# Computer Networks - Midterm

October 28, 2016

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 19 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. The network performance requirements of a voice & video chat application are:
   - (a) low latency, low throughput.
   - (b) low latency, high throughput. *(Correct)*
   - (c) high latency, low throughput.
   - (d) high latency, high throughput.
2. What is a major drawback of "connection-switching" versus "packet-switching"?
   - (a) Network resources are not used efficiently. *(Correct)*
   - (b) Queuing delays are substantially longer.
   - (c) They require the use of optical fibers.
   - (d) They do not support DNS resolution queries.
3. All root DNS servers become unreachable. This has the following effect:
   - (a) No names can be resolved.
   - (b) All names can be resolved, but it takes longer.
   - (c) As time passes, fewer and fewer names can be resolved. *(Correct)*
   - (d) Nothing.
4. Distributed hash tables (DHTs) are typically used in peer-to-peer file sharing applications to:
   - (a) store very large files that do not fit in the peers memory.
   - (b) segment content into pieces and distribute them among the peers.
   - (c) check the correctness of the shared content.
   - (d) locate the peers that store certain content. *(Correct)*
5. Which guarantees does TCP offer?
   - (a) Guaranteed security
   - (b) Guaranteed performance
   - (c) Reliable delivery *(Correct)*
   - (d) All of the above

6. A process can use the same socket to communicate with multiple remote processes in:

   (a) TCP
   (b) UDP *(Correct)*
   (c) None of the above
   (d) Both TCP and UDP

7. A client application creates two TCP sockets and successfully connects them in parallel to a remote TCP server application that is listening to port 9000. All packets that the client application sends to the server application share the same source address and destination IP address. Which of the following is true?

   (a) Both client sockets can generate packets with destination port 9000. (at the same time) *(Correct)*
   (b) Both client sockets can generate packets with source port 9000. (at the same time)
   (c) Both the above are correct.
   (d) Both of the above are false. *(This was also taken as correct)*

8. Consider a link with a transmission rate of 10Mbps, which is shared between multiple users that want to transfer data at a rate of 1Mpbs. How many concurrent transfers can be supported?

   (a) There is no limit.
   (b) $\leq 10$ *(Correct)*
   (c) $> 10$
   (d) It depends on whether we use connection switching or packet switching.

9. Consider a switch that forwards packets of size 500bits over a link with transmission rate of 1Kbps, using a buffer size of 10Kbits. What is the maximum queuing delay that the packets may experience?

   (a) 10 seconds.
   (b) 9.5 seconds. *(Correct)*
   (c) $\infty$
   (d) We cannot say. It depends on the arrival pattern (burstiness) of the packets.

10. A host has recently accessed www.epfl.ch, which contains two images. The host knows the IP address of the web server. How many round-trip-times (RTTs) are necessary to retrieve the entire page again?

    (a) At most 5 RTTs.
    (b) At least 3 RTTs. *(Correct)*
    (c) At most 3 RTTs.
    (d) At least 2 RTTs. *(This was also taken as correct)*

## 2 Web Browsing and DNS (38 points)

**Setup:**

Two users, Alice and Bob, are logged into the workstations `alice.ethz.ch` and `bob.ethz.ch`, all located inside ETHZ's network.

ETHZ offers a web server `www.ethz.ch`, a local name server `ns.ethz.ch` and a web proxy `proxy.ethz.ch`. EPFL offers a web server `www.epfl.ch` and a local name server `ns.epfl.ch`.

The setup is illustrated in Figure 1. Make the following assumptions:

- All DNS and HTTP caches are initially empty.
- All web-browsing traffic originating in the ethz.ch domain goes through the web proxy.
- All web browsers in the ethhz.ch domain know the IP address of the web proxy.
- The DNS server `ns.ethz.ch` is also the *authoritative* DNS server for the `ethz.ch` domain.
- The DNS server `ns.epfl.ch` is also the *authoritative* DNS server for the `epfl.ch` domain.
- Each HTTP message fits in a single packet.
- The web browsers, proxies and servers use persistent HTTP connections.
- The contents of the web server running on `www.epfl.ch` never change.

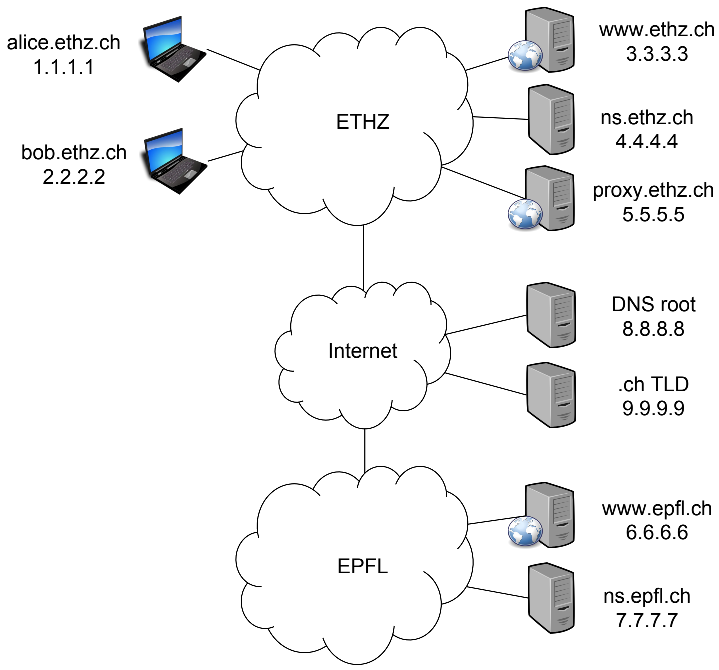


Figure 1: The setup for the exercise on Web browsing and DNS

**Question 1 (10 points):**

Alice uses her web browser to access URL `http://www.epfl.ch/index.html`. This page includes an image with URL `http://www.epfl.ch/logo.png`.

List all packets that are exchanged in the entire network, including any connection-setup packets, by filling Table 1. For each packet, show the source and destination IP address, the application-layer protocol, the transport-layer protocol and the purpose of the packet, as in the two examples.

| Packet | Source IP | Dest. IP | Application protocol | Transport protocol | Purpose |
|---|---|---|---|---|---|
| ex.1 | 1.0.0.1 | 1.0.0.2 | HTTP | | HTTP reply with image.png |
| ex.2 | 2.0.0.1 | 2.0.0.2 | DNS | | What is the IP address of example.com? |
| 1 | 1.1.1.1 | 5.5.5.5 | HTTP | TCP | Connection setup: TCP SYN |
| 2 | 5.5.5.5 | 1.1.1.1 | HTTP | TCP | Connection setup: TCP SYN ACK |
| 3 | 1.1.1.1 | 5.5.5.5 | HTTP | TCP | HTTP request index.html, host: www.epfl.ch |
| 4 | 5.5.5.5 | 4.4.4.4 | DNS | UDP | What is the IP address of www.epfl.ch? |
| 5 | 4.4.4.4 | 8.8.8.8 | DNS | UDP | What is the IP address of www.epfl.ch? |
| 6 | 8.8.8.8 | 4.4.4.4 | DNS | UDP | Ask DNS server: 9.9.9.9 |
| 7 | 4.4.4.4 | 9.9.9.9 | DNS | UDP | What is the IP address of www.epfl.ch? |
| 8 | 9.9.9.9 | 4.4.4.4 | DNS | UDP | Ask DNS server: 7.7.7.7 |
| 9 | 4.4.4.4 | 7.7.7.7 | DNS | UDP | What is the IP address of www.epfl.ch? |
| 10 | 7.7.7.7 | 4.4.4.4 | DNS | UDP | IP address of www.epfl.ch is 6.6.6.6 |
| 11 | 4.4.4.4 | 5.5.5.5 | DNS | UDP | IP address of www.epfl.ch is 6.6.6.6 |
| 12 | 5.5.5.5 | 6.6.6.6 | HTTP | TCP | Connection setup: TCP SYN |
| 13 | 6.6.6.6 | 5.5.5.5 | HTTP | TCP | Connection setup: TCP SYN ACK |
| 14 | 5.5.5.5 | 6.6.6.6 | HTTP | TCP | HTTP request of index.html |
| 15 | 6.6.6.6 | 5.5.5.5 | HTTP | TCP | HTTP reply with index.html |
| 16 | 5.5.5.5 | 1.1.1.1 | HTTP | TCP | HTTP reply with index.html |
| 17 | 1.1.1.1 | 5.5.5.5 | HTTP | TCP | HTTP request of logo.png, host: www.epfl.ch |
| 18 | 5.5.5.5 | 6.6.6.6 | HTTP | TCP | HTTP request of logo.png |
| 19 | 6.6.6.6 | 5.5.5.5 | HTTP | TCP | HTTP reply with logo.png |
| 20 | 5.5.5.5 | 1.1.1.1 | HTTP | TCP | HTTP reply with logo.png |

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

Alternatively, a recursive DNS solution would also be acceptable, incurring the following changes:

| | | | | | |
|---|---|---|---|---|---|
| 6 | 8.8.8.8 | 9.9.9.9 | DNS | UDP | What is the IP address of www.epfl.ch? |
| 7 | 9.9.9.9 | 7.7.7.7 | DNS | UDP | What is the IP address of www.epfl.ch? |
| 8 | 7.7.7.7 | 9.9.9.9 | DNS | UDP | The IP address of www.epfl.ch is 6.6.6.6 |
| 9 | 9.9.9.9 | 8.8.8.8 | DNS | UDP | The IP address of www.epfl.ch is 6.6.6.6 |
| 10 | 8.8.8.8 | 4.4.4.4 | DNS | UDP | The IP address of www.epfl.ch is 6.6.6.6 |

Alternative solution in Question 1 using recursive DNS resolution

**Question 2 (8 points):**

Right after Alice views `http://www.epfl.ch/index.html`, Bob uses his web browser to access URL `http://www.epfl.ch/quiz.html`. This page includes an image with URL `http://www.epfl.ch/logo.png`.

List all packets that are exchanged in the entire network, including any connection-setup packets, by filling Table 2. For each packet, show the source and destination IP address, the application-layer protocol, the transport-layer protocol and the purpose of the packet.

| Packet | Source IP | Dest. IP | Application protocol | Transport protocol | Purpose |
|---|---|---|---|---|---|
| 1 | 2.2.2.2 | 5.5.5.5 | HTTP | TCP | Connection setup: TCP SYN |
| 2 | 5.5.5.5 | 2.2.2.2 | HTTP | TCP | Connection setup: TCP SYN ACK |
| 3 | 2.2.2.2 | 5.5.5.5 | HTTP | TCP | HTTP request `quiz.html`, host: `www.epfl.ch` |
| 4 | 5.5.5.5 | 6.6.6.6 | HTTP | TCP | Connection setup: TCP SYN |
| 5 | 6.6.6.6 | 5.5.5.5 | HTTP | TCP | Connection setup: TCP SYN ACK |
| 6 | 5.5.5.5 | 6.6.6.6 | HTTP | TCP | HTTP request of `quiz.html` |
| 7 | 6.6.6.6 | 5.5.5.5 | HTTP | TCP | HTTP reply with `quiz.html` |
| 8 | 5.5.5.5 | 2.2.2.2 | HTTP | TCP | HTTP reply with `quiz.html` |
| 9 | 2.2.2.2 | 5.5.5.5 | HTTP | TCP | HTTP request `logo.png`, host: `www.epfl.ch` |
| 10 | 5.5.5.5 | 6.6.6.6 | HTTP | TCP | HTTP *header* request of `logo.png` |
| 11 | 6.6.6.6 | 5.5.5.5 | HTTP | TCP | HTTP *header only* of `logo.png` |
| 12 | 5.5.5.5 | 2.2.2.2 | HTTP | TCP | HTTP reply with `logo.png` |

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

In both Questions 1 & 2 the followins solutions were also accepted:

- Alice and Bob, respectively, contact `ns.ethz.ch` to learn the IP of `www.epfl.ch`.
- `proxy.ethz.ch` requests `logo.png` *immediately after* receiving `index.html` in Question 1, but before Alice requests for it. Nevertheless, Alice should eventually request `logo.png`.

The following common mistakes were observed in Questions 1 & 2:

- Completely omitting the use of `proxy.ethz.ch` for the purposes of web traffic.
- Using `proxy.ethz.ch` as the DNS server to perform DNS querries instead of `ns.ethz.ch`.
- Demanding the IP address of `proxy.ethz.ch`; it is known to both Alice and Bob.
- `proxy.ethz.ch` has already cached the IP of `www.epfl.ch` in Question 2; therefore it should not request it again from `ns.ethz.ch`.
- Requesting both the header and contents of `logo.png` from `www.epfl.ch` in Question 2.
- Omitting to request the header *only* of `logo.png` from `www.epfl.ch` in Question 2.
- There is no `GET` request for `logo.png` from Alice or Bob. They have to request the image in order to receive it.
- The connection setup between `proxy.ethz.ch` and `www.epfl.ch` in Question 2 was omitted, unless an assumption was clearly stated that the connection was still open.

**Question 3 (6 points):**

Now suppose that the web server at `www.epfl.ch` uses cookies. Do you think that this might change your answer to Question 2? If so, briefly describe how (you do not need to list all the packets again, just explain what would be the difference, if any, in a couple of sentences).

Clarification: we have not discussed cookies in detail, so you are not expected to immediately know the answer. Just try to make a guess, based on your understanding of the role of cookies.

*Given that HTTP is a stateless protocol cookies are used from web sites to connect subsequent requests from the same user. They are personalised information that is stored in the user's browser and they are sent as part of the HTTP request. (2)*

*Since that cookies are personalised information it wouldn't make sense for the proxy to cache any response that includes cookies, becuse this could lead to leaking sensitive information about users. (2)*

*The sequence of messages in Q2 would change. The web proxy would not cache logo.png since the response from the epfl server includes Alice's cookie. So, when Bob requests logo.png the proxy has to re-ask the epfl server to retrieve it again. (2)*

**Question 4 (8 points):**

Web caching and DNS caching use two different mechanisms for ensuring that the cached data is not stale.

a. Why is that?

*Web-caching uses a conditional request in order to verify the frenshness of data (1). DNS uses a time-to-live (TTL): each record is accompanied by a time after which its entry in the cache is invalidated (1).*

*The mechanisms are different because the data to be cached has different properties: Many Web objects (images, videos, etc) are much larger compared to DNS records (2). Also, Web sites are updated much more frequently than DNS records (2).*

b. Would it make sense for web caching to use the mechanism used by DNS caching and vice versa?

*Using a conditional request in DNS wouldn't make sense due to the fact that the records are small, thus it would take almost the same amount of time to request the record without any caching. So, caching would be useless.*

*Using TTLs for Web objects would not make sense either. Since websites contain many objects that might be changed frequently, it would be difficult to prevent the use of stale information. Using small values of the TTL would help solve this problem, but it would create another: cached entries might expire although the objects have not been changed, generating unnecessary traffic. (2)*

**Question 5 (6 points):**

Many ETHZ users often visit the URL *http://www.bank.ch*. Alice wants to launch an attack in order to redirect the traffic between all these users and *www.bank.ch* to her own machine, which has IP address 1.1.1.1. At some point, there is a power failure on the ETHZ campus, which causes all the servers to reboot and brings them back to their initial state (the one they had at the beginning of Problem 2). Alice uses this as an opportunity to launch her attack.

a. Explain briefly how she will do it, i.e., what traffic will she send out and what will that cause.

   *There are two possible answers (either one is valid):*

   *Method 1: DNS cache poisoning attack on the DNS resolver from ETHZ, ns.ethz.ch.*

   *Right after power is restored, she floods the resolver with DNS replies containing a fake A record for bank.ch with her IP address 1.1.1.1, appearing to originate from one of the DNS servers that will be queried by ns.ethz.ch, for example a root server or the nameserver of bank.ch.*

   *When the first user tries to visit bank.ch, a DNS query for bank.ch is sent to the resolver, which will send it further; it will accept one of the fake replies before the legitimate one arrives and will store in its cache the incorrect record. Afterwards, whenever the ETHZ proxy (or any other ETHZ hosts) sends a query for bank.ch, it will receive the fake record and will connect to the attacker's website.*

   *Method 2: similar, but attacking the proxy server instead of the DNS server.*

   *Either one gets 4 points.*

   *Attacking the users directly does not work since they access the website through the proxy, thus do not send any DNS requests. But even if we assumed they didn't use the proxy, it is difficult to attack all users at once due to the large traffic volume that would have to be sent in a short amount of time. The attack would probably succeed only for some of the users. This answer would get only 2 points.*

b. Why is the server reboot necessary for Alice to launch her attack?

   *Because the record must not be in the cache for the attack to succeed. Otherwise the attack would be ineffective until the time the entry expires and a query is sent again. However Alice has no way to know when this happens, thus cannot launch her attack at the right time. (2 points)*

## 3 Traffic log **(12 points)**

Consider a machine on the EPFL campus that hosts a web server. The machine has two hostnames: *moodle.epfl.ch* and *images.epfl.ch* (in other words, both of these hostnames map to the same IP address). Suppose *http://moodle.epfl.ch/index.html* references *http://images.epfl.ch/logo.jpg* and *http://images.epfl.ch/background.jpg*.

The following log was captured at the web server:

| Index | Time | Source | Destination | Protocol | Length | Description |
|---|---|---|---|---|---|---|
| 1 | 0 | 1.2.3.4 | 5.6.7.8 | HTTP | 76 | TCP handshake request |
| 2 | 0 | 5.6.7.8 | 1.2.3.4 | HTTP | 76 | TCP handshake reply |
| 3 | 0.1 | 1.2.3.4 | 5.6.7.8 | HTTP | 100 | GET request for /index.html, Host: moodle.epfl.ch |
| 4 | 0.1 | 5.6.7.8 | 1.2.3.4 | HTTP | 480 | GET reply with /index.html |
| 5 | 0.4 | 1.2.3.4 | 5.6.7.8 | HTTP | 76 | TCP handshake request |
| 6 | 0.4 | 5.6.7.8 | 1.2.3.4 | HTTP | 76 | TCP handshake reply |
| 7 | 0.5 | 1.2.3.4 | 5.6.7.8 | HTTP | 109 | GET request for /logo.jpg, Host: images.epfl.ch |
| 8 | 0.5 | 5.6.7.8 | 1.2.3.4 | HTTP | 960 | GET reply with /logo.jpg |
| 9 | 0.6 | 1.2.3.4 | 5.6.7.8 | HTTP | 76 | TCP handshake request |
| 10 | 0.6 | 5.6.7.8 | 1.2.3.4 | HTTP | 76 | TCP handshake reply |
| 11 | 0.7 | 1.2.3.4 | 5.6.7.8 | HTTP | 100 | GET request for /background.jpg, Host: images.epfl.ch |
| 12 | 0.7 | 5.6.7.8 | 1.2.3.4 | HTTP | 1250 | GET reply with /background.jpg (packet 1 of 4) |
| 13 | 0.71 | 5.6.7.8 | 1.2.3.4 | HTTP | 1250 | GET reply with /background.jpg (packet 2 of 4) |
| 14 | 0.72 | 5.6.7.8 | 1.2.3.4 | HTTP | 1250 | GET reply with /background.jpg (packet 3 of 4) |
| 15 | 0.73 | 5.6.7.8 | 1.2.3.4 | HTTP | 1250 | GET reply with /background.jpg (packet 4 of 4) |

Table 3: Packet capture log. Time is shown in seconds, with 2 decimal places; the packet length is shown in bytes.

Answer the following questions, briefly justifying your answers:

a. What is the IP address of the web server?

*The address of the web server is 5.6.7.8. We can see this from the source IP address of the HTTP reply in packet 4.*

b. How many clients connect to the web server? What are their IP addresses?

*Only one client connects to the web server, with IP address 1.2.3.4. We can see this from the source IP address of the HTTP request in packet 3, for example.*

c. Do the clients use persistent or non-persistent connections?

*The client is using non-persistent connections. We can see that it creates a different connection for each HTTP request sent to images.epfl.ch.*

d. Estimate the round-trip time between the client(s) and the web server.

*The round-trip time between the client and the server can be computed from the time it takes the TCP handshake reply from packet 2 to travel to the client, plus the time it takes the GET request from packet 3 to reach the server; minus the transmission delays which we approximate to zero, since both packets are small. From the capture times of the packets, this is 0.1 - 0 - 0 = 0.1 seconds or 100 ms.*

e. Estimate the time it took the client(s) to perform DNS resolution of *images.epfl.ch*.

*This can be computed from the difference between we received the TCP handshake request for the first image (packet 5) and the time we sent the page (packet 4), minus one RTT and the transmission delay of packet 4 (which we approximate to zero since the packet is small). Thus we have: d = 0.4 - 0.1 - 0.1 - 0 = 0.2 s = 200 ms.*

f. Estimate the rate of the link connecting the web server to the Internet (hint: from the transfer of */background.jpg*).

*This can be computed from the transfer of /background.jpg. We can see from packets 12 and 13 (also 13-14 and 14-15) that it takes 0.01 seconds to send 1250 bytes or 8 * 1250 = 10,000 bits. Thus the link rate is 10,000 / 0.01 b/s = 1 Mbps.*

## 4 File Distribution (40 points)

The network in Figure 2 contains the following elements:

- Server $S$, which stores a file of size $F$ bits.
- Nodes $1, 2, \ldots N$, where $N \geq 4$ and $N$ is even $(4, 6, 8, \ldots)$.
- Packet switches $S1$ and $S2$, which perform store-and-forward packet switching and introduce insignificant processing delays.
- Each link has length $\ell$ and propagation speed $c$.
- The transmission rate of the link between $S1$ and $S2$ is $\frac{R}{k}$ in each direction, where $k > 1$.
- The transmission rate of all the other links is $R$ in each direction.
- The end-systems communicate over UDP which means that, when you compute delays, you do not need to worry about connection setup.
- No other approximations/ simplifying assumptions can be made.

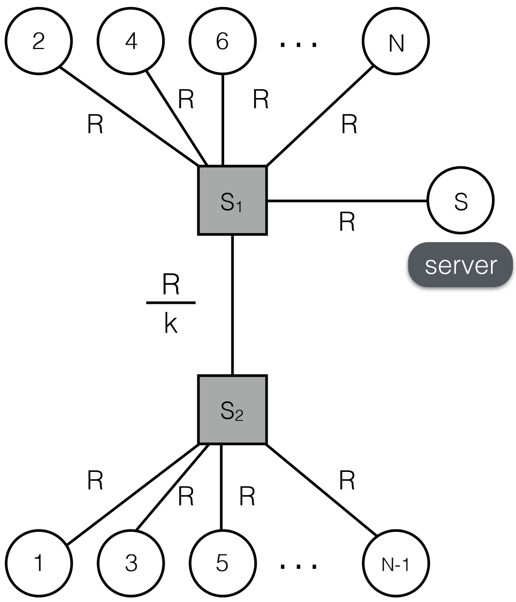


Figure 2

**Question 1 (3 points):**
Node 1 requests the file from Server $S$. Suppose the request message fits in a single packet of size $Q$ bits. How long does it take for the request message to reach $S$?

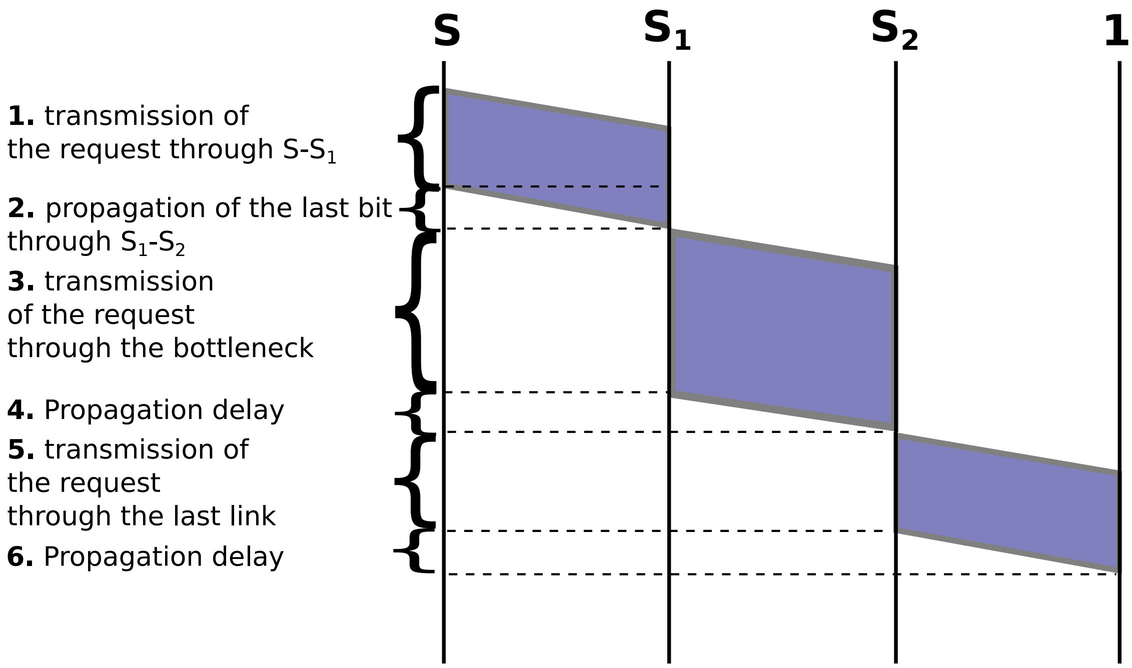


Figure 3: Solution to Question 1.

1. $\frac{Q}{R}$
2. $\frac{\ell}{c}$
3. $k\frac{Q}{R}$
4. $\frac{\ell}{c}$
5. $\frac{Q}{R}$
6. $\frac{\ell}{c}$

$$d_{total_1} = \frac{Q}{R} + k\frac{Q}{R} + \frac{Q}{R} + 3 \cdot \frac{\ell}{c} \tag{1}$$

**Question 2 (5 points):**
In response to the request of Node 1, Server $S$ splits the file into $P$ packets of equal length and sends them to Node 1, one after the other. Ignore the processing time for splitting the file into packets. How long does it take for the file to be fully delivered to Node 1?
(**Hint**: Careful, there may be queuing delays!)

*Let* $f = \frac{F}{P}$

1. $\frac{f}{R} + \frac{\ell}{c}$
2. $P \cdot k\frac{f}{R} + \frac{\ell}{c} = k\frac{F}{R} + \frac{\ell}{c}$
3. $\frac{f}{R} + \frac{\ell}{c}$

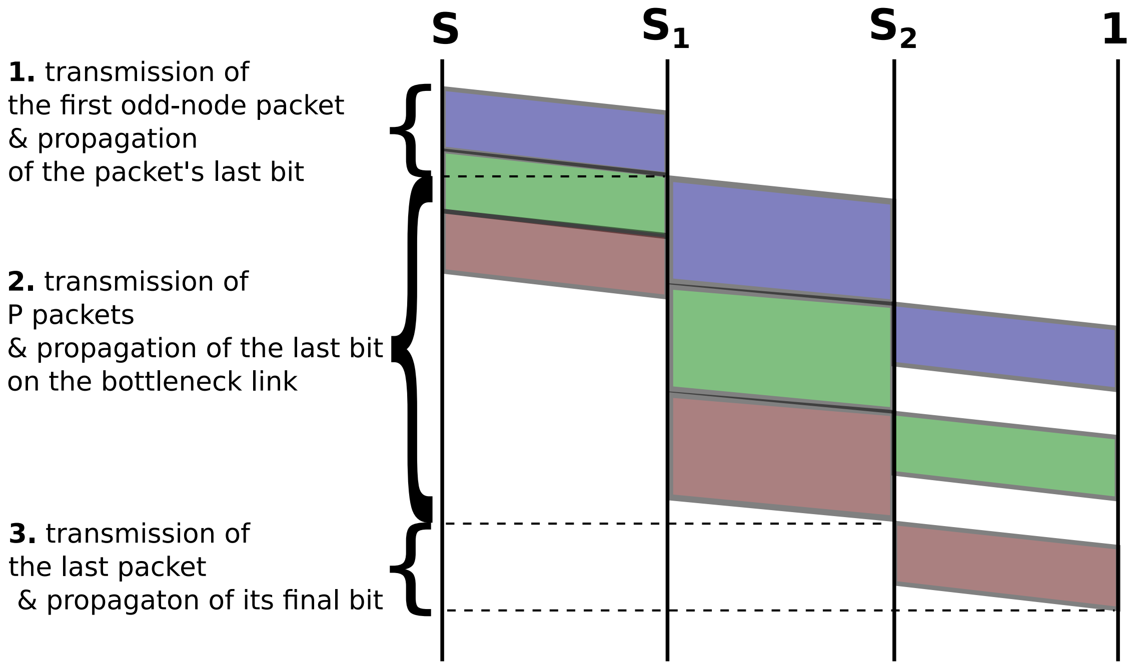


Figure 4: Solution to Question 2.

*Since all packets are of equal size and link $S_2 \rightarrow 1$ has higher throughput than $S_1 \rightarrow S_2$, we can trivially prove that packets will not experience queueing delay on $S_2 \rightarrow 1$.*

$$d_{total_2} = \frac{f}{R} + k \cdot \frac{F}{R} + \frac{f}{R} + 3 \cdot \frac{\ell}{c} \tag{2}$$

**Question 3 (12 points):**
Instead of only Node 1, all nodes $(1, 2, \dots N)$ request the file from Server $S$.

a. Suppose all nodes begin transmitting their requests at exactly the same time. Which requests will arrive earlier at the server, the ones coming from the odd nodes $(1, 3, 5, \dots)$ or the even nodes $(2, 4, 6, \dots)$?

   Briefly explain why.

b. After having received all $N$ requests, Server $S$ splits the file into $P$ packets of equal length and sends them to the nodes using the client-server approach.

   Suppose that the server handles the requests in the order in which they were received. For example, if the first request to arrive at the server is from Node $i$, the server will transmit all $P$ packets needed by Node $i$.

   Based on your answer to (a), how long will it take for the file to be fully delivered to all nodes?

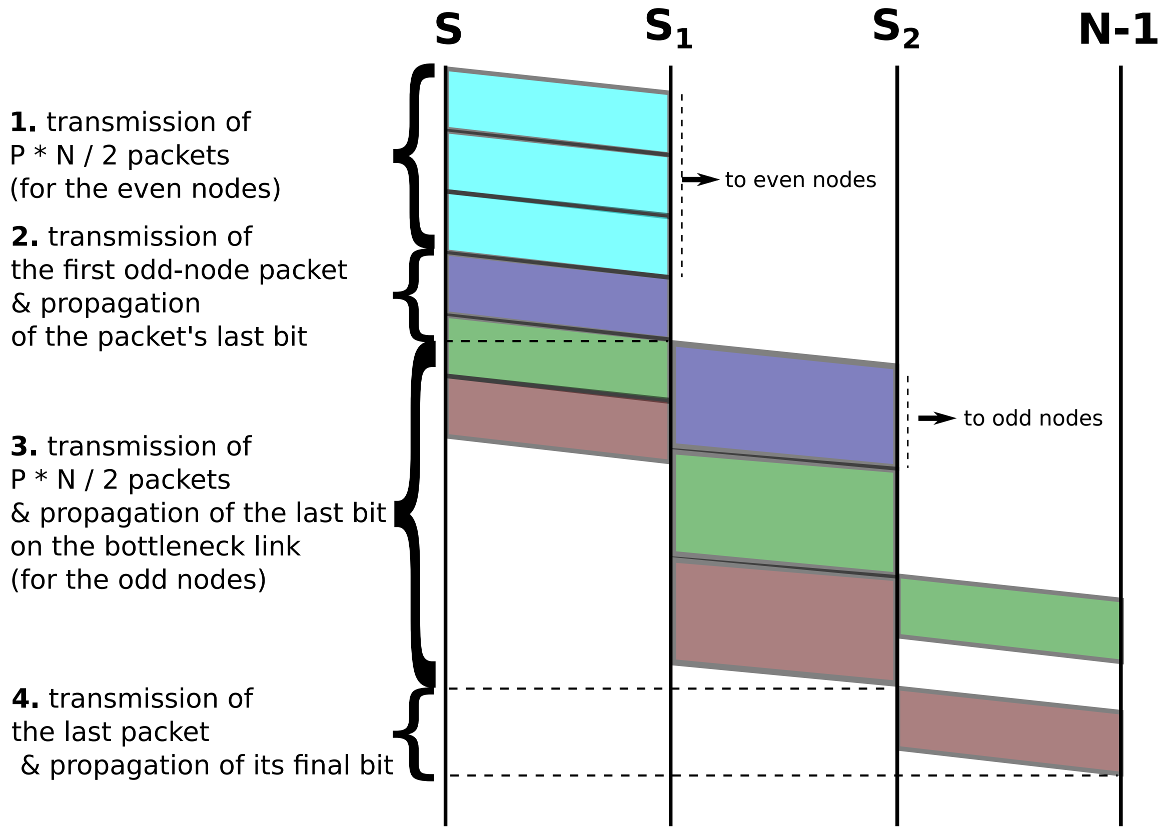


Figure 5: Solution to Question 3.

*a. Let us call the requests of the even nodes, "even requests" and the requests of the odd nodes, "odd requests".*

*The even requests will arrive earlier at $S$. This is because even nodes are closer to switch $S_1$, both in terms of propagation delay and transmission delay. Moreover even requests do not experience any queueing before they reach switch $S_1$.*

*As a result result, $_1$ will place all even requests before the odd requests in the output queue of link $S_1 \rightarrow S$. (3 pts)* *b. Let $f = \frac{F}{P}$. Then:*

1. $\frac{N}{2}P \cdot \frac{f}{R}$ *(4 pts)*
2. $\frac{f}{R} + \frac{\ell}{c}$
3. $\frac{N}{2}P \cdot k\frac{f}{R} + \frac{\ell}{c}$ *(3 pts)*
4. $\frac{f}{R} + \frac{\ell}{c}$ *(2 pts)*

$$d_{total_3b} = \frac{N}{2}P \cdot \frac{f}{R} + \frac{f}{R} + \frac{N}{2}P \cdot k\frac{f}{R} + \frac{f}{R} + 3 \cdot \frac{\ell}{c} \tag{3}$$

**Question 4 (12 points):**

a. Instead of using the client-server approach, the server S distributes the file using the following peer-to-peer (P2P) approach:

- The server $S$ first splits the file into 2 equally sized packets (chunks) and sends them to Node 1; when $S$ has sent both packets to Node 1, it stops distributing the file.
- As soon as Node $i$ receives a packet, it sends it to Node $i+1$; when Node $i$ has sent both packets to Node $i+1$, it stops redistributing the file.
- So, each of the two packets goes from $S$ to Node 1, then Node 2, etc.

The underlying physical topology is still the one shown in Figure 2. How long will it take for the file to be fully delivered to all nodes? Justify your answer.

b. Briefly describe a different P2P approach that could offer shorter file distribution time.

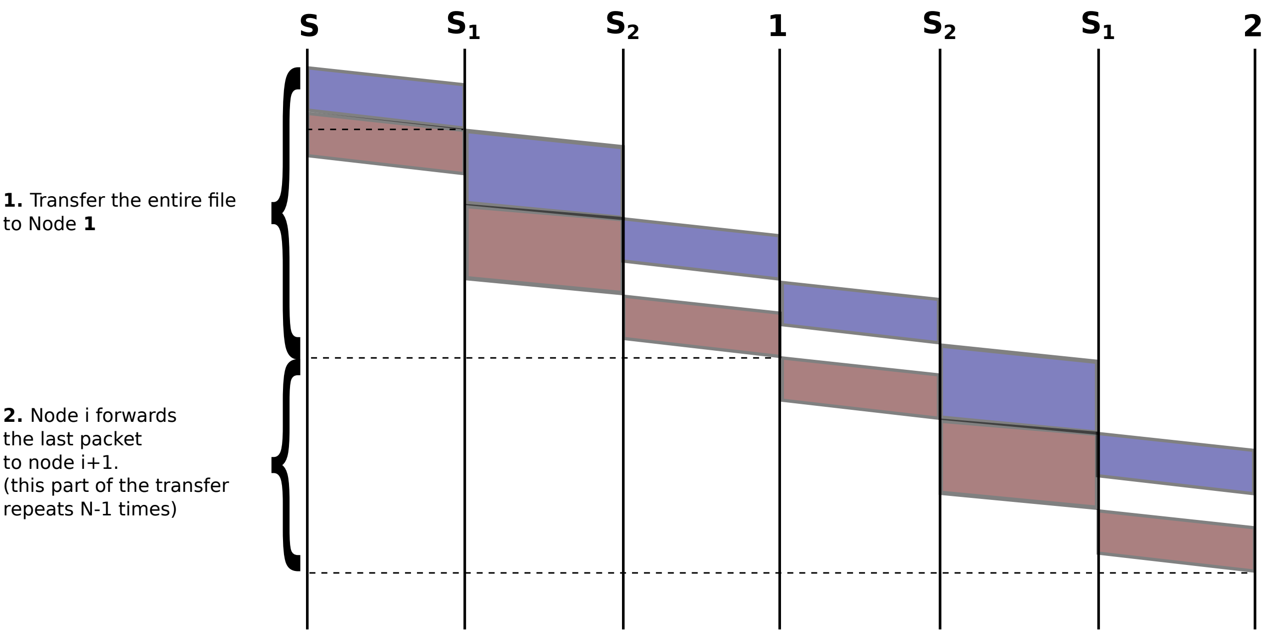


Figure 6: Solution to Question 4.

*a. Let $f = \frac{F}{2}$. Then:*

1. $\frac{f}{R} + 2 \cdot k\frac{f}{R} + \frac{f}{R} + 3\frac{\ell}{c}$
2. $(N-1) \cdot \left(\frac{f}{R} + k\frac{f}{R} + \frac{f}{R} + 3\frac{\ell}{c}\right)$ *(8 pts)*

$$d_{total_{4a}} = \frac{f}{R} + 2 \cdot \frac{k \cdot f}{R} + \frac{f}{R} + \left(\frac{f}{R} + \frac{k \cdot f}{R} + \frac{f}{R}\right) \cdot (N-1) + 3N\frac{\ell}{c} \quad (4)$$

*To prove that our result is correct, we need to prove that the second packet (shown in red in figure 6) will not experience any additional queueing from the first packet anywhere else in the network. (1 pts)*

*The quickest way to prove this is through a proof by induction. We claim that the two packets will arrive at different nodes with a time difference of $k\frac{f}{R}$ from each other.*

- *We first prove our claim for node* $1$*.*
- *Then, for node* $i$*, and for the given delay that we claim, we can prove that there will be no queueing as node* $i$ *transmits the packets to node* $i+1$
- *Finally, we also prove that the delay between the two packets as they arrive at node* $i+1$ *will remain the same.*

*Since both the base case (node* $1$*) and inductive step (node* $i$ *to node* $i+1$*) holds, all nodes will receive the two packets with the same delay between them.*

*We have also proven that the second packet does not experience additional delays due to the first packet when the two packets are transmitted with a delay of* $k\frac{f}{R}$*. Thus, the second packet will not experience any additional queueing after it reaches node* $1$*. Thus, our result is correct.*

b. *One way to improve the performance of our overlay is to make sure that the nodes that peer with one another are also more closely physically located. A trivial way to achieve this is make each node* $i$ *connect to node* $i+2$*, and connect node* $N-1$ *to node* $2$*.*

*The new overlay guarantees that all nodes will be on the same side of the bottleneck link. Most importantly, each node now only has to traverse 2 links instead of 3 links, thus dramatically cutting down the propagation delay.*

*Finally, If we carefully examine our overlay, perhaps the most important deficiency that holds performance back is that each node only contributes to the file transfer very briefly (i.e., just to forward the file to the next peer). This leaves most of the bandwidth capacity of our network underutilized.*

*In order to improve the performance of our P2P overlay in that regard, we should make it so that each node will transmit the file's contents to multiple other nodes, until all nodes have received the file. (3 pts)*

**Question 5 (8 points):**

a. An even node wants to launch a denial-of-service flooding attack on the server, such that none of the other nodes (odd or even) can reach the server.

   Is that possible? How would such an attack work?

b. Could an odd node launch such an attack?

*a. Yes, an even node can perform a bandwidth flooding attack. It can send requests to the server with a rate R. (4)*

*b. No, an odd node is restricted by the bottleneck link. The maxim rate that odd nodes can send traffic to the even nodes or the server is $R/k$. So, the even nodes will always be able to access the server. Such an attack will prevent the odd nodes from accessing the server, due to infinite queuing on S2. (4)*