}.
   (d) I don't have enough information to answer this question. *(Correct)*

4. In the context of a peer-to-peer system like BitTorrent, what information does a distributed hash table (DHT) store?
   (a) Content files.
   (b) Which peers host each content file. *(Correct)*
   (c) Metadata files (e.g., .torrent files).
   (d) Pointers to metadata files (e.g., magnet links).

5. Alice wants to send 10 bytes of data to Bob and she has the option to use UDP or TCP. Which one will cause Alice and Bob to exchange more packets?
   (a) UDP.
   (b) TCP. *(Correct)*
   (c) They will cause the same number of packets.
   (d) It depends on the network conditions.

6. A Network Address Translator (NAT gateway) changes the following fields of a packet going from the internal (local area) network to the external (wide area) network:

   (a) Source IP address.

   (b) Source IP address and source port number. *(Correct)*

   (c) Destination IP address.

   (d) Destination IP address and destination port number.

7. The goal of an intra-domain routing protocol is:

   (a) All link-layer switches in the same IP subnet learn the best path to each other.

   (b) All IP routers in the same IP subnet learn the best path to each other.

   (c) All IP routers in the same Autonomous System (AS) learn the best path to each other. *(Correct)*

   (d) All IP routers in the Internet learn the best path to each other.

8. An IP router has the following entry in its forwarding table: destination IP prefix $P \rightarrow$ output link x. IP prefix P belongs to a different AS than the router. How did the router learn this forwarding entry?

   (a) By observing traffic.

   (b) By participating in a spanning-tree protocol.

   (c) By participating in an intra-domain routing protocol.

   (d) Through BGP. *(Correct)*

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. If a packet switch that performs store-and-forward packet switching changes to cut-through packet switching, the packets that traverse the switch may experience lower:

    (a) Propagation delay.

    (b) Processing delay.

    (c) End-to-end delay. *(Correct)*

    (d) None of the above.

Problem 2 (35 points)

Consider the network in Figure 1, consisting of:

- An end-system that runs both a web server process and a DNS server process. This end-system has two DNS names: `www.epfl.ch` and `dns.epfl.ch`. Both DNS names map to the same IP address.
- A set of other end-systems, which use `dns.epfl.ch` as their local DNS server. They know `dns.epfl.ch`'s IP address, but they don't know that `www.epfl.ch` maps to the same IP address.
- IP routers R_1, R_2, and R_3.
- Link-layer switches S_1, S_2, and S_3 (plus others that are not explicitly shown).

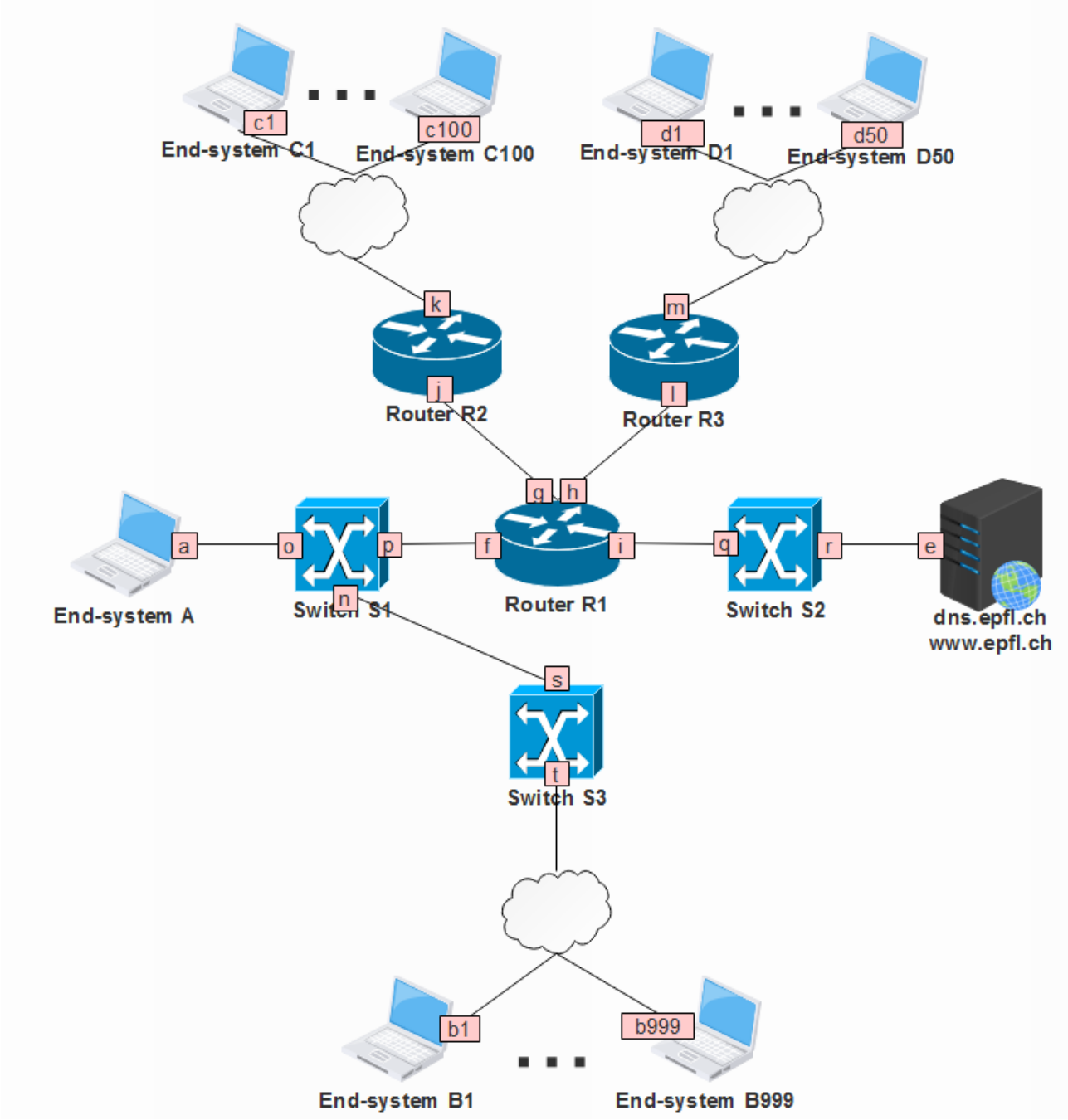


Figure 1: The Network Topology used in Problem 2

Question 1 (10 points):

Allocate an IP prefix to each IP subnet and an IP address to each end-system and IP router network interface, following these rules:

- All IP addresses must be allocated from $100.0.0.0/16$.
- Each IP subnet must be allocated the smallest possible IP prefix and must have one broadcast IP address.
- Each end-system and each IP router (but not link-layer switch) interface has an IP address.

Please explain how you compute each IP prefix and fill in Table 1 in the next page.

First, let's write $100.0.0.0/16$ in its binary form:

$$01100100.00000000.xxxxxxxx.xxxxxxxx.$$

Subnet behind R1, interface f:

- 1002 IP addresses (1000 for the end-systems, 1 for the router interface, and 1 broadcast IP address). Hence, 10 bits.
- IP prefix $01100100.00000000.000000xx.xxxxxxxx = 100.0.0.0/22$.

Subnet behind R2, interface k:

- 102 IP addresses, hence, 7 bits.
- IP prefix $01100100.00000000.00000100.0xxxxxxx = 100.0.4.0/25$.

Subnet behind R3, interface m:

- 52 IP addresses, hence, 6 bits.
- IP prefix $01100100.00000000.00000100.10xxxxxx = 100.0.4.128/26$.

Subnet behind R1, interface g:

- 3 IP addresses, hence, 2 bits.
- IP prefix $01100100.00000000.00000100.110000xx = 100.0.4.192/30$.

Subnet behind R1, interface h:

- 3 IP addresses, hence, 2 bits.
- IP prefix $01100100.00000000.00000100.110001xx = 100.0.4.196/30$.

Subnet behind R1, interface i:

- 3 IP addresses, hence, 2 bits.
- IP prefix $01100100.00000000.00000100.110010xx = 100.0.4.200/30$.

Subnet	IP prefix	Interfaces and IP addresses	Broadcast IP address
Behind $R1$, interface f	$100.0.0.0/22$	f: 100.0.0.0 a: 100.0.0.1 $b1$: 100.0.0.2 $b999$: 100.0.3.232	100.0.3.255
Behind $R2$, interface k	$100.0.4.0/25$	k: 100.0.4.0 $c1$: 100.0.4.1 $c100$: 100.0.4.100	100.0.4.127
Behind $R3$, interface m	$100.0.4.128/26$	m: 100.0.4.128 $d1$: 100.0.4.129 $d50$: 100.0.4.178	100.0.4.191
Behind $R1$, interface g,	$100.0.4.192/30$	g: 100.0.4.192 j: 100.0.4.193	100.0.4.195
Behind $R1$, interface h	$100.0.4.196/30$	h: 100.0.4.196 l: 100.0.4.197	100.0.4.199
Behind $R1$, interface i	$100.0.4.200/30$	i: 100.0.4.200 e: 100.0.4.201	100.0.4.203

Table 1: Allocation of IP prefixes and IP addresses for the network in Figure 1

Question 2 (5 points):

IP routers R_1, R_2, and R_3 participate in a least-cost path routing algorithm, which has converged. Show the forwarding table of R_1 and R_2. You do not need to optimize the routes, i.e., you do not need to merge routes so as minimize the number of entries in each table.

a) Router R_1:

Destination IP prefix	Output link
10.0.0.0/22	f
10.0.4.0/25	g
10.0.4.128/26	h
10.0.4.192/30	g
10.0.4.196/30	h
10.0.4.200/30	i

b) Router R_2:

Destination IP prefix	Output link
10.0.0.0/22	j
10.0.4.0/25	k
10.0.4.128/26	j
10.0.4.192/30	j
10.0.4.196/30	j
10.0.4.200/30	j

Question 3 (10 points):

All link-layer switches have just been rebooted, and all end-system caches are initially empty. Then, the user of desktop A visits web page `www.epfl.ch/index.html`, which contains only one image, `image.png`.

State all the packets that are **transmitted or forwarded by all end-systems and IP routers until A's user can view the web page**. For example, if a packet follows the path $A \rightarrow R_1 \rightarrow R_2 \rightarrow C1$, then you should state it 3 times: when it is transmitted by A, forwarded by R_1, and forwarded by R_2.

Answer by filling in Table 2. When you want to refer to the IP address of interface x, write "x". When you want to refer to the MAC address of interface x, write "x". If a field is not applicable, indicate that with a "–". To repeat a field from the above cell, write ".

#	Source MAC	Dest MAC	Source IP	Dst IP	Transp. prot.	Src Port	Dst Port	Application & Purpose
1	a	broadcast	-	-	-	-	-	ARP request for f's IP
2	f	a	-	-	-	-	-	ARP reply
3	a	f	a	e	UDP	2000	53	DNS query for e's IP
4	i	broadcast	-	-	-	-	-	ARP request for e's IP
5	e	i	-	-	-	-	-	ARP reply
6	i	e	a	e	UDP	2000	53	DNS query for e's IP
7	e	i	e	a	UDP	53	2000	DNS reply
8	f	a	e	a	UDP	53	2000	DNS reply
9	a	f	a	e	TCP	3000	80	TCP SYN
10	i	e	a	e	TCP	3000	80	TCP SYN
11	e	i	e	a	TCP	80	3000	TCP SYN ACK
11	f	a	e	a	TCP	80	3000	TCP SYN ACK
12	a	f	a	e	TCP	3000	80	HTTP GET index
13	i	e	a	e	TCP	3000	80	HTTP GET index
14	e	i	e	a	TCP	80	3000	HTTP OK
15	f	a	e	a	TCP	80	3000	HTTP OK
16	a	f	a	e	TCP	3000	80	HTTP GET image
17	i	e	a	e	TCP	3000	80	HTTP GET image
18	e	i	e	a	TCP	80	3000	HTTP OK
19	f	a	e	a	TCP	80	3000	HTTP OK

Table 2: Packets transmitted or forwarded by all end-hosts and IP routers in Question 3

Question 4 (5 points):

Show the forwarding table of each link-layer switch right after the last packet you stated above has arrived at its destination. Assume that no other traffic was exchanged.

a) Switch S_1:

Destination MAC address	Output link
a	o
f	p

b) Switch S_2:

Destination MAC address	Output link
i	q
e	r

c) Switch S_3:

Destination MAC address	Output link
a	s

Question 5 (5 points):

Suppose there is a firewall between S_1 and R_1. Fill the firewall table (use as many rows and columns as necessary) such that end-systems $B1 \ldots B999$ can access the web pages hosted by `www.epfl.ch` but end-system A cannot. Allow the minimum amount of traffic that accomplishes this goal.

Action	Protocol	Source IP	Dest. IP	Source Port	Dest. Port
deny	any	100.0.0.1/32	any	any	any
deny	any	any	100.0.0.1/32	any	any
allow	UDP	100.0.0.0/22	100.0.4.201/32	any	53
allow	UDP	100.0.4.201/32	100.0.0.0/22	53	any
allow	TCP	100.0.0.0/22	100.0.4.201/32	any	80
allow	TCP	100.0.4.201/32	100.0.0.0/22	80	any
deny	all	all	all	all	all

Problem 3 (20 points)

In the context of this problem, Alice wants to communicate with Bob and achieve some security properties. Persa is an adversary.

Question 1 (4 points):

Consider the scenario where Persa is sitting on the communication channel between Alice and Bob. Alice sends a message m to Bob. In each scenario, explain why or why not authenticity (the message is indeed coming from Alice) is guaranteed.

Scenarios:

a. Alice sends $\left[m,\ H\left(K_B^+,\ m\right)\right]$.

b. Alice sends $\left[m,\ H\left(K_A^+,\ m\right)\right]$.

where:

- K_A^+ and K_B^+ are Alice's and Bob's public keys, respectively.
- H is a cryptographic hash function that is known to everyone.

Authenticity does not hold in either scenario. This is because public keys K_B^+ and K_A^+ as well as hash function H are publicly available, hence anyone can generate $H\left(K_B^+,\ m\right)$ or $H\left(K_A^+,\ m\right)$. Moreover, Bob cannot verify $H\left(K_A^+,\ m\right)$, because he does not know K_A^-.

Question 2 (5 points):

Consider the scenario where Persa is sitting on the communication channel between Alice and Bob. Alice uses the following protocol to send a sequence of messages $m_1, m_2, ..., m_N$ to Bob:

- Alice sends m_1.
- Alice sends m_2.
- ...
- Alice sends m_N.
- Alice sends $H(K, m_1), H(K, m_2), ..., H(K, m_N)$.

where:

- K is a symmetric key, shared between Alice and Bob.
- H is a cryptographic hash function that is known to everyone.

Bob wants to ensure that the messages were (a) indeed sent by Alice and (b) they were sent by Alice in the order in which he received them. Does this protocol guarantee each of these? Explain why or why not.

(a) Bob can verify that Alice sent each message m_i at least once, because, apart from Bob, Alice is the only one who knows K, hence the only one who can generate $H(K, m_i)$.

However, replay attacks are possible, because Persa can intercept Alice's last message, read any MAC she wants and append it to Alice's last message where and as many times as she wants.

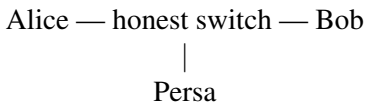
For example:

- Alice sends m_1.
- Persa sends m_1.
- Alice sends m_2.
- ...
- Alice sends m_N.
- Alice sends $H(K, m_1), H(K, m_2), ..., H(K, m_N)$.
- Persa modifies Alice's last message to $H(K, m_1), H(K, m_1), H(K, m_2), ..., H(K, m_N)$.

(b) Bob cannot verify that Alice sent the messages in the order that Bob received them, because the MACs in Alice's last message do not embed information about message order. Hence, Persa can intercept Alice's last message and reorder the MACs arbitrarily.

Question 3 (6 points):

Consider the scenario where Persa is NOT on the communication channel between Alice and Bob:



where the switch is honest in the sense that it always forwards packets to the destination specified by the sender.

Alice sends a message to Bob. Bob knows Alice's true IP address. In each scenario, explain why or why not authenticity (the message is indeed coming from Alice) is guaranteed.

Scenarios:

a. Alice sends the message to Bob using UDP.

b. Alice sends the message to Bob using TCP: she establishes a TCP connection to Bob, sends her message using the connection, then closes the connection.

a. Authenticity is not guaranteed, because Persa can spoof Alice's IP address, i.e., send a message to Bob using Alice's IP address as the source IP address.

b. Authenticity is guaranteed in the following sense: To send a message over TCP, Persa must first establish a TCP connection with Bob, i.e., send a SYN and receive a SYN ACK. If Persa spoofs Alice's IP address, Bob will send his SYN ACK to Alice. Since Persa is not on the communication channel between Bob and Alice, she cannot see Bob's SYN ACK, hence is very unlikely to guess Bob's initial sequence number. If Persa tries to complete the 3-way handshake but ACKs the wrong sequence number, Bob will know that she is an impersonator.

(Lab) Question 4 (5 points):

What is an SSH fingerprint and what is it useful for?

An SSH fingerprint is a summary of a public key, which can be used to authenticate the public key, i.e., verify that it belongs to a given entity. The small size of the fingerprint makes it easy for humans to read and exchange out of band, e.g., over the phone.

Problem 4 (35 points)

Assume the following for all the questions in this problem:

- The maximum segment size is $\texttt{MSS} = 1$ byte.
- The round trip time (RTT) is the same in both directions.
- Each TCP receiver sends an ACK every time it receives a data segment.
- Each TCP sender's retransmission timeout is fixed and equal to $2 \times RTT$.

When you complete the diagram in Question 1, the following information should be visible:

- All the segments (including the ACKs) exchanged between the communicating end-hosts.
- The sequence numbers of all data segments.
- The acknowledgment numbers of all ACKs.
- The state of the TCP sender's congestion-control algorithm.
- The status of the TCP sender's congestion window and its size (`cwnd`) in bytes.
- The value of the TCP sender's congestion threshold (`ssthresh`) in bytes.
- If your answer includes any dropped segments or ACKs, mark them clearly.
- If your answer includes any timeouts, mark them clearly and indicate the duration of each timeout and the sequence number of the data segment that timed out.

Question 1 (10 points):

In this question, Fast Retransmit/Fast Recovery are DISABLED.

Alice establishes a TCP connection to Bob and then sends 12 bytes of data.
The 3rd, 5th, 6th, 8th, 9th, and 10th segment **sent by Bob** (counting from the SYN ACK) is dropped.
No other segment, sent by Alice or Bob, is dropped or corrupted.

Show all the segments sent by Alice and Bob, including connection setup (not connection teardown), by completing the diagram in Figure 2 in the next page.

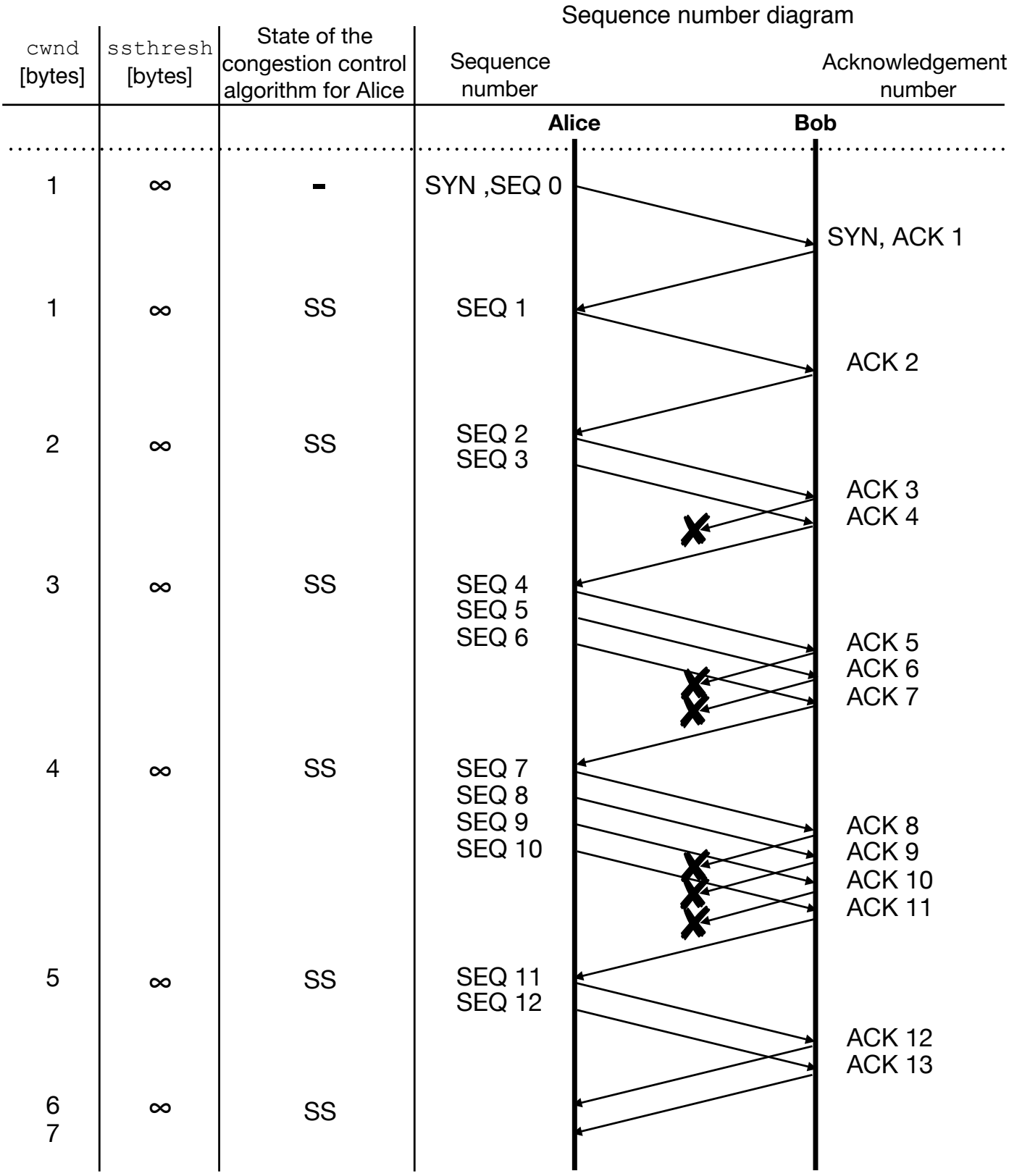


Figure 2: Sequence diagram to be completed for Question 1.

Question 2 (10 points):

In this question, Fast Retransmit/Fast Recovery are ENABLED.

When a TCP sender receives 3 duplicate ACKs, she takes that as a hint that a segment was lost, and she retransmits the oldest unacknowledged segment.

(a) Describe a scenario where this mechanism makes the sender retransmit a segment unnecessarily. Draw a small diagram to illustrate your scenario.

When reordering in the network happens, such scenario can occur:

- Receiver sends ACK N.
- Sender sends SEQ N, N+1, N+3, and N+3.
- Due to reordering inside the network, receiver receives N+1, N+2, N+3, N and sends ACK N, ACK N, ACK N, ACK N+4.
- Once the sender receives the three duplicate ACKs, she retransmits SEQ N unnecessarily.

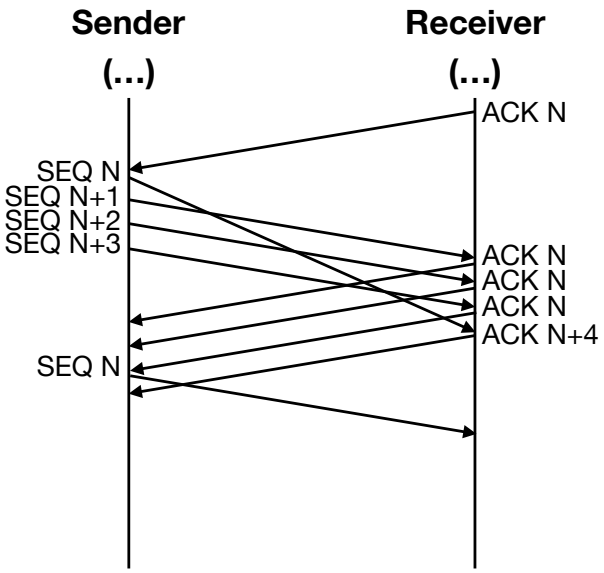


Figure 3: Scenario example for Question 2.

(b) When the sender enters Fast Recovery, she sets her congestion window to `cwnd=ssthresh+3`. Why does the sender inflate its congestion window (why does it not set `cwnd=ssthresh`)? Why does it do it by 3?

Suppose a sender has already sent `cwnd` unacknowledged bytes, hence can send no more.

If the sender receives 3 duplicate ACKs, says for byte N, it infers that the receiver did not receive byte N, but it did receive bytes $N+1$, $N+2$, and $N+3$.

Since the receiver received 3 bytes, the sender should now be able to send 3 new bytes. It cannot do that, however, because it has not received any *new* ACKs, hence cannot advance its window.

If the window is stuck, and the sender has already sent `cwnd` unacknowledged bytes, the only way to send 3 new bytes is by inflating (artificially increasing) `cwnd` by 3.

Question 3 (15 points):

(a) How does TCP's congestion control algorithm guess that there is network congestion and that the sender should decrease her congestion window? Answer in one short sentence.

Based on packet loss: if the sender times out waiting for an ACK or receives 3 duplicate ACKs, this indicates packet loss, which implies network congestion.

(b) A network architect proposes to make the packet queues of all packet switches/routers very very large, in order to ensure almost 0 packet loss. How would this affect TCP's congestion control algorithm? Do you think it would do its job better or worse? Justify your answer.

This would make TCP congestion control less effective.

TCP detects network congestion by detecting packet loss. If packet queues were very very large, there would not be significant packet loss, hence, TCP would not be able to detect network congestion.

In particular, if packet queues were very very large, network congestion would lead to increased queuing delays. This would have two effects on TCP congestion control:

- The sender would receive ACKs less frequently, hence increase its congestion window more slowly.
- However, the sender would observe longer RTTs to the receiver and adapt its timeout accordingly. As a result, the sender would not time out, hence would not reset its congestion window, despite the presence of network congestion.

Scratch Paper

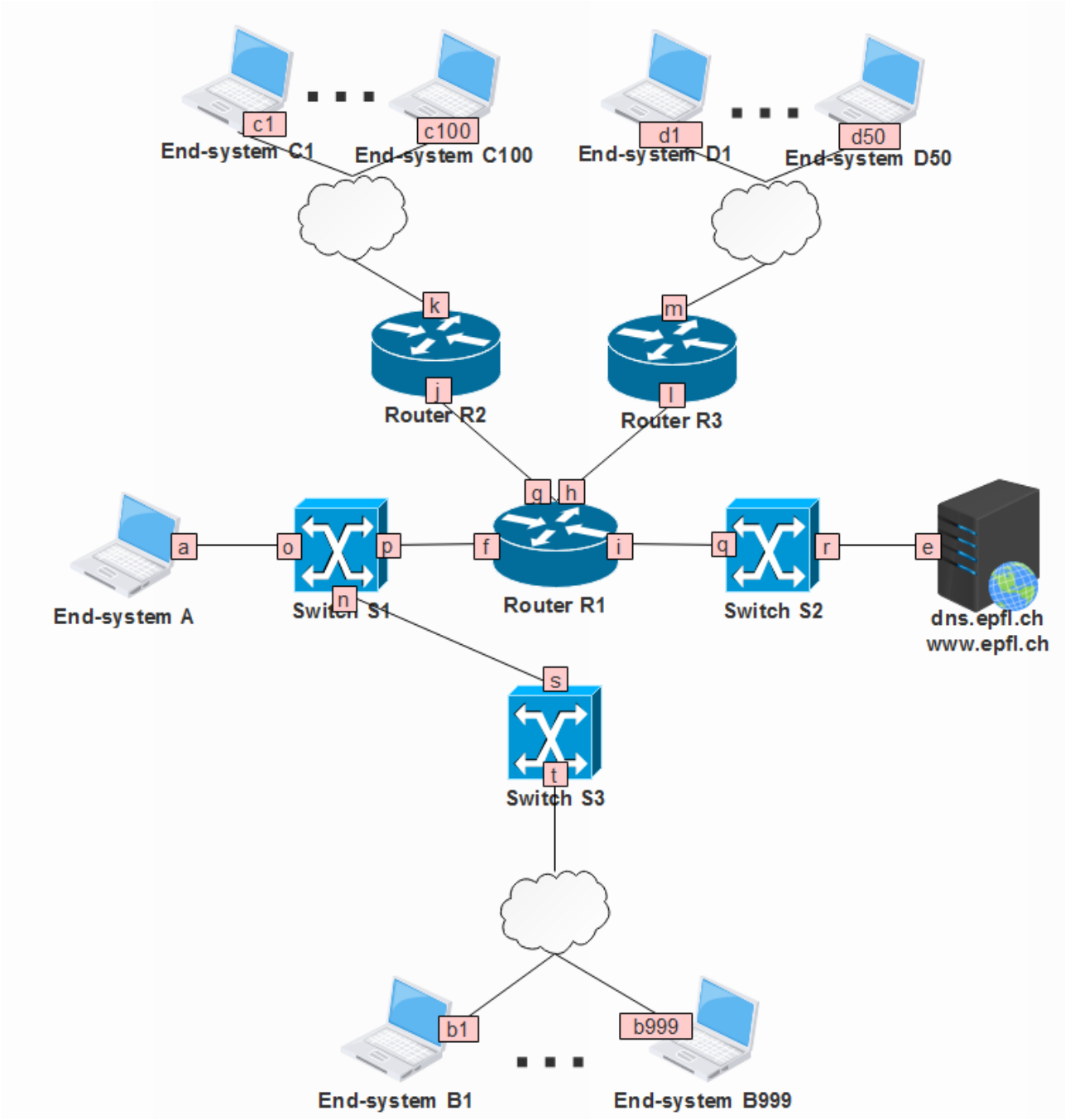


Figure 4: The Network Topology used in Problem 2